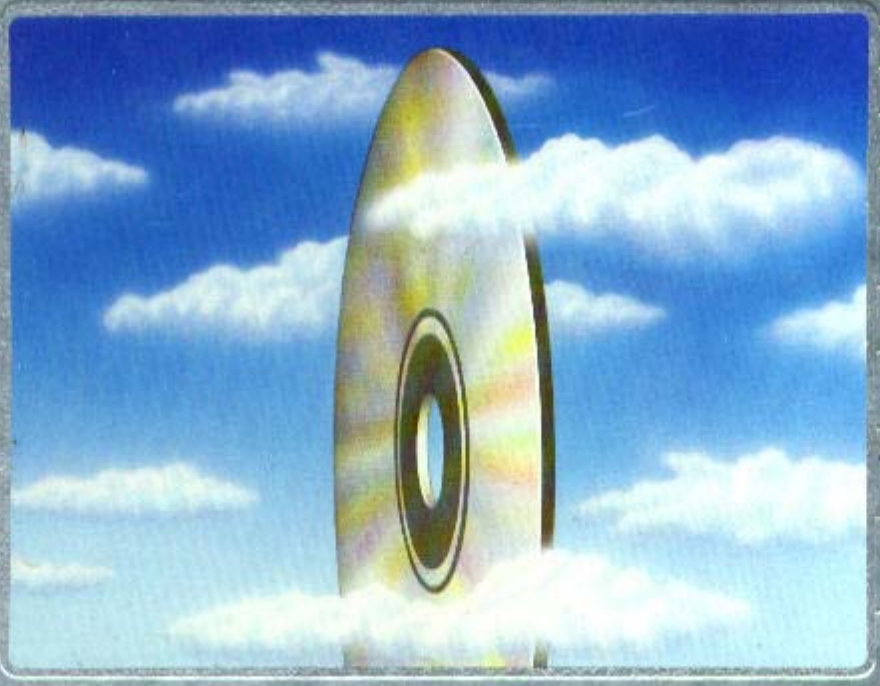
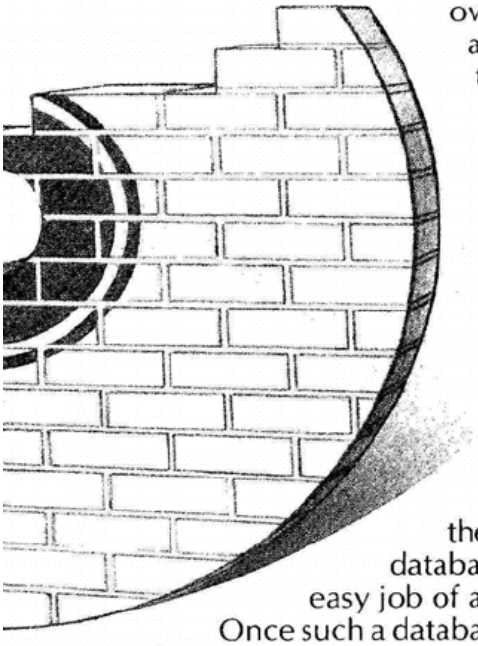


The Software Challenge

first developed. As a consequence, maintaining—that is, upgrading—old software became the single most time-consuming chore of programmers and analysts alike. Often forced to redesign a substantial portion of a software system almost from scratch, they spent as much as 80 percent of their time on these activities. With only one-fifth or so of programming resources available for new software, it is little wonder that delays in getting a project started often stretched on for years.

In 1976, Peter Chen, a professor at M.I.T., published an article entitled "Entity-Relationship Model: Toward a Unified View of Data." His paper would help change the way in which software developers viewed the business world. Chen had noted that the information an enterprise needs to operate—customer names, stock numbers, prices, personnel data, dates that orders are received and filled—remains remarkably constant. A study conducted of Washington State University's computerized records, for example, revealed that over a decade, between retiring unneeded data elements and adding new ones, the university's databases retained 80 percent of the original set. Yet during the same period of time, many new data-processing requirements emerged that combined all the data elements in different ways.



Chen proposed that systems analysts, rather than setting out to follow the crooked path of process, delve instead into the nature of the information fundamental to a user's business. The object was to understand the interrelationships between data elements, a process that Chen called data modeling. This would allow the construction of central databases containing all the information a company might wish to collate in various ways, some of them perhaps unimaginable at the time a new system is conceived. Furthermore, the databases were to be structured in a way that would make an easy job of adding new data elements or deleting outmoded ones. Once such a database had been built and loaded with information, the task of programming new uses of the information would be simplified. "Business algorithms are really very simple," says Greg Boone of the CASE Research Corporation, a market-research and consulting firm, consisting of little more than procedures for selecting appropriate information from the immense amount of data on file, sorting it, comparing it, updating it, and performing a limited repertoire of well-understood mathematical operations upon it.

LINKING FRONT END TO BACK END

The work of Chen, the academician, attracted the attention of James Martin, the computer iconoclast. Martin travels the world conducting three-day seminars on the future of data processing in the business world, serving as consultant—at the rate of $25,000 a day—to captains of industry and commerce, taping television workshops, and writing. Since 1963, he has published dozens of books about computers and computing.

Martin was born in Ashby-de-la-Zouch, a village in the English countryside bearing a name that commemorates the Norman conquest of England in the eleventh century. Shedding his "peasant roots," as he describes his early years, Martin took bachelor's and master's degrees in physics at Oxford University, and after military service he joined the British arm of International Business Machines. Sent to the United States in 1961, Martin contributed to one of the landmark programming projects of all time—SABRE, the innovative reservations network set up for American Airlines. On a later assignment, to IBM's office in London, he sold a similar system to BOAC (now British Airways), the largest order on record for that office.

That Martin, a technician and not one of the sales staff, snared the contract demonstrated a knack for explaining difficult computer science to laymen, a skill that he realized he could capitalize on. In 1977, having established a name for himself in a variety of computer specialties—communications and database management among them—Martin took a year's sabbatical from IBM to begin conducting the seminars that would make him wealthy. His income for the year exceeded that of IBM's president, John Opel. "I was surprised," he said some years later, "that it was possible to make that amount of money." He never returned to IBM.

In 1979, Martin founded a company called Database Design, Inc. Renamed KnowledgeWare a few years later, the purpose of the venture was to develop a CASE product based in part on Chen's ideas of data modeling. In a business context, the entities that Chen wrote of are records in a database of everything that plays a role in, say, manufacturing a product and selling it—sales and work orders, contracts, suppliers, invoices, and inventory, to name but a few.

In Martin's opinion, front-end CASE tools, though helpful, were far from a complete set of software-engineering programs. He envisioned an integrated system that would take the software-development process from beginning to end, starting with planning, analysis, and design, and ending with fully functioning programs. Searching for a supplier of back-end features, Martin eventually turned to an Atlanta company, Tarkenton Software, Inc. Founded by football quarterback Fran Tarkenton, the company had already developed an application generator—software that produces code from a detailed design—when sports star met computer guru in 1986. "We're both hard-driving guys," Tarkenton says, "and we hit it off right away." Tarkenton Software merged with KnowledgeWare later that year, and their collaboration bore fruit in 1988 in the form of a fully integrated CASE system called Information Engineering Workbench. Front-end analyis and back-end code generation were at last combined in a single, unified product.

By this time, others had come to appreciate the benefits of the data-modeling approach to the design of CASE systems intended for business-software development. For example, Bachman Information Systems in Cambridge, Massachusetts, had been hard at work on their version of a data-oriented, integrated CASE system called the Bachman Re-Engineering Product Set. In nearby Waltham, the Cortex Corporation had CorVision underway, also a front-end-to-back-end system—the very software that had come to the rescue of Herman Nichols and East Bay Water. And Texas Instruments had begun the design of an integrated system the company called the Information Engineering Facility, or IEF.