charting, data presentation features, and database capabilities. Lotus introduced the concept of naming cells, cell ranges, and macros. Lotus 1-2-3 became a bestseller, allowing Lotus Development Corporation to acquire Software Arts and VisiCalc, which it discontinued shortly thereafter.

Microsoft was the next big player to enter the spreadsheet market, with its product Excel. Originally developed in 1985 for the **Macintosh**, Excel used a graphical interface with a point-and-click capability. When implemented on DOS-based machines in the mid-1980s, it offered users an interface that was easier to use than the old command line standard of MS-DOS. Microsoft released the **Windows** operating system in 1987 and Excel was one of the first products available for it. By 1989, Excel was Microsoft's flagship application program.

By the end of the 1980s, most vendors in the spreadsheet arena were in court. Lotus Development filed a lawsuit against Mosaic Software and Paperback Software claiming they had infringed on Lotus 1-2-3. Software Arts, the original developer of VisiCalc, took Lotus to court saying that Lotus 1-2-3 was an infringement of VisiCalc. Lotus won the legal battles, but while these were going on, Microsoft won the market-share war, pushing Lotus 1-2-3 from its market-leading position. In 1995, IBM acquired Lotus Development, and Microsoft Excel was the clear spreadsheet market leader.

Today, spreadsheets are used for a wide variety of applications, particularly situations where 'what-if' analysis must be performed. A major feature of spreadsheet programs is that formulas can be embedded in cells (intersections of rows and columns). These formulas refer to values based on location. If a user changes a value, the implication on related values can be observed immediately. This makes the spreadsheet ideal for building a model.

FURTHER READING

Bricklin, Dan, and Bob Frankston. "VisiCalc '79." *Creative Computing,* Vol. 10, Nov. 1984, pp. 122, 124.

Mattessich, Richard. "Budgeting Models and System Simulation." *Accounting Review,* July 1961, pp. 384–397.

Mattessich, Richard, and Giuseppe Galassi. "History of the Spreadsheet: From Matrix Accounting to Budget Simulation and Computerization." *Accounting and History: A Selection of Papers Presented at the 8th World Congress of Accounting Historians.* Madrid: Asociación Española de Contabilidad y Administración de Empresas, 2000.

"VisiCalc Production Ends." *PC Magazine*, Vol. 4, 6 Aug. 1985, pp. 33.

—*Roger McHaney*

Spread Spectrum

Spread spectrum is a communication technique used in cellular telephones and radio systems. It is very resilient to noise and interference as well as to multiple-path reception of the same signal. Using spread spectrum, several users can access the same communication frequency without interfering with each other.

There are several spread spectrum techniques, but one of the easiest to implement and more commonly used is the *direct-sequence method.* Assume that we want to transmit a message coded as a sequence of signed **bits** (i.e., the sequence consists of the digits 1 and −1). The message to be sent could look like the following sequence of five signed bits: 1, 1, 1, −1, 1. However, there could be interference from another user transmitting the sequence −1, 1, 1, 1, 1, for example. The receiver gets the sum of these two signals, and since all results are positive or zero, the receiver assumes, incorrectly, that the transmitted data were 1, 1, 1, 1, 1.

In the direct-sequence spread spectrum method, each signed bit is coded using many shorter signals. The coding for a 1 could consists of 10 signed bits, which are selected randomly (e.g., the sequence 1, −1, 1, 1, −1, 1, −1, 1, −1, −1). Each time a 1 has to be transmitted, this sequence is sent; if a −1 has to be transmitted, the negative sequence is sent. At the reception end, the user multiplies the bits obtained with the known random sequence and obtains a total sum of 10 when a 1 is transmitted, and a sum of −10 when a −1 is transmitted. The purpose of the random sequence is to pull the two signals apart and add fault tolerance to the system.

Assume that there is interference in the channel. If a sequence consisting only of 1's is being sent at the same time as our signal, multiplying these 1's with the random sequence and adding the result gives zero (because there are as many 1's as −1's in the random sequence). This means that the method attenuates any signal that does not conform to the random patterns of 1's and −1's that we selected for transmission. Since the sequence is random and is known only to server and receiver, it is unlikely that noise would exhibit the same variation pattern. Also, if several users want to use the same frequency, they just get assigned different random sequences, with zero correlation. To each user, the signals of the other users look like random noise that is canceled by the system.

There are alternative spread spectrum techniques, such as *frequency hopping* (changing the frequency of transmission according to a random sequence, known only to sender and receiver), and others. These techniques are called spread spectrum because the original data are spread over different signal or transmission frequencies.

In cellular telephony the company Qualcomm pioneered spread spectrum techniques and a special technology called CDMA (code-division multiple access). *Code division* refers to the process of assigning different users different random code sequences for their data.

FURTHER READING

Peterson, Roger L., Rodger E. Ziemer, and David E. Borth. *Introduction to Spread Spectrum Communications.* Englewood Cliffs, N.J.: Prentice Hall, 1995.

Viterbi, Andrew. *CDMA: Principles of Spread Spectrum Communication.* Reading, Mass.: Addison-Wesley, 1995.

—*Raúl Rojas*

Stallman, Richard

1953–

U.S. Computer Scientist

Richard M. Stallman's major contribution to computing is as founder of and major contributor to the **GNU Project**, which was launched in 1984 to develop the free **Unix**-compatible **operating system** GNU (a self-referential acronym standing for "GNU's Not Unix"). Although Unix had been distributed freely to academic institutions before the GNU, this was the first project to emphasize the open source approach for **software** distribution as its major mission. Stallman's aim was for GNU to be completely free software. The copyright notice ensured that anyone was free to modify and redistribute the software provided that the original copyright notice remained intact. Stallman coined the term *copyleft* to describe the GNU copyright terms. Copylefted software is free software whose distribution terms do not let distributors add any additional restrictions when they redistribute or modify the software. The result is that every copy of the software, even if it has been modified, is free software.

Throughout his working life, Stallman has been based at the Massachusetts Institute of Technology (MIT), in the highly respected Artificial Intelligence Laboratory. In the 1970s he developed Emacs, an extensible editor, which can be extended using a **LISP**-like language. Xemacs, a version to work with the **X-Windows system** developed by MIT and widely adopted as the basis for windows systems on Unix-based workstations, is also available. Emacs became part of the GNU system that Stallman developed largely in the 1980s.

The **Free Software Foundation** (FSF), based in Boston, was founded by Stallman with the aim of eliminating restrictions on copying, redistribution, understanding, and modification of computer programs. Its activities include promoting the development and use of free software for all types of computing and especially in helping to develop the GNU operating system.

Stallman was the principal author of GNU C, a portable optimizing compiler that was designed to support diverse architectures and multiple languages. The compiler supports over 30 different architectures and seven programming languages. Stallman also wrote the GNU symbolic debugger (GDB), GNU Emacs extensible editor, and various other GNU programs.

The subsequent **Linux** effort, another free Unix-based operating system, initiated by **Linus Torvalds** (1969–), owes much to the GNU project. Linux-based variants of the GNU system, built on top of the kernel Linux developed by Torvalds, are in widespread use. There are probably over 10 million users of GNU/Linux systems around the world.