

The CHEMKIN Experience and Open-Source Scientific Software

Robert J. Kee

Engineering Division, Colorado School of Mines

We began to design and develop the CHEMKIN software in the late 1970s. This effort was intended only to support our own combustion research. However, as Sandia's Combustion Research Facility began to grow, an increasing number of visitors learned about our work and requested copies of the software. By the late 1980s CHEMKIN was being distributed widely and used worldwide. By the late 1990s the costs associated with distribution and maintenance had grown to the point Sandia decided to license the software to Reaction Design as a commercial venture.

CHEMKIN was designed initially to assist the incorporation of large gas-phase reaction mechanisms into combustion simulations. Although programmed in Fortran, we adopted a highly modular software architecture, which provided very great flexibility. We soon extended the modular CHEMKIN architecture to handle molecular transport, which was needed to model flame structure. As we became increasingly interested in applications like chemical vapor deposition for the manufacture of semiconductor devices, we made a major extension to accommodate heterogeneous surface reactions. Because of the flexible, modular, architecture, these new capabilities also bore fruit in applications like catalytic combustion. We continue to extend the capabilities into new areas, enabling applications in new technologies. For example, incorporation of electrochemistry is central to modeling fuel cells or other electrochemical technologies.

Some of lessons learned over some 25 years of experience with CHEMKIN include:

- Low-level, general-purpose, modular software is extremely valuable for minimizing programmer time needed to develop new applications. Source-code availability is essential to widespread enhancement and development by the general community.
- Good documentation is extremely valuable. The underlying theory can be complex, and it is essential that users and developers are able to understand the theory itself and any complexities associated with the software implementation.
- Most users are not developers. Probably fewer than 5% of CHEMKIN users actually wrote and documented substantial new applications. Therefore, high-level applications that build on the low-level tools are very important. Even in these cases, source code availability is important to enable small, but often needed, modifications for particular purposes.
- Defacto standards are greatly valuable in facilitating the exchange and cross-fertilization of scientific results. For example, communicating reaction mechanisms in CHEMKIN format has been valuable in the development and enhancement of combustion chemistry.
- Freeware or shareware is not really "free." There are real costs associated with development, distribution, and maintenance. So far, despite the obvious leverage of available software, federal agencies are less than anxious to support such efforts.
- Going forward, modular software is best written in an object-oriented language (e.g., C++). Because of reduced development and maintenance costs, and increased flexibility, CANTERA will certainly supersede CHEMKIN.
- Commercial development and support has its place, but the customer base is relatively small. It will be valuable to find middle ground where academic developers can innovate and contribute, while commercial ventures can also provide value-added be financially successful.