



ELSEVIER

Available online at [www.sciencedirect.com](http://www.sciencedirect.com)

SCIENCE @ DIRECT®

International Journal of Forecasting 22 (2006) 599–615

*international journal  
of forecasting*

[www.elsevier.com/locate/ijforecast](http://www.elsevier.com/locate/ijforecast)

# Forecasting software: Past, present and future <sup>☆</sup>

Ulrich Küsters <sup>a,\*</sup>, B.D. McCullough <sup>b</sup>, Michael Bell <sup>c</sup>

<sup>a</sup> Catholic University of Eichstätt, Department of Business Administration, Auf der Schanz 49, D-85049 Ingolstadt, Germany

<sup>b</sup> Drexel University, United States

<sup>c</sup> Paulaner Brewery, Germany

## Abstract

We present an overview of the history of forecasting software over the past 25 years, concentrating especially on the interaction between computing and software. Initially we create a framework by describing important developments of computing technology in terms of hardware and software environments. We then concentrate on two different application areas of forecasting software: (1) research oriented forecasting software often used to analyze a small number of series (for example, in market research); and (2) forecasting modules in planning environments which are often partially automated due to the large number of time series involved. Finally we make some suggestions as to where forecasting software has room for improvement. © 2006 International Institute of Forecasters. Published by Elsevier B.V. All rights reserved.

**Keywords:** Forecasting software; Demand planning; Econometric analysis systems; History of computing; Statistical software packages; Operational planning; Sales planning; Software development; Supply chain management

## 1. Introduction

Surveys of specific software packages are already plentiful (e.g., Rycroft, 1989, 1993, 1995, 1999; Yurkiewicz, 1996, 2000, 2003), and there are many articles on forecasting software. The majority of these articles describe how to apply a package to solve specific forecasting problems (e.g., Hill, Griffiths, &

Judge, 2001). A substantial number of forecasting programs have been reviewed in journals and newsletters such as the *International Journal of Forecasting*.<sup>1</sup> Some reviews provide insights into the functionality of the software (e.g., Tashman & Tashman, 1993, Küsters, 1995). Some compare several packages on a common database (e.g., Ord & Lowe, 1996 on automatic forecasting; Makridakis & Hibon, 2000 on forecasting competitions). Others compare software features without evaluation (Francis, 1981), or with evaluation (Küsters & Bell, 1999).

In this retrospective, we will not recount the reviews of specific packages. Rather we will look at

<sup>☆</sup> Our special thanks go to Robert Bell, Len Tashman, Oliver Vogt, the referees and the editor for their very helpful corrections and suggestions, and for useful discussions we thank Charlie Chase, Jerry Duvall, Eric Stellwagen and Jack Yurkiewicz.

\* Corresponding author. Tel.: +49 841 937 1846 or 1848; fax: +49 841 937 1965.

E-mail address: [ulrich.kuesters@ku-eichstaett.de](mailto:ulrich.kuesters@ku-eichstaett.de) (U. Küsters).

URL: <http://www.ku-eichstaett.de>.

<sup>1</sup> Many of these reviews can be found online at [www.forecastingeducation.com](http://www.forecastingeducation.com).

how we got from the “there and then” of mainframes to the “here and now” with PCs and workstations which are more powerful than mainframes. We will also examine the effects of computing advances on forecasting software. We can describe the current state of computing technology for forecasting in terms of the coexistence of forecasting software (which concentrates on forecasting methodology) and planning systems, such as enterprise resource planning (ERP) systems, which address the entire business process (Meta Group, 2003).

Our selective history of technological advances is presented in Section 2. In Section 3 we consider forecasting software for researchers and market analysts, a market segment requiring forecasting methodologies that are sophisticated and do not necessarily lend themselves to automation. In Section 4, we turn to forecasting software for business and operational planners, who, by contrast, often rely upon simple methods and automated routines. Some conclusions are drawn in Section 5.

## 2. A selective history of forecasting software

The development of forecasting and time series methods can be traced back to the 17th century when numbers of sunspots and price indices were analyzed by scientists. However, the practical use of statistical and econometric methods has been made possible by the invention and propagation of digital computers in the 1950s. Some of the elements we will examine are:

1. Hardware capabilities with respect to speed, memory, and disk space.
2. Hardware architectures and operating systems.
3. Hardware capabilities with respect to the user interface via general steps: Hollerith cards, IBM typewriters, ASCII and EBCDIC based character user interface (CUI) terminals and windows-based graphical user interfaces (GUIs) with a mouse.
4. Interpreters and compilers, numerical and statistical libraries.
5. Data storage software ranging from flat files up to object-oriented databases and OLAP (on line analytical processing) via hierarchical and relational database.
6. The development of transaction and planning software, ranging from dedicated material replenishment (MRP) systems via manufacturing resource planning (MRPII) to integrated approaches such as supply chain management (SCM) systems.
7. Hardware capabilities for networking including the Internet.

### 2.1. Overview of major developments in computing

In the early days, the use of computers for forecasting was severely limited by inadequate processor speed, random access memory and disk space. In the 1960s, forecasting was restricted to short and isolated series, collected in flat files, and processed by batch runs using Hollerith cards on mainframes. This was the period when programming was done in Assembler and FORTRAN under a variety of different and largely incompatible operating systems. In this stage, forecasting techniques existed mainly as subroutines.

The introduction<sup>2</sup> of OS/360 in 1967 as a scalable operating system for IBM mainframes allowed fast migration between hardware platforms, simplifying the movement and sharing of programs. OS/360 was developed by IBM (Brooks, 1975) with the intention of creating programs that would run on IBM computers of different sizes. Prior to the development of OS/360, operating systems were mainly designed for individual computer architectures (Blaauw & Brooks, 1997). With the arrival of the OS/360, software could finally be moved from one computer to another.

The introduction of UNIX in 1969 resulted in the development of portable software for smaller systems. This activity is still ongoing with LINUX and other UNIX derivatives such as BSD, AIX and Sun Solaris. The introduction of personal computers such as IBM PCs and Apple Macintoshes in the first half of the 1980s allowed for the use of computers at every desk, independent of mainframes.

Many advances in computer science influenced forecasting. The continued increase of processor speed, memory and disk space allowed the forecaster

<sup>2</sup> Much information on the history of computing systems can be found in Wikipedia, the free encyclopedia (<http://www.wikipedia.org>). This often includes announcements dates, periods of introduction and the meaning of acronyms.

to deal with larger data sets and more complex algorithms. The move from Hollerith cards to typewriter terminals, such as IBM's APL keyboard in 1968, enabled the online use of interpreters for performing analyses. Next, cathode-ray terminals, such as the IBM 3270 terminal, facilitated the design and use of interactive applications with their character user interfaces (CUI), screen reports, and full-screen graphs. This development occurred in the 1970s, when a large number of mainframe-based management information systems and manufacturing resource-planning systems appeared on the market. The subsequent step to graphical user interfaces (GUI) in the 1980s changed the software environment dramatically. While first experiences were gathered on expensive UNIX-based workstations equipped with X11, MacOS and GEM for IBM PCs, the introduction of Microsoft Windows in 1985 allowed a much larger community to use forecasting software.

Application software also changed: initially in the 1960s, forecasting methods were individually programmed using either Assembler or FORTRAN. Looking back, this becomes clear when considering the first published paper on the Holt–Winters method by Winters (1960), which included detailed flow diagrams on how to compute the forecasts. Numerical and statistical libraries, with stable and efficient optimization routines like those from IMSL and NAG, and forecasting methods became widely available in the late 1970s. This allowed the selective use of simple techniques such as smoothing, as well as complex techniques such as Box–Jenkins models. At the same time, APL interpreters on IBM mainframes became widely available (usually as time sharing systems), which allowed the interactive computation of forecasts. In this period a split between analytical environments on the one hand and planning and transaction environments on the other hand occurred. Several statistical and econometric software systems — some which are still present in the market—were developed in this period. At the same time, material replenishment systems, which focused on inventories and production, were developed independently of the statistical forecasting tools. These formed the roots of enterprise resource planning (ERP) and management information systems (MIS). This is important because this gap has never been closed entirely: only in the last few years have the data produced by ERP systems been

used as input into forecasting software, but this process is still in its infancy.

The development of databases ran in parallel. At the beginning, data was collected manually and stored on punched cards, or transferred to text files (EBCDIC/ASCII and predecessors) on disks and tapes. While this remained the primary method of data input into statistical and econometric systems for a long time, transaction and planning data were migrated to database management systems (DBMS) fairly quickly. The development moved from hierarchical and network databases in the 1970s to relational database systems in the 1980s, enhanced by object-oriented DBMS in the 1990s. The latter two resulted in the object-relational database systems used nowadays. Within business planning, however, data processing was often replaced by PC-based spreadsheets, which were stored and manipulated on local PCs. This separation of local planning data and centralized transaction data still prevails today, often resulting in problems of consistency and concurrency of the underlying database(s).<sup>3</sup> In a further step, the integration of database systems and transaction systems took place. With the immense increase of available data (e.g., by point-of-sales data), databases moved on to data warehouses, offering also a wide range of tools for extraction (on-line analytical processing, OLAP), visualization, and analysis, including predictive data mining techniques.

Despite the early development of TCP/IP and SNA as network communication protocols in the late 1960s and early 1970s, the influence of network techniques on forecasting software was, for a long time, negligible. With the exception of time sharing services and the international worldwide private networks of big companies up to the end of the 1980s, networks were usually restricted to local area networks (LAN). This, in turn, restricted the use of corporate information exchange. Flight reservation systems were a notable exception at that time.

Obviously, the rise of the internet at the beginning of the 1990s allowed the world wide exchange of information. The influence of wide area networks (WAN) was noticeable in the area of transaction

<sup>3</sup> Suppose, for example, that an item has been sold during the day, but the databases are not updated until the evening. A retrieval of data from the non-updated database will produce incorrect data.

systems such as CISC (customer information control system), SAP/R3, and others. Planning and forecasting software, however, often remained as stand-alone and small workgroup systems, which were loosely linked to other systems.

While computer development has been evolutionary, the development and use of forecasting software can be categorized into phases. Its genesis preceded the introduction of the OS/360 architecture by IBM in 1967 (Brooks, 1975), but this era was really more about computing environments than forecasting. Subsequently there have been three phases.

- Phase 1. Mainframe forecasting software until the introduction of the IBM PC in 1984.
- Phase 2. PC and workstation period, mainly single user-oriented, until 1995.
- Phase 3. Advancement of process-oriented and highly integrative software through to the present.

### 2.2. Phase 1 (1969–1984): mainframe domination

Mainframe software, either in batch or time-sharing mode, dominated forecasting software<sup>4</sup> in Phase 1. However, some very popular programs took quite a while to incorporate even rudimentary forecasting features. For example, SPSS essentially had no forecasting functionality until it added the “Trends” module in 1994 and the first editions of MINITAB offered very little in the way of forecasting. By contrast, SAS/ETS was first released in 1980.

In industry, commercial firms devoted to forecasting for industrial clients did so with mainframe computing power. Wharton Econometric Forecasting Associates (WEFA) offered the package DAMSEL, Data Resources Incorporated (DRI) clients could use EPS, and Chase Econometrics had XSIM (a commercial version of TROLL). Other mainframe packages such as AUTOBJ (forerunner of AUTOBOX), B34S, and TSP, were available not only to industry but also to academia. One industrial practitioner from this era confessed that he made his forecasts by running the same model on four different packages. If three of the four gave the same approximate answer, then that was his forecast. He took this approach because none of

the packages was benchmarked for accuracy (matters haven’t changed much in this regard in the past 25 years). Many algorithms are developed for academic purposes and, as far as scholarly journals are concerned, accuracy does not matter very much.

Prior to this time, microcomputers (PCs had yet to be invented) had been the domain of hobbyists and computer scientists, primarily due to a lack of application software. The first popular spreadsheet package, Visicalc, turned the microcomputer into a business tool and this was nothing less than a revolution for accounting and financial management. Lotus 1-2-3, released in 1982, combined a spreadsheet, presentation graphics, and simple database functionality to become the “killer application” for the PC. However, there was essentially no forecasting Phase 1. However, some very popular software that was not oriented to the mainframe. There were two primary reasons for this. First, the lack of reliable compilers: perhaps only BASIC interpreters might be used to produce such PC software. Second, mainframe packages could not be ported directly to the PC, as the PC was not nearly powerful enough. Only some parts of the mainframe package could be used on a PC version of the program, and even then it might have to be rewritten in BASIC instead of the original FORTRAN. For example, the mainframe package TSP was not ported to the PC until 1985. Instead, in 1981 Micro-TSP (TSP for the microcomputer) was written from scratch, in BASIC, for the Apple II, and was quickly ported to the PC; its successor, EViews, was released in 1994.

One very nice thing about mainframe software (and this applies to mainframe software that was ported to the PC) was that it was capital intensive. While this did limit the number of developers, it also ensured that developers had to make a sizeable investment before seeing any chance of making a return. Very often a well-known statistician or econometrician was associated with the product, and a numerical analyst as well. These circumstances gave the developer an incentive to develop quality software. The same was not true of PC software.

### 2.3. Phase 2a (1985–1989): the IBM PC and the rise of its clones

By 1985, the popular IBM PC and its clones had been around long enough that software was available.

<sup>4</sup> While there is a distinction between statistical software and forecasting software, the two sometimes overlap.

By 1989, Rycroft could compile a list of 104 software forecasting packages for the PC. Anybody with a PC and a compiler or interpreter could be in the statistical/econometric/forecasting software business. This often led to numerical problems in the software packages, as the formulas that are presented in textbooks are often not the best to use when coding software. For example, the least-squares regression coefficients are represented in textbooks as  $b = (X'X)^{-1}X'y$  but should not be computed this way (Kennedy & Gentle, 1980); yet even today it is easy to find software that computes  $b$  in this way.

Nonetheless, with a 20 MB hard drive and 640 kB of RAM, a PC was sufficiently powerful that it could compete with mainframes in a limited way. As Renfro (2004 p. 51) observed in his sweeping account of the history of econometric software, “the original PC was not the equivalent of a mainframe in processing power, but neither was it shared.” Fried (1984) recounted a relevant tale: “a PC user recently discovered that his CRAY [super] computer executed one of his applications 180 times faster than his PC. But because the CRAY was serving 100 users, the [actual] turnaround time was only a factor of 2 better than the PC.”

Around this time, universities began to move away from mainframes, setting up PC labs. PCs were good for small jobs, but for big jobs or jobs that required speed, the mainframe was still the place to go. But not for long. In industry, the situation was slightly different. Corporate IT departments, by virtue of controlling the mainframe, had long dictated computer use. The advent of the PC—loaded with Lotus 1-2-3 or Excel—produced a decentralized revolution. Any small department could purchase and use a PC, and they didn't need the IT department to approve the purchase or maintain the computer. For such departments, the PC had become powerful enough to engage in large scale forecasting, e.g., for production and inventory purposes.

However this decentralization produced problems of its own: different departments might track the same data yet maintain them differently, or use different numbers to represent the same “facts”. Thus, the databases would produce conflicting data, and different forecasts, too. The PC was at this stage just a batch engine—all it could do was produce forecasts for large numbers of items, and then write these forecasts

to a file. It was not yet powerful enough to do this work interactively.

The proliferation of standalone PCs produced its own urgent need: these PCs might solve the problems of individual departments, but they did nothing to improve the flow of information between departments. The persons making the forecasts for widget production had no idea how many widgets were in inventory. Solving this problem would be up to Enterprise Resource Planning (ERP) systems.

#### 2.4. Phase 2b (1990–1995): powerful PCs

The Intel 486 processor made its debut in April 1989, and the class of forecasting problems for which mainframes were necessary grew much smaller. Generally speaking, only for new, computationally-intensive forecasting methods (e.g., those based on the bootstrap) was a mainframe or minicomputer (e.g., VAX 11-785) a big improvement over the PC. Rycroft's (1993) update of his work from only four years earlier revealed that the number of PC packages that also had a mainframe version had declined from 46 (of 104 packages surveyed) to 28 (of 103), a rather precipitous drop. Developers were abandoning the mainframe in favor of the PC.

The sophistication of forecasting software had reached a stage where even persons with no technical training could benefit from methods that previously had required technical training. Given just a univariate time series, a program could determine which method best suited the data (e.g., exponential smoothing or ARIMA) and then optimize the parameters for the chosen method. The importance of this advance cannot be underestimated: in much the same way that control chart methods could be used by persons with no training in statistics, now these same persons could obtain very good forecasts without consulting an expert. However for these forecasts to be good, they needed good data. At this stage, each department maintained its own database and each database had to be updated individually, i.e., multiple databases could not be updated from the same source. Consequently, these databases contained conflicting information and could not serve the entire firm. ERP firms such as ORACLE, SAP and PeopleSoft jumped at the opportunity, connecting the disparate computers and databases via a client-server

architecture. However, it would be many years before this task was completed.

### 2.5. Phase 3a (1996–99): death of the mainframe?

While the mainframe had disappeared from academia, it still hung on in some industrial quarters for two reasons. First, to run legacy software. For these computers, the looming Y2K problem would sound the death knell. Firms had to decide whether to graft a “Y2K solution” onto a legacy system, or build a new system from scratch. Often, the latter choice was more cost-efficient. The construction of a new system meant that data flow would be an integrated part of the firm’s new computing system, rather than an afterthought appended to the legacy system. There was another reason for using mainframes: for transactions systems, huge mainframe systems worked well with the newly designed distributed databases, and Y2K had no impact on this use of mainframes. In either situation, opportunities for enhanced applications of statistical and forecasting methods abounded, but firms were slow to take advantage of them.

Almost as amazing as the increase in computational power over this time span was the fact that, while forecasting software had undergone a similar transformation, its basic algorithms remained unchanged. Consider the table of contents of the classic text by Makridakis et al. (1998), which first appeared in 1978 (Makridakis & Wheelwright, 1978). It bears a remarkable resemblance to the table of contents of the third edition, which appeared in 1998. In sharp contrast, a 1978 econometrics text bears little resemblance to a 1998 econometrics text.<sup>5</sup> A good argument can be made that this stability is a good thing: the same methods that worked 25 years ago still work today; can the same be said of econometric methods? So the basic forecasting algorithms are univariate time series analysis, including Box–Jenkins methods and varieties of exponential smoothing, with multiple regression (perhaps with lagged variables) thrown in for good measure. Other forecasting texts (e.g., Diebold, 2004) add vector autoregressions and

GARCH models, but the basic workhorses of forecasting remain the same. This may be largely due to the fact that users of forecasting methods tend not to be academically trained in sophisticated methods, preferring the tried-and-true methods that have always worked. Sanders and Manrodt (2003), in their survey of 240 U.S. corporations, found that the two most important features of any forecasting software were (1) ease of use and (2) easily understandable results. VAR and GARCH models defy these criteria—at least at present.

### 2.6. Phase 3b (2000–2006): PC and UNIX workstation/server domination

Y2K had largely killed the mainframe used to support legacy software, but a few remained. A recent issue of *DM Review* (Swiggett, 2005) reported that, until 2002, the grocery company “Food Lion” (1200 stores in 11 states) used “hard copy reports generated by mainframe reporting programs. These reports had to be stuffed into envelopes and then mailed to the store managers and the operations team in the field.” What is particularly telling about Swiggett’s account is the list of activities the new ERP software provides: forecasting is not one of them. Even after ERP has been implemented, with all the data flowing back and forth, forecasts are often made with Excel<sup>6</sup>, if at all (Sanders & Manrodt, 2003). Forecasting methodology has made great strides (Chatfield, 1997), but the pace at which these advances have been incorporated into software is not nearly fast enough, whether industrial software or academic software. As far as industrial software is concerned, ERP firms can’t produce forecasting software. The “Next Big Thing” is happening already: the integration of existing forecasting software with the ERP platforms.

To understand differences and similarities between forecasting systems, it can be useful to consider classes of forecasting software. For example, the *Forecasting Report* (Küsters & Bell, 1999) classifies forecasting software into five groups: forecasting systems, statistical packages, planning and forecasting systems, econometric systems, and libraries. These

<sup>5</sup> Of course, the reason may be that the complicated methods don’t work any better than the simple methods (Makridakis & Hibon, 2000), at least in the hands of non-experts.

<sup>6</sup> Excel’s statistical functionality is not up to professional standards (McCullough & Wilson, 2005).

five classes, in addition to pure planning systems, reflect the current typology of forecasting software. While such a classification was useful for the report, it does not readily lend itself to examining the genesis of the systems. Therefore this paper classifies forecasting software by usage: (1) market research and analysis; and (2) business and operational planning. We consider these categories in the next two sections.

### 3. Forecasting software developments for analysts and market researchers

#### 3.1. Analysis and market research

Forecasting for analysis and market research concentrates on the solution of specific forecasting problems, which typically do not arise regularly. Examples include:

- Forecasting the demand for a technological product as a basis for an investment decision.
- Analyzing the impact of an advertising campaign on the demand for a brand.
- Assessing country risks for a commercial bank.
- Forecasting market potential for an automotive part supplier.

In these situations, forecasters apply complex and sophisticated methods, such as multivariate modeling and Delphi processes. The forecast analyst is usually highly qualified in the field and thoroughly familiar with the methodology, research area, and computational tools. Often he/she is an expert on forecasting, econometrics or market research.

#### 3.2. Forecasting software for analysts and market researchers

Forecasting tools for analysis and market research were found in general Assembler and FORTRAN routines and libraries in the 1960s, including Call-Statpack and FAMS, developed by IBM (Brockhoff, 1977). The analyst had to write a program to access these routines, a considerable effort that probably detracted from time devoted to the problem itself. Anscombe (1981) characterizes this process in his famous book on computational statistics: “Before, say,

1970, the run was indeed likely to be long, because the available languages were rather hard to learn and remember, containing numerous arbitrary quirks and restrictions and rather clumsy to use.”

The situation improved when command line systems were developed in the 1970s, permitting statistical, econometric and forecasting procedures to be called by simple line commands. Examples are SPSS, Genstat and BMDP as statistical systems and TSP as well as RATS as time series processors. While Hollerith cards were still used to feed these programs, systems like APL from IBM emerged, replacing Hollerith cards first by typewriters and then by cathode ray tube (CRT) terminals. This new time-saver explains why APL was so popular for business analysis, not only for forecasting but also for problems such as investment calculations, insurance mathematics, and planning.

Many new statistical routines emerged from the mid-1970s to the end of the 1980s, including exponential smoothing, Box–Jenkins models, intervention and transfer function models, state space models and neural networks. Early versions of Box–Jenkins models with automatic modeling capabilities appeared, such as Autobox and SCA-Expert. Improved computing power combined with reduced prices facilitated the wide distribution and use of this software.

The introduction of PCs in the 1980s, and the development of spreadsheet software such as Visi-Calc, Lotus 1-2-3, and MS-Excel, hastened a shift from mainframe applications to PC-based designs. This led to a highly competitive market for statistical and econometric software, ranging from ported mainframe software versions to newly designed PC-based software. The introduction of the graphical user interface (GUI) further eased the work of the analyst and market researcher by replacing tedious commands with smart menus, dialog boxes and help systems. No longer did the analyst need a detailed knowledge of programming and command languages.

Even as late as the early 1990s, database interfaces were uncommon in statistical, econometric, and time series analysis packages. A notable exception was the SAS system, which introduced database interfaces for a large variety of database management systems (DBMS) starting with DB2 and SQL/DS interfaces in 1987. However, computer scientists from the research

fields of artificial intelligence (expert systems, neural networks, Bayesian networks) and database analysis (association rules) created the new field “Data Mining and Knowledge Discovery” for finding nuggets of information within huge amounts of data (Fayyad, Piatetsky-Shapiro, Smyth, & Uthurusamy, 1996). A typical example is IBM’s Intelligent Miner for Data, which has an interface to DB2 databases (Cabena, Hadjinian, Stadler, Verhes, & Zanasi, 1998). Because these systems incorporated statistical algorithms along with computational approaches, software vendors such as SPSS and S-Plus were able to enhance their statistical software with data mining capabilities, which improved their database interfaces considerably.

Along with the improved database interfaces, many systems now offer interfaces to programming languages. In S-Plus, statistical routines can be called from other software packages or from programming languages (e.g., Visual Basic for Applications) via ActiveX automation under Windows (Venables & Ripley, 2000).

### 3.3. *The future of forecasting software for analysts and market researchers*

In the early days of forecasting, there was a large gap between the development of statistical methods and their availability in commercial software. This gap has become smaller in recent years. However, the software is not perfect yet. As will be seen shortly, there remain deficiencies with regard to data handling, user interfaces, and intelligent support of forecasting processes. Unfortunately, up to now, most forecasting systems have not been organized around a workflow model with well defined processes, including cookbook-like recipes. The use of cases which define different user roles according to competence and tasks, as well as the application of process-oriented software development tools like UML (universal modeling language) may help.

One explanation for the persistence of these deficiencies may lie in the priorities of the software vendors, who concentrate on refining methodologies rather than on user support, especially if the vendor’s clients are analysts who are well acquainted with data handling and programming. Nevertheless, time pressures on analysts are growing, affording less time for data collection, data preprocessing and model devel-

opment. For example, the daily demand for beer is highly dependent on calendar effects, weather, and advertising. However sophisticated causal methods such as transfer functions and state space models are rarely applied (Hanssens, Parsons, & Schultz, 2003). There are several reasons for this.

First, the data requirements are onerous: heterogeneous data must be collected and merged from different local and remote bases. Beer sales are taken from an on-line transaction processing system such as SAP/R3, advertising effects from spreadsheets, and calendar and holiday effects from a database or handled by the software itself, and weather data are provided by meteorological offices. To accommodate the diversity of data sources, forecasting systems should include a large number of data import and export facilities such as online database connectivity (ODBC), Java database connectivity (JDBC), extensible markup language (XML) parsers, and interfaces to online analytical processing (OLAP) software such as Cognos, as well as facilities for joining queries. Ideally the forecasting system would permit access to relational database management systems (RDBMS) by SQL (structured query language) commands. It would also be useful to give analysts a standardized access to commonly used economic databases, the way EViews remotely links to DRI databases via Global Insight.

The second difficulty is that for the collected data to be used as regressors in dynamic models, they must first be coded to represent the dynamic effects. For example, the Easter holiday can have a variety of lead and lag effects on beer demand, and these effects can be parameterized either as a single effect parameter or separated by different days. The same applies to weather effects, where rules have to be defined such that beer demand *decreases* above a threshold temperature, say 30 °C. Software should offer routines that facilitate coding of dynamic effects, and possibly include an automated rule-based system to create sensible combinations of regressors.

Third, there is the modelling burden. Software can do a better job in guiding the user through the necessary modeling steps. As the analyst will seldom rely on the results of one model, he has to have a convenient way of comparing the results of different models. One elegant solution might result from treating model results hierarchically as objects in tree

view controls (Microsoft, 1995) as they are commonly used to organize subdirectories in file systems. This approach is used by tsMetrix. By “drag and drop”, the analyst could select the models and place the results in a particular window or folder. In the same way, the analyst should be able to make minor changes in the structure of the model. It is also important to be able to add comments about the data (especially calendar data, advertising data, and special situations such as outliers due to announced price increases), computed models, the modeling steps within each model and the evaluation results. Otherwise, important contextual information will be lost.

While beer demand and other established consumer goods have long time series for modeling, suppliers very commonly are faced with time series that are short (due to new and replacement items) or intermittent, with frequent periods of zero activity. From a practical viewpoint, as time series grow from new products (zero length series) to short series and then to long series, we require model switches, but such requirements are usually not supported by the software.

Anticipatory information about the future is also common. Ideally a forecaster would be supported by some kind of a rule base, which allows him to find an appropriate forecast approach. This is also the area where techniques from data mining, multivariate statistics, conjoint measurement, etc., are potentially useful. For example, pattern recognition and cluster analysis techniques can be used to find similarities and common profiles between different time series or product features.

Forecasting software should also support the analyst in his daily repetitive tasks. To do so, the system should produce presentable reports, mixing tables, graphics and comments and offer a library of report templates. It would be useful to have a tool that can generate a template on the basis of existing reports (prototype reports). An example would be the request “generate report A without component B and add component C for time series  $X$  instead of series  $Y$ ”. Interfaces to common presentation software such as MS-PowerPoint and text processors like MS-Word should exist in all packages, which is not the case today.

As noted by one referee, simulation methodologies are still rarely used. Random number generation

techniques have been used to evaluate the estimation properties of forecasting methodologies for many years. Nevertheless there are three other branches of simulation which are promising but which did not find their way into commercial forecasting software. The first is the utilization of bootstrap techniques which can be used, for example, to calculate forecast prediction intervals as in PhiCast. These techniques have great potential, especially in cases where analytical results are difficult to derive (as for forecast combinations). The second potential application area is Monte Carlo Markov Chain (MCMC) techniques as (Bayesian) substitutes for traditional estimation techniques (see, for example, West & Harrison, 1997). The third (and nearly unexplored) area is the application of risk evaluation software like Crystal Ball and @Risk which can potentially be used to integrate point, interval and distribution-forecasts into a wider framework including user defined strategies for sales and marketing and also for supply chain management including inventory control—topics which are covered in the next section.

#### 4. Forecasting software developments for business and operational planners

##### 4.1. Business and operational planning

Business planning occurs on a regular basis, often with the development of a monthly sales plan. In turn, the sales plan is the basis for marketing plans, purchase decisions and investment planning. A sales plan differs from an operational plan in that it addresses a higher level of aggregation in terms of both time and product, and is rooted in revenues rather than volume.

Business planning forecasts are usually produced on a monthly, quarterly or annual basis for product groups (instead of products), brands, and different business units such as sales regions. The maximum forecasting horizon usually ranges from 1 to 5 years.

An operational plan is the basis for production and logistic decisions. It usually includes all aspects of the company’s supply chain:

- demand plan,
- inventory plan,

- transport and distribution plan,
- replenishment plan,
- production plan,
- maintenance plan, and
- collaborative plans.

For operational planning, forecasting programs are needed to compute future demand per stock-keeping-unit (SKU) on a daily, weekly or monthly basis. In energy planning, forecasts are required by the hour and at 15 min intervals. Typically, forecasts at the SKU-level must be made for a large number of items, very often in the thousands, and these items are usually grouped into a product hierarchy, by distribution channels and by sales regions. When numerous items must be forecast on a frequent basis, the use of pre-defined or automatic forecasting techniques is critical. Forecasting systems must not only address organizational requirements for accuracy but also for processing speed, robustness and user interfaces.

The distinction between business planning and operational planning depends on the way a company organizes its planning processes. Ideally, a forecasting system should integrate both components into a mutually consistent set of plans. This is not the reality in many companies today, where we find separation more common than integration. In operational planning, the forecast horizon is short and often does not exceed 6 weeks. These forecasts affect decisions on the production levels per line (production schedule), lot sizes, transportation schedules, and the purchasing of materials for particular time periods. In business planning forecasting the number of time series is relatively low, allowing individual inspection and manipulation by the planner. This allows at least partial application of the approaches described in the last section. In contrast, in operational planning the number of time series is huge, which severely limits the possibility of individual inspection and manipulation. Thus, automatic procedures for forecast computation are necessary.

Additionally, business planners are usually less expert in forecasting methodologies than in the functional areas of the business, such as marketing, finance and accounting. In contrast, operational planners are often engineers or business administrators with a substantial knowledge of the logistical and

technical processes, but little familiarity with forecasting methodologies.

#### *4.2. Forecasting software for business and operational planners*

Although forecasting libraries in FORTRAN and Assembler were available from the 1960s, their use was tiresome, as described above. This meant that, generally, sales plans were still set up on paper. Very slowly, larger companies (e.g., ICI, see [West & Harrison, 1997](#)) began to make use of these routines for business planning. However, these routines only provided forecasts, with hardly any integration to other systems. To compute forecasts, batch runs had to be programmed and intervention in the forecasting process itself (e.g., alteration of parameters) was not possible. While an analyst concentrating on forecasting a few series could invest the time to try out different forecasting models to improve forecast accuracy over various batch runs (see above), the time involved for an operational planner with a much larger number of series was too great. Therefore, model selection within the business forecasting process was rare. As a consequence, forecasting accuracy was usually poor. It took a long time for companies to introduce forecasting for business planning and even longer for operational planning.

In pure batch processing systems, the user could not interact with the software as it executed. This deficiency was remedied by the introduction of line-oriented terminals, which allowed the system to ask the user for his input during the different processing steps. For example, a seasonal decomposition could be computed before deciding whether to apply seasonal or nonseasonal forecasting methods. A typical example is the early version of the forecasting software Sibyl/Runner ([Makridakis, Hodgson, & Wheelwright, 1974](#)). Simple menus could be designed listing the different options by numbers, so that the user could enter the number of his selection. The quality of a forecasting system at that time was highly dependent on the number of interactive branches, as well as on the modeling choices available.

With the invention of character-based user interfaces (e.g., the IBM 3270 architecture introduced in

1972, see Herrmann, Kebschull, & Spruth, 2004), the user could move the cursor all over the screen, and enter instructions without following a prescribed sequence. As common as it is today, this was the first time the forecaster could make technique selections by setting all parameters simultaneously on the same screen before invoking the forecast computation. Early versions of LOGOL (Brown, 1977, 1982) and FORSYS (Lewandowski, 1982) are two examples which used full-screen character-based user interfaces.

The first software systems that permitted business planners to interact closely with the forecasting process appeared in the 1970s. These offered simple planning features such as administration of time series, aggregation and disaggregation of series, planning screens, report generators, and functions to manipulate data and produce elementary graphs. Initially, graphs were character based, but later these were replaced by screen and presentation graphs (e.g., on IBM mainframes in the early 1980s by using the graphical data display manager GDDM).

Until the development of the PC in the 1980s, the use of forecasting software was still an unusual sport, practiced by a few large companies. But the arrival of the IBM PC ushered in a new era. The cost of computing fell sharply, stimulating the demand for software and creating an industry of software developers. Abetted by user interface architecture like the CUA (IBM's common user application) we began to see features such as overlapping menus and scrollable text boxes. By the mid-1980s, Peer Planner, Demand Solutions, and Forecast Pro had appeared on the market along with a wide variety of forecasting packages (Rycroft, 1989, 1993), some of which could partially automate the forecasting process (Tashman & Leach, 1991). However, by the end of the decade, many of these systems had disappeared from the market as they did not meet the requirements of practitioners in terms of simple usage, robustness etc.

Simultaneous progress was being made in data base programs (e.g., dBase) and spreadsheet systems (e.g., Lotus-1-2-3). These, along with rapidly increasing hardware capabilities, sparked major advances in forecasting software including functionality for parameter optimization (e.g., optimizing smoothing

constants in exponential smoothing), multi-level forecasting for product and geographic hierarchies, data and forecast overrides, and others. Forecast Pro introduced an eyeball adjustment feature allowing the forecaster to alter data as well as forecasts by adjusting observations and forecasts in time series plots (BFS, 1994). The forecaster of the 1990s now had the luxury of concentrating on the forecasting and planning problems.

Graphical facilities, interfaces to databases, spreadsheets, external data sources, numerically and statistically robust methods, and simple automatic algorithms for the selection and specification of forecasting models were now common features of business forecasting software (Küsters & Bell, 1999). Not surprisingly, awareness of forecasting software tools grew rapidly, although the majority of companies at the close of the 20th century still used spreadsheets to develop sales plans (Sanders & Manrodt, 2003).

Most recently, with the emergence of computing networks and intranets since the mid-1990s, participants in the forecasting process who were located at different sites could more readily collaborate, particularly on sales plans. Collaborative forecasting found its way into systems such as Demand Solutions, Futurmaster and Futurcast.

The acceptance of the internet and email as standards for worldwide communication nurtured collaborative forecasting in a wider sense. Forecasts developed at the customer side of the supply chain could be passed along the supply chain to suppliers and raw material providers, giving rise to CPFR (collaborative planning, forecasting and replenishment), especially within the consumer goods industries, and to VMI (vendor managed inventory) systems, where the manufacturer takes responsibility for the replenishment of the retailer's stock level.

The distinction between business planning and operational planning, never sharp in the first place, was now disappearing. Systems such as Peer Planner and Logol could be used to produce forecasts at the product level for operational planning, as well as at the product-group level for business planning. The emphasis in such systems was not on the planning process but on the forecasting engine. However, the use of methodologically sophisticated forecasting

systems strongly linked to production scheduling, transport planning, inventory and purchasing was hardly known in the past. For a long period of time the main obstacles were missing interfaces between forecasting and production planning.

The first commercial forecasting systems, like IMPACT (IBM, 1963), were purely operational forecasting and replenishment systems, providing SKU forecasts. However, product-level forecasts were needed for production scheduling and material replenishment. As a consequence, simple forecasting models were embedded in production planning systems, including BAAN, i2, Peoplesoft and SAP/R3 (Fandel, François, & Gubitz, 1997). In comparison to the business forecasting systems, these operational systems (e.g., SAP/R3, mySAP) incorporated only simple techniques such as trend curves, elements of exponential smoothing and tracking signals. Standard concepts, such as probability-based prediction intervals and out-of-sample evaluations, were hardly implemented. An overview of the forecasting features of production planning systems in the mid-1990s can be found in Fandel et al. (1997). This gap was partially closed by the end of the 1990s. SAP for example developed a module called APO (Bartsch & Bickenbach, 2002), where forecasting methods and sophisticated optimization routines augment the simpler functions included in SAP/R3. Still, sophisticated modeling such as the automatic Box–Jenkins systems as implemented in Autobox and SCA-Expert, as well as rule based forecasting (Collopy & Armstrong, 1992), have not found their way into operational planning.

#### 4.3. The future of forecasting software for business and operational planners

The majority of business and operational planners focus their attention on similar data, mainly sales figures. Sales effects at both the item and group levels have common origins, such as seasonality, trading days, and promotions, so it would seem that the same forecasting techniques could be applied. On the other hand, there are substantial differences. While a business planner concentrates on forecasting a small number of aggregated series and takes pains to provide detailed explanations and reporting, operational planners have to spread their attention across a

large number of series and frequent forecast rounds. Consequently, the operational planner can devote much less time to the specific features of the data and the forecast models, and seeks to automate the forecast process as much as feasible.<sup>7</sup>

The challenge for the future is to integrate the business and operational planning components. First, DBMS interfaces are required, just as they are by analysts. However, for rolling planning environments it is not so important to have a large number of interfaces. Instead, a stable, robust and fast interface to the transaction database or online data warehouse must be available. Obviously this process is simplified (and cheaper) if little or no interface programming is necessary. The stability and online synchronization of the forecasting database with the actual enterprise data are very important here.

Furthermore, in supply chains of consumer products bullwhip effects often occur, which can be defined as an increase in variability as fluctuations travel up the supply chain. This means that retailers directly observe the customer demand without much variation while inventory and back-order levels fluctuate considerably across their supply chain. An approach to handling this problem is the introduction of collaborative planning and forecasting replenishment (CPFR) and vendor managed inventory (VMI) systems. As these forecasting processes involve several companies of the supply chain, the software must offer a standard interface by which data can be exchanged. Some organizations have developed standards regulating the information exchange processes, as well as the data structures and contents. For example, a standard for exchanging forecasts within the German consumer goods industry has been developed by the *Centrale für Coorganisation (CCG, 2002)*. Standardization of supply chain management processes is also in progress, according to the *Supply Chain Council (2004)*, which designed the “Supply

<sup>7</sup> There is another issue which is often overlooked. As demonstrated by West and Harrison (1997, pp. 604), general factors like economic fluctuations may exhibit a stronger impact on the variability of aggregates than on the variability of their components. This implies that factors which are important for corporate-wide planning differ from factors which are important for SKU-specific planning.

Chain Operations Reference". An increasing number of SCM vendors follow this process architecture, so forecasting software vendors will also have to pay heed to it.

There is a specific problem with truncated supply which often occurs in practice. In the case of a surplus demand, most systems usually do not record the actual demand but only the actual sales, so that only the latter can be used for forecasting. In such cases all forecasting methods, with the exception of subjective approaches, generate biased forecasts which lag substantially behind actual demand. Another typical problem which sometimes occurs in practice is that sales data are recorded on the day of invoicing which often does not fall on the day of shipment - the latter being relevant for production and logistics scheduling. Such problems cannot be handled by methodological inventions but only by correct database design. Nevertheless, software and database developers should set up environments to save this information jointly with the time series to be forecasted to allow auxiliary analysis and to make the software suitable for future methodological enhancements.

Second, when incorporating special effects like advertising and calendar events, modeling is still often on a case-specific base requiring user interaction, for example by setting up a distributed lag structure for the advertising effect. Given the large number of time series in planning, some of the procedures indicated above (pre-defined effect profiles and lag specifications) need to be performed automatically and over hierarchies. Manual interaction must be limited to a small number, requiring in turn the use of some kind of effect prorating or automatic modeling.

Procedures which are prone to errors can be used if exceptions will be caught by trap mechanisms. Unfortunately there are still many systems that are not able to detect and handle numerical errors (such as overflow, denormalized operands, and insufficient data) appropriately. Furthermore there are still well-known and widely-distributed systems where the forecasting methodology is constrained to a small number of trend curves and exponential smoothing models.

While many techniques often work satisfactorily for some longer series, especially on a monthly base

for short horizons, the increased application of high frequency data (such as quarter-hour data in energy and point-of-sales data aggregated to hourly and daily figures) requires the incorporation of causal effects. Unfortunately, current systems do not include a well established but simple methodology. Consequently, most planners are forced to restrict their forecasting repertoire to techniques which do not take causal effects into account.

Most planning systems rely on internal rather than external forecast routines for application to automatic modeling. Unfortunately, internal components are often badly designed and inaccurate. While there is no direct proof of this assertion, detailed surveys of production planning systems (e.g., Fandel et al., 1997) and forecasting systems (Küsters & Bell, 1999) indicate that many systems still implement a restricted set of pre-defined models, rely on crude initialization techniques, apply rough and computationally expensive grid searches for parameter estimation, etc. Despite these problems it is often cumbersome to substitute third-party forecasting software in their place. There are several small software companies that provide excellent forecasting tools but which cannot be readily embedded into ERP systems. Some ERP vendors set up rules requiring vendor certification before they will be supported by the ERP system, and certification can be expensive. For example, SAP offers a formal SAP integration and certification program, which "guarantees an SAP proven, high quality integration of third-party products, which allow a seamless flow of business data through SAP and non-SAP components via open interfaces" (Hoeffner, 2003).

Forecasting software has been designed as stand-alone systems concentrating on model selection for obtaining accurate forecasts. The vendors invest little or nothing in the processing of the forecasts for subsequent decisions such as those involved in inventory replenishment and production scheduling. Moreover, many of them neglect interfaces to other systems. Notable exceptions are systems like ForecastPro and, to a lesser extent, Smart Forecast and Delphus, the latter with its dynamic link library PegelsDLL.

A major failing of planning systems is the lack of attention paid to the theoretical basis of modeling, and therefore to the measurement of uncertainty in the

forecasts. Without measures of uncertainty, the forecasts are not directly applicable to replenishment and scheduling strategies. If, for instance, forecast error-distributions were computed and passed on to ERP software, forecast uncertainty could be incorporated into the calculations of lot sizes and replenishment levels and intervals.

It is desirable that forecast model selection should not be based simply on forecast error metrics but also on the costs of forecast errors in terms of replenishment policies (Gardner, 2005). For instance, the frequency of out-of-stock occurrences resulting from a certain model should suggest the need to switch to a different model, as should excessive inventory costs. The necessary feedback between the forecast and the decision is not provided in today's planning systems. The problem is aggravated by the focus on point forecasts in optimization routines for production scheduling. In doing so they fail to capture the capabilities of modern forecasting methodologies to measure uncertainty.

In logistical systems, out-of-stock occurrences are often tracked by key-performance indicators (KPI). Most KPI-systems do not include appropriate indicators of forecast errors—the deviations of forecasts from actual demand. While tracking signals have been around since the 1960s (see Gardner, 1983), these metrics are more often found in planning software than in forecasting software. Sometimes statistical monitors can be found in systems with a primary focus on planning, but the majority of advanced planning systems only offer non-statistical, threshold-based alerts. From a business perspective, KPI-like out-of-stock percentages and excess stock are useful, but these statistics are rarely used for reporting forecast accuracy. For the latter, static and dynamic forecast simulations are useful, but their availability in integrated planning and forecasting systems is rare.

Forecasting and planning software often seem to follow different paths, yet, their integration should be a high priority for future software enhancements. The same applies to other application areas. Energy management systems already include forecasting subsystems (see, for example, the PSI system). However, as one of the referees pointed out, there are many other areas like automatic teller machine (ATM) cash replenishment management, and staff

management software for call centers etc., where a tight integration of forecasting modules with the business optimization module may help to improve business decisions.

## 5. Conclusions

Few forecasting software systems provide state-of-the-art functionality. Too many systems rely on outdated methods. Examples are non-optimized smoothing parameters, naïve initialization in exponential smoothing, erroneous formulas for calculating safety stocks, graphs with inadequate time scales, lack of capability for forecast adjustments, aggregation of individual item forecasts and prorating of aggregate forecasts, as well as erroneous prediction intervals. Software vendors need to upgrade to incorporate more recent methodology.

Many advances in forecasting are not incorporated into forecasting software within a reasonable time, if at all. Software developers naturally have to wait to see which new methods stand the test of time (i.e., become popular in the applied journals), but these methods are biased toward the simpler ones. If a researcher wishes to apply a new method, he must duplicate the effort of the method's inventor: he must rewrite code that already has been written once. This is not an efficient way for science to progress. Given a choice between programming a complex method and a simpler method, many researchers will choose the simpler method, since it is easier. Thus, it is the easy-to-program methods that get used in applied journals.

Those software developers who pay attention to new approaches face an additional obstacle. They see that a method has become popular in the applied literature, and then attempt to write their own code for the proposed forecasting algorithm. Often they must make educated guesses (which are sometimes wrong) about the details of the algorithm that were omitted in the article.<sup>8</sup> Moreover, testing new methods is rendered more difficult than it should be because the

<sup>8</sup> See McCullough and Renfro (1999) for the history of how Bollerslev's (1986) initial GARCH article omitted some crucial algorithmic details, resulting in different packages giving markedly different answers to the same GARCH problem.

originators of the methods were not required to create an archive of the data used. In some cases it is impossible for the developer to verify that his version of the program produces the same answer as the inventor's published results.

So while hardware processes are much faster, software advances at a snail's pace. To test a new method, the developers and researchers have to re-invent the wheel and program the code from scratch. In the age of the world-wide-web, there is no reason not to archive the data and the code used as the basis for an academic article. Consider how easy it would be for a researcher to compare two or three different methods if he did not have to program each one from scratch. See Anderson, Greene, McCullough, and Vinod (2005) for a detailed discussion of this topic.

In an operational setting, software now permits automatic forecasting and the integration of forecasts into planning. But large numbers of series are still being forecast by the crude methods contained in planning systems while opportunities to apply more sophisticated and precise techniques are not offered. So there is still much room to apply advances in statistical forecasting to current business processes.

Forecasting is very often dependent on branch, production and sales channels. So we need forecasting systems tailored to the needs of specific branches and user groups. This might require the development of branch-specific forecast and operations architectures that take into account the peculiarities of branch-specific information (data) and decision-making requirements.

We believe that progress in forecasting requires forecasting methodology to be closely linked to the available data and to the context in which business decisions are made. We require closer integration between the information offered by forecasting and the use of this information in optimization and decision-making.

Twenty-five years ago, forecasting software existed only on mainframe computers, and was used largely on aggregated data by technically-trained individuals. Now it exists on nearly every desktop and is profitably used by many who have little technical training. To the sophisticated forecaster, the methods available boggle the imagination. Yet problems persist. Then, as now, the computing accuracy of

software cannot be taken for granted (McCullough, 2001; Newbold et al., 1994). We did not know then, though we do now, what constitutes "best practices" in forecasting (Armstrong, 2001), yet these best practices have not been incorporated into software (Tashman & Hoover, 2001). Much has been accomplished, yet much remains to be done.

## References

- Anderson, R. A., Greene, W. H., McCullough, B. D., & Vinod, H. D. (2005). Data/code archives and the future of economic research. *Manuscript*. Federal Reserve Bank of St. Louis.
- Ancombe, F. (1981). *Computing in statistical science through APL*. Berlin: Springer.
- Armstrong, J. S. (2001). Standards and practices for forecasting. In J. S. Armstrong (Ed.), *Forecasting principles* (pp. 679–738). Norwell, MA: Kluwer Academic Publishers.
- Bartsch, H., & Bickenbach, P. (2002). *Supply chain management mit SAP APO*. Bonn: Galileo Press.
- BFS (1994). *Forecast Pro XE Version 2.0 manual*. Belmont, USA.
- Blaauw, G. A., & Brooks, F. P. (1997). *Computer architecture: Concepts and evolution*. Reading, Massachusetts: Addison-Wesley.
- Bollerslev, T. (1986). Generalized autoregressive conditional heteroskedasticity. *Journal of Econometrics*, 31, 307–327.
- Brockhoff, K. (1977). *Prognoseverfahren für die Unternehmensplanung*. Wiesbaden: Gabler.
- Brooks, F. P. (1975). *The mythical man-month*. Reading, Massachusetts: Addison-Wesley.
- Brown, R. G. (1977). *Materials management systems — A modular library*. New York: Wiley.
- Brown, R. G. (1982). *Advanced service parts inventory control (ASPIC)*. Thetford Center, VT, USA: MMSI.
- CCG (2002). *Der Weg zum erfolgreichen supply chain management*. Köln: Centrale für Coorganisation.
- Cabena, P., Hadjinian, P., Stadler, R., Verhes, J., & Zanasi, A. (1998). *Discovering data mining — from concept to implementation*. Upper Saddle River, NJ: Prentice Hall.
- Chatfield, C. (1997). Forecasting in the 1990s. *The Statistician*, 46(4), 461–473.
- Collopy, F., & Armstrong, J. S. (1992). Rule-based forecasting: Development and validation of an expert system approach to combining time series extrapolations. *Management Science*, 38, 1394–1414.
- Diebold, F. X. (2004). *Elements of forecasting*, 3e. Cincinnati: Southwestern Publishing.
- Fandel, G., François, P., & Gubitz, K. -M. (1997). *PPS- und integrierte betriebliche Softwaresysteme. Grundlagen, Methoden, marktanalyse*. Berlin: Springer.
- Fayyad, U. M., Piatetsky-Shapiro, G., Smyth, P., & Uthurusamy, R. (Eds.). (1996). *Advances in knowledge discovery and data mining*. Menlo Park, California: MIT Press.
- Francis, I. (1981). *Statistical software: A comparative review*. New York: North Holland.

- Fried, S. S. (1984). Evaluating 8087 performance on the IBM PC. *Byte*, 9(9), 52–56.
- Gardner, E. S. (1983). Automatic monitoring of forecast errors. *Journal of Forecasting*, 2, 18–22.
- Gardner, E. S. (2005). *Exponential smoothing: The state of the art: Part II*. Paper presented at the ISF 2005, San Antonio, TX.
- Hanssens, D. M., Parsons, L. J., & Schultz, R. L. (2003). *Market response models: Econometric and time series analysis, 2e*. Boston: Kluwer.
- Herrmann, P., Kebschull, U., & Spruth, W. G. (2004). *Einführung in z/OS und OS/390: Web-Services und Internet-Anwendungen für mainframes*. München: Oldenbourg-Verlag.
- Hill, R. C., Griffiths, W. E., & Judge, G. G. (2001). *Using EViews for undergraduate econometrics, 2e*. New York: Wiley.
- Hoeflner, J. J. (2003). *Integration of third-party software products into the My SAP Business Suite*. Foilset from SAP's internet site: <http://www.sdn.sap.com/irj/servlet/prt/portal/prtroot/docs/library/uuid/db4ebe90-0201-0010-73be-a3b887f67930>
- IBM. (1963). *Einführungsschrift IMPACT-Wirtschaftliche Lagerhaltung mit wissenschaftlichen Methoden*. IBM Deutschland, IBM Form 74899.
- Kennedy, W., & Gentle, J. (1980). *Statistical computing*. New York: Marcel-Dekker.
- Küsters, U. (1995). Software review: Autobox 3.0 and SCA PC-Expert. *The Forum (Joint Newsletter of the IABF and IIF)*, 8(2), 1–6.
- Küsters, U., & Bell, M. (1999). *The forecasting report: A comparative survey of commercial forecasting systems*. Höhenkirchen, Germany: It-Research.
- Lewandowski, R. (1982). Mit Prognoseverfahren Absatzplanung verbessern und Kosten senken. *Logistik Heute*, 6, 20–22.
- Makridakis, S., & Hibon, M. (2000). The M3-competition: results, conclusions and implications. *International Journal of Forecasting*, 16, 451.
- Makridakis, S., Hodgson, A., & Wheelwright, S. C. (1974). An interactive forecasting system. *The American Statistician*, 28(4), 153–158.
- Makridakis, S., & Wheelwright, S. (1978). *Forecasting: Methods and applications*. New York: Wiley.
- Makridakis, S., Wheelwright, S., & Hyndman, R. (1998). *Forecasting: Methods and applications, 3e*. New York: Wiley.
- McCullough, B. D. (2001). Is it safe to assume that software is accurate? *International Journal of Forecasting*, 16(3), 349–357.
- McCullough, B. D., & Renfro, C. (1999). Benchmarks and software standards: A case study of GARCH procedures. *Journal of Economic and Social Measurement*, 25(2), 59–71.
- McCullough, B. D., & Wilson, B. (2005). On the accuracy of statistical procedures in Microsoft Excel 2003. *Computational Statistics and Data Analysis*, 49(4), 1244–1252.
- Meta Group (2003). *ERP im Wandel (2002–2005): Vom Stückwerk zur Unternehmenslösung*. META Group Deutschland GmbH.
- Microsoft (1995). *The Windows interface guidelines for software design*. Redmond, Washington: Microsoft Press.
- Newbold, P., Agiakloglou, C., & Miller, J. (1994). Adventures with ARIMA software. *International Journal of Forecasting*, 10, 573–581.
- Ord, K., & Lowe, S. (1996). Automatic forecasting. *The American Statistician*, 50, 88–94.
- Renfro, C. (2004). Econometric software: The first fifty years in perspective. *Journal of Economic and Social Measurement*, 29, 9–107.
- Rycroft, R. S. (1989). Microcomputer software of interest to forecasters in comparative review. *International Journal of Forecasting*, 5, 437–462.
- Rycroft, R. S. (1993). Microcomputer software of interest to forecasters in comparative review: An update. *International Journal of Forecasting*, 9, 531–575.
- Rycroft, R. S. (1995). Software review: Forecasting software capabilities: What's hot, what's not. *The Forum (Joint Newsletter of the IABF and IIF)*, 7(2), 6–8.
- Rycroft, R. S. (1999). Microcomputer software of interest to forecasters in comparative review: Updated again. *International Journal of Forecasting*, 15, 93–120.
- Sanders, N. R., & Manrodt, K. B. (2003). Forecasting software in practice: Use, satisfaction, and performance. *Interfaces*, 33, 90–93.
- Supply Chain Council (2004). *Supply chain operations reference model (SCOR) 6.1*. [www.supply-chain.org](http://www.supply-chain.org)
- Swiggett, J. (2005). Food Lion is king of the jungle with WebFOCUS. *DM Review*, 15(6), 59.
- Tashman, L., & Hoover, J. (2001). Diffusion of forecasting principles: An assessment of forecasting software programs. In J. S. Armstrong (Ed.), *Forecasting principles* (pp. 651–676). Norwell, MA: Kluwer Academic Publishers.
- Tashman, L., & Leach, M. (1991). Automatic forecasting software: A survey and evaluation. *International Journal of Forecasting*, 7, 209–230.
- Tashman, L., & Tashman, P. A. (1993). Software review: Demand solutions: A focus on simple forecasting methods. *The Forum (Joint Newsletter of the IABF and IIF)*, 6(1), 9–11.
- Venables, W. N., & Ripley, B. D. (2000). *S-Programming*. New York: Springer.
- West, M., & Harrison, J. (1997). *Bayesian forecasting and dynamic models, 2e*. New York: Springer.
- Winters, P. R. (1960). Forecasting sales by exponentially weighted moving averages. *Management Science*, 6, 324–342.
- Yurkiewicz, J. (1996). Forecasting software survey. *OR/MS Today*, 23(6), 70–75.
- Yurkiewicz, J. (2000). Forecasting 2000. *OR/MS Today*, 27(1), 58–63.
- Yurkiewicz, J. (2003). Forecasting software survey. *OR/MS Today*, 30(1), 44–51.

**Ulrich Küsters** studied business administration and mathematics at the University of Wuppertal. In 1986 he received a Ph.D. in statistics with a thesis on latent and qualitative variable models. In 1991 he received his habilitation in statistics and system information science with a thesis on the development of rule based expert systems. His industrial activities started in 1989 when he worked on automatic data analysis systems for econometric forecasting at the IBM Science Centre in Pisa, Italy. Between 1992 and 1994 he developed sales planning and forecasting systems at the IBM Science Centre Heidelberg, Germany. Since 1994 he has been Professor of Statistics and Quantitative Methods in the Ingolstadt School of Management at the Catholic University of Eichstätt, Germany.

**B.D. McCullough** did his undergraduate work in economics at Georgetown University, and received a Ph.D. in economics from the University of Texas at Austin in 1989, where he was a University Fellow. Subsequently he taught on the economics faculty at Fordham University in New York City, and then served as a senior economist at the Federal Communications Commission in Washington, D.C. He is currently Professor of Decision Sciences at Drexel University.

**Michael Bell** studied economics in Leeds before he moved to the Catholic University of Eichstätt to study business administration with majors in production management and system information science. After obtaining his degree as a Diplom-Kaufmann in 1997 he has been employed as a research assistant at the Chair of Statistics and Quantitative Methods in the Department of Business Administration at the Catholic University of Eichstätt. Since 2001 he has worked for Paulaner, one of the biggest breweries in Germany, now being deputy head of logistics. In 2003 he received a Ph.D. with a thesis on exponential smoothing models with metric regressors.