Software has been one of the fastest growing industries in most Far Eastern countries in the last few years. Japan, well-known for its tailor-made software, is now producing packaged software for foreign markets. Other countries like Singapore have initiated similar plans to capture their share of worldwide software revenues. Most Far Eastern countries now recognize that they must keep abreast with advanced software technology to compete.

To do so, these countries either import technology from the West or develop their own. As we assembled this issue, we found that cultural differences and language barriers have forced each Far Eastern country to approach the problem differently. On one hand, most have imported and integrated modern technology from the West and made it available to software professionals through some very effective national initiatives. On the other hand,
some countries have invented unique techniques to serve the needs of their peculiar software market.

As the articles in this issue document, the one common thread is that the effort to industrialize software is likely to be made through a national, government-directed, and publicly funded initiative. Typically, these initiatives seek to establish an advanced software-industry infrastructure and provide a modern software-development environment that is available to the public (including programmers at software firms, government agencies, and research institutions) via national networks. In this way, governments seek to consolidate state-of-the-art software techniques into an advanced state-of-the-practice environment and thus improve the productivity and quality of their domestic software industries.

Such national initiatives are not as popular in Western countries such as the US. We know that developed nations, with their long-standing software industries, experience tremendous inertia when they try to build a more cooperative infrastructure via an organized, national effort. It is also true that it is easier to develop and implement an effective national plan for a small country than for a large country.

Articles. The nine articles in this issue fall into three categories: invited articles from Japan, South Korea, Singapore, China, and Taiwan that report on the national status of their software industries; submitted articles that report new research from these countries; and one article from the US on the status of the software-factory concept in Japan and North America.

Japan. Laslo Belady of MCC has reported that, in his experience, Japanese managers are more aware of the importance of software than Western managers. The scope of Sigma, a five-year, $200-million project, would seem to bear him out. Noboru Akima and Fusakata Ooi examine Sigma's status and explain how Sigma can address the software crisis in Japan.

Mikio Aoyama, a software specialist from Fujitsu, has collaborated with specialists from Hitachi, NEC, and NTT on an article that describes the background and uses of one of the most widely used design representations in Japan: the tree-structured chart. These charts integrate both English and Japanese representations to enhance the overall productivity of Japanese software designers.

A third contribution from Japan addresses telecommunications software, an often-overlooked segment of the industry. Yasushi Wakakara, Yoshiaki Kakuda, Atsushi Ito, and Eiji Utsunomiya, from KDD, present an advanced environment for specifying and validating communications software requirements.

South Korea. As Chisu Wu and Yoseok Han report, South Korea could not have administered the 24th Olympic Games so smoothly without computer technology. The country's Super project was begun just in 1988 with a small budget ($5 million for the first year). But, judging from its past achievements, South Korea's national plan has many attainable goals.

Singapore. Singapore's software industry got some important exposure when it hosted the 10th International Conference on Software Engineering last year. A very young and aggressive nation, Singapore has established the National Computer Board to foster its information industry. Within NCB, the Information Technology Institute is undertaking many ambitious software projects, especially in the area of software engineering. A sister organization, the Institute of Systems Science, based at National Singapore University and associated with IBM since 1981, is working on advanced research projects such as optical-disc file-management systems. Tahn Joo Chin and Kai Yuen Wang summarize the status of both institutions.

Taiwan. A late starter, China's plan is to develop its own software technology very rapidly. Two problems we foresee with its ambitious plans are two unknowns: the effectiveness of its educational system and the willingness of young professionals to receive training in information technology.

Taiwan. Largely because of its successful educational system, Taiwan is already a serious player in the global hardware market. As Irving Ho points out, upgrad-

ing Taiwan's R&D efforts in the software industry and attracting more computer giants to become strategic allies are two proven weapons that Taiwan will try to use to improve its position in the information industry.

A second contribution from Taiwan describes a new modeling tool, the timed PQ net, which was developed at National Chiao Tung University of Taiwan. The TPQN article arose from a joint effort between researchers at NCTU (Ching-Rang Chou and Jong-Jeng Chen) and the University of Illinois at Chicago (Carl K. Chang, Yang-Fu Chang, and Lin Yang). The TPQN developed at NCTU has been very useful in modeling a real-time multi-tasking scheduler developed at UIC. These results are useful for many real-world applications.

Small world. The same software monster that Frederick Brooks has referred to faces Western and Eastern software engineers and programmers alike. While the world does seem to get smaller and smaller all the time, we anticipate with delight that this special issue will further bridge the knowledge gap between East and West. We hope it enhances your understanding of software technology in the Far East as well as enriches the literature in the most recent software advances.

Acknowledgments are deeply grateful to the advice provided by Ted Lewis, IEEE Software's editor-in-chief. We also thank the many referees who offered their precious time so unselfishly to enhance the quality of this special issue. Among them, John Chang deserves special mention for his most insightful reviews on many of the selected articles. Finally, we thank all the authors for their efforts and patience throughout the editing process.

References


Carl K. Chang is a coauthor of an article in this issue. His biography and address are on p. 51.

Mikio Aoyama is a coauthor of an article in this issue. His biography and address are on p. 37.