

Algorithms and Today's Practitioner

Giuseppe Prencipe, University of Pisa

Cesare Zavattari and Alessandro Tommasi, CrowdEngineering, Pisa

John Favaro, Intecs SpA

THE IMPORTANCE OF algorithms today is not lost on members of the United States Congress: in September 2011, Google chairman Eric Schmidt faced a senate judiciary antitrust subcommittee that grilled him on how the company's search algorithm works and ranks products. Unfortunately, the same can no longer be said of the software engineering community.

Beautiful Code

It wasn't always this way. In the early

years of computer science, with few hardware and software resources available, an adequate algorithm for performing a task as elemental as searching or sorting could spell the difference between success and failure. Every programmer had, as part of his or her repertoire of skills, a hip-pocket collection of well-known algorithms for dispatching a variety of tasks. But enormous advances in computing power and programming environments have obscured the importance of algorithms,

one of the foundational pillars of our discipline. Today, even university curricula too often pay only lip service to the teaching of algorithmic fundamentals, reinforcing the popular belief that their place at the core of a software engineer's education is past.

Algorithms came back into the headlines recently with the announcement of a possible proof of the so-called $P = NP$ question that has stymied researchers for decades (www.nytimes.com/2010/08/17/science/17proof).

html?_r=1). Informally, *NP*-complete problems can be verified quickly, but nobody has shown yet whether they can also be computed quickly—and an answer in the affirmative could shake the very foundations of computer science (not to mention the security industry). Although the proof attempt ultimately didn't succeed, the level of

glue code, to convince programmers of the continuing relevance of algorithmic know-how to their professional development. Yet the effects of neglect are evident everywhere. Widespread and unnecessary quadratic complexities appear in industrial applications in daily use, introduced most often through carelessness or ignorance,

tables, balanced trees, and so forth. But what about massive amounts of data, whose size also keeps growing in time? In this case, traditional data structures fail in providing efficient solutions.

In "Approximating Data with the Count-Min Data Structure," Graham Cormode and S. Muthukrishnan describe sketch data structures that provide an efficient and elegant solution to the count-tracking problem by observing that, in many cases, it's reasonable to just provide a high-quality approximation.

In many areas of software engineering, there's a growing interest in exploiting the new possibilities offered by the Semantic Web. For example, requirements engineers are intrigued by the prospect of enriching their traditional natural language requirements with semantics so that they can begin to automate some of the tasks (like checks for consistency) that they must perform manually today. But there's a catch: somebody has to annotate all that text with the semantic information, and that's a big job.

In "Fast and Accurate Annotation of Short Texts with Wikipedia Pages," Paolo Ferragina and Ugo Scaiella describe a method to automatically annotate short fragments of text with concepts coming from a knowledge base we all know and contribute to: Wikipedia. This avoids the manual annotation envisioned by the Semantic Web approach, offering an opportunity for applications in information retrieval to enhance the user experience with semantic features. The innovative approach described here copes with huge amounts of text and can be applied to several languages in a very efficient and effective way, without the need for complex and expensive natural language processing techniques.

In "Developing with DBM and the Floyd-Warshall Algorithm," Lorenzo Ridi, Jacopo Torrin, and Enrico Viciario give a case-based example of

Even today, the importance of algorithms in software engineering hasn't diminished, and many domains depend on algorithmic breakthroughs in order to progress.

excitement and intense discussions it raised in the community (and the interesting role played by the Internet in opening up and mediating the exploration of the proposed proof) demonstrated that software engineers can still appreciate the importance of algorithmic advances.

And well they should: even today, the importance of algorithms in software engineering hasn't diminished, and many domains depend on algorithmic breakthroughs in order to progress (such as computer graphics, where Clark's geometry pipeline brought 3D rendering to the masses). Transportation companies such as Federal Express and the United Parcel Service (both of which must deal with *NP*-complete problems) are affected, as well as companies such as Align (which builds dental appliances and uses 3D mouth models to custom-make these devices) or Cadence (which builds tools for chip design and also must deal with *NP*-complete problems). Cell-phone manufacturers are affected, too—we need only to consider the compression algorithms for streaming music and video.

It's particularly challenging today, in a world where Web-centric software systems are assembled with snippets of

making the difference between a quick and easy refresh and an appalling user experience. Simply put, the everyday relevance of algorithms to today's practicing software engineer is clear and present.

In This Issue

With the articles in this special theme section, we hope to illustrate some of the ways in which algorithmic savvy can help the software engineer find good answers to the questions that arise in daily work: Will my application scale up or become a victim of its own success? What new, value-adding features can I offer to my customer? I'm stuck: Is there some other perspective that could help me solve my problem in a completely different way?

Consider a popular website that wants to keep track of statistics on search queries, an online retailer that needs to count the number of purchases, or an application that helps users choose safe passwords by tracking the frequency of popular online passwords. All these applications are representative of the "count-tracking" problem: a set of items with an associated (changing) frequency value. Typically, we can solve this problem by using traditional data structures, such as hash



GIUSEPPE PRENCIPE is a research fellow at the University of Pisa, Italy. His research interests include algorithms with a particular focus on parallel and distributed computing. Prencipe has a PhD in computer science from the University of Pisa. Contact him at prencipe@di.unipi.it.



CESARE ZAVATTARI is a scientific advisor at CrowdEngineering in Pisa, Italy. His research interests include text analytics, machine learning, and social network analysis. Zavattari has an MS in computer science from the University of Pisa. Contact him at cesare.zavattari@crowdengineering.com.



ALESSANDRO TOMMASI is a scientific advisor at CrowdEngineering in Pisa, Italy. His research interests include artificial intelligence, natural language processing, and knowledge representation. Tommasi has an MS in computer science from the University of Pisa. Contact him at a.tommasi@crowdengineering.com.



JOHN FAVARO is a senior consultant at Intecs SpA in Pisa, Italy. His research interests include efficient safety analysis of critical systems, real-time architectural patterns, and requirements engineering. Favaro has an MS in electrical engineering and computer science from the University of California, Berkeley. Contact him at john@favaro.net.

how knowledge of the relationships and properties of various classes of algorithms allowed them to “think laterally” when faced with an extremely practical scenario. A very engineering-related problem of scheduling jobs on an essentially embedded system was effectively abstracted and tackled in a way that was less than obvious, given the actual task. The results achieved serve as a lesson about the usefulness of the knowledge applied and essentially give us a metapattern of problem study and analysis.

Last but not least, this special theme section includes an interview with Yahoo chief architect, David Chaiken. David’s wit and wisdom are so compelling that there is no need to write a conclusion to this introduction; we gladly leave to him the last word on the importance of algorithms to today’s practitioner. ☺



Selected CS articles and columns are also available for free at <http://ComputingNow.computer.org>.

PURPOSE: The IEEE Computer Society is the world’s largest association of computing professionals and is the leading provider of technical information in the field. Visit our website at www.computer.org.

OMBUDSMAN: Email help@computer.org.

Next Board Meeting: 1–3 Feb., Long Beach, CA, USA

EXECUTIVE COMMITTEE

President: John W. Walz*

President-Elect: David Alan Grier;* **Past President:** Sorel Reisman;* **VP, Standards Activities:** Charlene (Chuck) Walrad;† **Secretary:** Andre Ivanov (2nd VP);* **VP, Educational Activities:** Elizabeth L. Burd;* **VP, Member & Geographic Activities:** Sattupathuv Sankaran;† **VP, Publications:** Tom M. Conte (1st VP);* **VP, Professional Activities:** Paul K. Joannou;* **VP, Technical & Conference Activities:** Paul R. Croll;† **Treasurer:** James W. Moore, CSDP;* **2011–2012 IEEE Division VIII Director:** Susan K. (Kathy) Land, CSDP;† **2012–2013 IEEE Division V Director:** James W. Moore, CSDP;† **2012 IEEE Division Director VIII Director-Elect:** Roger U. Fujii†

*voting member of the Board of Governors

†nonvoting member

BOARD OF GOVERNORS

Term Expiring 2012: Elizabeth L. Burd, Thomas M. Conte, Frank E. Ferrante, Jean-Luc Gaudiot, Paul K. Joannou, Luis Kun, James W. Moore, William (Bill) Pitts

Term Expiring 2013: Pierre Bourque, Dennis J. Frailey, Atsuhiko Goto, André Ivanov, Dejan S. Milojicic, Paolo Montuschi, Jane Chu Prey, Charlene (Chuck) Walrad

EXECUTIVE STAFF

Executive Director: Angela R. Burgess; **Associate Executive Director, Director, Governance:** Anne Marie Kelly; **Director, Finance & Accounting:** John Miller; **Director, Information Technology & Services:** Ray Kahn; **Director, Membership Development:** Violet S. Doan; **Director, Products & Services:** Evan Butterfield; **Director, Sales & Marketing:** Chris Jensen

COMPUTER SOCIETY OFFICES

Washington, D.C.: 2001 L St., Ste. 700, Washington, D.C. 20036-4928

Phone: +1 202 371 0101 • **Fax:** +1 202 728 9614

Email: hq.ofc@computer.org

Los Alamitos: 10662 Los Vaqueros Circle, Los Alamitos, CA 90720-1314 • **Phone:** +1 714 821 8380 • **Email:** help@computer.org

Membership & Publication Orders

Phone: +1 800 272 6657 • **Fax:** +1 714 821 4641 •

Email: help@computer.org

Asia/Pacific: Watanabe Building, 1-4-2 Minami-Aoyama, Minato-ku, Tokyo 107-0062, Japan •

Phone: +81 3 3408 3118 • **Fax:** +81 3 3408 3553 •

Email: tokyo.ofc@computer.org

IEEE OFFICERS

President: Moshe Kam; **President-Elect:** Gordon W. Day; **Past President:** Pedro A. Ray; **Secretary:** Roger D. Pollard; **Treasurer:** Harold L. Flescher; **President, Standards Association Board of Governors:** Steven M. Mills; **VP, Educational Activities:** Tariq S. Durrani; **VP, Membership & Geographic Activities:** Howard E. Michel; **VP, Publication Services & Products:** David A. Hodges; **VP, Technical Activities:** Donna L. Hudson; **IEEE Division V Director:** James W. Moore, CSDP; **IEEE Division VIII Director:** Susan K. (Kathy) Land, CSDP; **President, IEEE-USA:** Ronald G. Jensen

revised 1 Dec. 2011

