

Evolution of ERP systems

Citation for published version (APA):

Wortmann, J. C. (1998). *Evolution of ERP systems*. (BETA publicatie : working papers; Vol. 31). Technische Universiteit Eindhoven, BETA.

Document status and date:

Published: 01/01/1998

Document Version:

Publisher's PDF, also known as Version of Record (includes final page, issue and volume numbers)

Please check the document version of this publication:

- A submitted manuscript is the version of the article upon submission and before peer-review. There can be important differences between the submitted version and the official published version of record. People interested in the research are advised to contact the author for the final version of the publication, or visit the DOI to the publisher's website.
- The final author version and the galley proof are versions of the publication after peer review.
- The final published version features the final layout of the paper including the volume, issue and page numbers.

[Link to publication](#)

General rights

Copyright and moral rights for the publications made accessible in the public portal are retained by the authors and/or other copyright owners and it is a condition of accessing publications that users recognise and abide by the legal requirements associated with these rights.

- Users may download and print one copy of any publication from the public portal for the purpose of private study or research.
- You may not further distribute the material or use it for any profit-making activity or commercial gain
- You may freely distribute the URL identifying the publication in the public portal.

If the publication is distributed under the terms of Article 25fa of the Dutch Copyright Act, indicated by the "Taverne" license above, please follow below link for the End User Agreement:

www.tue.nl/taverne

Take down policy

If you believe that this document breaches copyright please contact us at:

openaccess@tue.nl

providing details and we will investigate your claim.

Evolution of ERP systems

J.C. Wortmann
WP 31

BETA publicatie	WP 40
ISBN	90-386-0809-8
ISSN	1386-9213; WP 40
NUGI	684
Eindhoven	December 1998
Keywords	ERP; Enterprise modelling; Implementation; Component-based architecture
BETA-Research Programme	Network Management
Te publiceren in:	Strategic management of the manufacturing value chain

EVOLUTION OF ERP SYSTEMS

J.C. Wortmann

Eindhoven University of Technology

P.O. Box 513

5600 MB EINDHOVEN

The Netherlands

Tel. +(31)40 247 2290

Fax +(31)40 243 2612

E.mail j.c.wortmann@tm.tue.nl

Baan Institute

Vanenburgerallee 13

PUTTEN

The Netherlands

+(31)341 37 5569

+(31)341 37 5550

hwortmann@baan.nl

Abstract

This paper describes the development of ERP systems over time. It explores both the functionality and the technology of ERP-systems. It explains the increased importance of ERP systems in practice. The paper also addresses the difficulties encountered in implementing ERP systems in real life.

Modern ERP systems are very flexible, and can be tuned to many different business environments. This flexibility leads to increased complexity. The paper explains the nature of the flexibility and the complexity. It discusses promising ways to master complexity such as enterprise modeling.

Modern ERP systems are well suited for support of a multi-site enterprise. However, smooth cooperation of such systems in a supply chain of independent partners is still a long way to go.

Keywords

ERP, Enterprise Modelling, implemetation, component-based architecture

1 INTRODUCTION

ERP systems are standard application programs, which support execution of business processes. In the context of manufacturing, these business processes

comprise well known areas such as production planning and control, inventory control, and manufacturing execution systems. However, also business processes in other areas have become part of ERP-systems in the last ten to fifteen years. These areas are quality control, finance, human resources, product development, marketing, sales, purchasing, and many others. Section 2 of this paper gives an overview of the nature of today's ERP-systems for manufacturing and logistics.

The importance of ERP systems in practice is rapidly increasing. In every branch of industry there are many companies in a process of implementing these systems. This shows that it is possible to develop standard functionality, which can be applied in different companies and in different branches of industry. We will argue in this paper that academia should also pay more attention to ERP. The reason is, that ERP-systems represent explicit best-of-breed knowledge in many fields of industry. Moreover, these systems rapidly convey this best of breed knowledge over the earth. Both the knowledge itself and the way in which it is represented are interesting areas of scientific investigation. Section 3 discusses the importance of ERP-systems in more detail.

The rapid increase in functionality is not the only reason for the success of ERP-systems in general or for some ERP-systems in particular. New technologies play a role which is just as important. For example, the emerging RDBMS (relational database management systems) in the eighties represents such new technology. The same holds for "open" operating systems and for graphical user interfacing (GUI) techniques. Section 4 explains the role of the technology in the advent of ERP. More specifically, the consequences of technology for architectures will be explained.

Despite the success of ERP systems in terms of market growth, there are still considerable problems in implementing ERP systems. Reasons for this stem not only from the complexity of the systems, but also from the fact that implementations often lead to substantial business reorganizations. This point is discussed in Section 5. In addition, the role of enterprise modeling methods is discussed (Section 6).

An overview of ERP evolution should also contain a window in the future. This paper will argue in Section 7 that ERP systems will move towards configurations of loosely coupled systems. Consequently, future ERP systems will be less monolithic than current systems and these systems will show more interoperability than current systems.

Of course, there is a risk in predicting the future: unfortunately, the future is more difficult to predict than the past. Therefore, the discussion on this issue has to be more prudent. However, there are quite some arguments for the view presented here, both from practice and from the academic world.

Finally, Section 8 presents some conclusions.

2 WHAT IS ERP?

The sixties: algorithmic decision support: scheduling and MRP

The development of Enterprise Resource Planning systems started with simple inventory control applications. Systems such as IBM's IMPACT from 1960 were intended to control stock levels for a large number of items. These systems calculated forecasts of future demand through advanced algorithms. Based on these forecasting numbers, the applications determined ordering parameters such as safety stocks, reorder points and lot sizes. Generally speaking, the computer power was used for performing **calculations** in these early days.

Somewhat later, around 1965, the first planning and scheduling applications came into being. Again, an IBM package (CLASS) is a good example. Again, the focus of these packages was on advanced **calculations**, perhaps following the success of linear programming in the decade before.

At the end of the sixties, the concept of **Material Requirements Planning (MRP)** was invented (see Orlicky (1975) or Fogarty et al. (1991)). This concept was based on two principles. First, it relied on the consequent usage of **time-phased** information in calculating orders. The well-known time-phased representations of MRP is based on time-phased representation of all material requirements and time-phased representation of scheduled receipts. The MRP netting and lot-sizing logic leads then to a time-phased plan for releasing orders. This systematic use of time-phased representations was almost impossible in earlier, manual systems. Therefore, MRP-calculations employed the computer's calculation power and memory.

The second principle behind MRP is the clever employment of the **Bill-of-Material (BoM)**. The BoM represents the way in which subsequent manufacturing stages are passed when transforming raw materials into finished products. The BoM was used to make sure that the material planning calculations were not iterative, and that they used the computer power in an efficient way.

All these early applications have in common, that they use IT for algorithmic calculations in a way which is focussed on efficiency. It should be noted that transaction processing was still largely manual in the sixties. These activities were based on **sorted paper files** with entities such as orders, items with inventory data, or accounts receivables. The computer was mainly used to perform larger sorting operations with input based on punched cards. Gradually, more functionality was added, but the roles of various departments (Sales, Production, Purchasing, Finance, Quality, HRM, etc.) remained unchanged.

The seventies: online support of business processes and integrated databases: MRP II

The seventies brought a substantial change, because integrated support of business processes became possible. Writers such as Blumenthal (1969) made a plea for architectures which would combine decision support, transaction processing, and management information applications into an integrated whole. This vision slowly became reality, due to two technological innovations.

The first innovation was the advent of **online processing** via VCR-terminals. Online processing did improve data-entry efficiency because the human aspect of data-entry improved and the technical aspect improved. Therefore, enterprise applications such as sales order entry, forwarding, ordering and invoicing became possible.

The second innovation has been the breakthrough of standard packages for database management. Databases are important because they allow applications to cross the boundaries of functional areas. For example, at the moment of entering a sales order, reservations for delivery are booked in the logistics area and future receivables are booked in the financial accounting system. Such automatic bookings were quite difficult in earlier times when each department in a company developed their own systems.

The combination of online transaction processing and modern database management systems created **business information systems**. These systems paved the way for BPR, because they crossed functional boundaries. But at the same time these systems did not enforce BPR, because many systems grew more or less harmonious by ad-hoc integration of functional applications.

In manufacturing, these business information systems became known as **Manufacturing Resources Planning (MRP II)** systems. This name was well chosen, because it indicated the continuity from MRP to MRP II. There is indeed a proper extension in decision support for manufacturing of standardized products, because due attention is paid to capacity planning and master scheduling in MRP II. This is well known in the community of practitioners and scientists in Production Planning and Control. But from a perspective of IT applications, the difference lies not so much in expanded decision support. Rather, the innovation is support of business processes in an integrated way.

In the late seventies and during the early eighties, MRP II was almost synonymous to production planning and control, at least in the USA. Japanese practitioners, notably from Toyota (see Shingo (1985)), and European academics such as Burbidge (1989) criticised the MRP concept for planning and control, but that is not the point here. From an IT-point of view, the issue is that **standard software packages** could be designed.

This fact is both surprising and trivial. It is surprising on the one hand, because many people cannot believe that such different industries as automotive and furniture could be governed by the same business process types. The MRP-crusade, however, changed this mindset in the USA. Packages as COPICS (for IBM-mainframes) and MAPICS (for IBM-minicomputers) and many others claimed universal applicability for their support of business processes with reference to MRP II.

On the other hand, it is trivial that standard software would arrive, because the most characteristic property of software is the fact that the variable costs of another copy are zero. This explains the early success of MRP II packages.

The eighties: more functionality in standard software for client-server architectures

Standard software can be expanded in functionality, because development investments are paid back by many customers. And the need for more functionality is almost endless. Therefore, MRP-II packages rapidly expanded with functionality for new areas of manufacturing businesses, such as fixed assets management, shop-floor scheduling and control, distribution, transportation, service, financial services, EDI, tracking and tracing, product data management, etc. (N.B. Although quite a few packages were rooted in MRP in the eighties, there were also packages which came from financial applications, quality control applications, or other areas. For the purpose of simplicity, we will not expand on these other fields).

The increase in functionality in standard software packages leads to an increase in the number of parameters needed to “tune” the package to a particular environment.

The need to increase functionality created more complexity than most architectures could bear at the end of the eighties. For example, the need to support **customer-driven** manufacturing was not easily incorporated in MRP systems (see Wortmann et al. (1997)). Similarly, requirements from process industries (see Van Rijn, Schyns et al. (1993)) or repetitive manufacturing (see Shingo (1985)) are not easy to incorporate. But especially the integration of logistics solutions over several plants and warehouses within a company increased the requirements. These **multi-site** applications caused a dramatic increase in complexity. This is due to the fact that in multi-site applications almost every activity can be either implemented centrally or locally per site. Apart from that, every local activity can be implemented differently in different sites. To cover all these possibilities, the number of parameters increased drastically. This is one of the causes of long implementation times. This is elaborated in Section 5.

In parallel, there was a substantial shift in technology. In hardware, the dominant position of mainframes was not only threatened by mini-computers, but especially by **two-tier client-server (CS) architectures**. These consist of networks of computers, usually with one or more database-servers in the heart and application clients connected. These architectures allowed downsizing of mainframes, while reducing costs of investment and operation.

Closely related, there was a change in operating systems technology towards **open systems**. Especially the UNIX operating system was suitable to run on many different types of hardware. It created a market for application software independent of hardware, which in turn propelled the penetration of standard software with a CS architecture.

The combination of the CS architecture with multi-site applications opens the possibility to operate the main business processes both in a centralized and in a decentralized way. Partitions of the database can become locally owned. This makes it possible to implement systems in a site per site sequence. Usually, these local databases should be interpreted as parts of conceptually integrated whole. Conceptually, the client-server solution is still a solution based on one single

database schema with an explicit representation of the multi-site schema in this database.

The Nineties: GUI, Windows, Enterprise Modeling and E-commerce

Although the requirements for more functionality remained a pressure on ERP-vendors, the most eye-catching developments have not been in functionality. Rather, the outside look and feel of ERP-packages have changed considerably. Closely related, the architecture was brought up to date. Finally, tools were added to master the complexity of design, implementation and maintenance.

The requirements for a modern graphical user interface was under UNIX guided by the OSF-Motif standard in the early nineties. However, the market dominance Microsoft soon forced ERP vendors to release GUIs which are compliant with Microsoft Windows. The pressure for a modern GUI leads to a so-called **three-tier client-server architecture**. In this architecture, the database server(s) are connected to application clients in the same way as in the two-tier CS architecture. However, these application clients play the role of an application server for user-interaction programs with GUIs running on a PC. There are two variants of this architecture. The so-called **fat client architecture** takes considerable functionality to the PC. The so-called **thin client architecture** takes only the immediate user interaction to the PC, and leaves all application functionality with the application server.

The architecture of ERP systems was furthermore influenced by the requirement for **workflow** functionality. In the office world outside ERP, the notion of workflow functionality was quite well received. Roughly speaking, workflow functionality offers the possibility to monitor the flow of cases in an office in much the same way as the flow of **work orders** may be followed on the shop floor of a factory. Furthermore, workflow tools enable customers to define the activities to be performed for a "case" in an explicit and changeable way.

The growth of ERP functionality as described above is mainly a growth in support of the office world. Consequently, workflow management is a logical requirement. Moreover, quite some functionality of ERP systems is in fact a means to create flexibility in the activities to be performed – just as in workflow systems. Therefore, ERP vendors created workflow solutions in their systems. These workflow solutions have the ability to become an independent layer of middleware to relieve the application programmer from the necessity to foresee and define all work flows. In this way, workflow functionality can be compared to the functionality offered by a DBMS.

Work flows can also be interpreted as an explicit way to represent business processes. As we will explain in Section 5, implementation of ERP systems requires often an explicit documentation of business processes and organizational structures. Therefore, ERP systems and **enterprise modeling** tools become closely connected.

Finally, the nineties show the emerging role of internet and electronic commerce. Electronic commerce requires from ERP vendors not only, that applications are being internet enabled, but should also redesign these applications for usage by

remote customers. Architectures with **thin clients** have an advantage here. We will argue in Section 6, that the future of ERP will be reshaped by the requirements of connectivity.

3 WILL ERP REMAIN IMPORTANT?

As argued in the introduction, ERP systems seem to be everywhere, nowadays. Why? First of all, standard software is indeed an inexpensive solution for companies, when compared with maintenance of home-grown information systems. Stated differently, maintenance of these home-grown systems is extremely expensive. There are basically two reasons for this.

On the one hand, new technologies emerge, which should be incorporated in information systems if companies want to remain in business. No mature company can nowadays afford to handle business processes such as customer order entry in a manual way. The past was simply too time-consuming and inefficient. Similarly, office automation is today self-evident, just as a telephone. The next section discusses the role of new technologies.

On the other hand, standard software incorporates best of breed solutions for a broad spectrum of business processes. Therefore, adopting a standard software solution is a relatively cheap way to convey professional knowledge to all corners of an organization. Re-inventing the wheel is avoided.

Secondly, adopting standard software packages may lead to a kind of rationalization. It paves the way for BPR and it is an efficient means to implement BPR. It paves the way for BPR because implementation of ERP requires close examination of many business processes, which is anyway necessary for a BPR project. And even without a formal BPR project, an ERP implementation often leads to less steps in business processes and less walls in functional organizations (see Champy (1997)).

Thirdly, ERP systems can be used in close connection to quality drives, such as ISO 9000. Because of the fact that all business processes have to be documented, there is a logical connection here. But this point can be exploited much further, if appropriate enterprise modeling tools are employed. We will return to this point in Section 5.

Is ERP all-embracing? Certainly not. In the original field of algorithmic decision support, there are meanwhile **advanced planning systems** for shop floor control and supply chain planning which are superimposed on ERP. In office environments for professionals, such as in engineering departments, there are dedicated ICT-solutions such as CAD and PDM. For collaborative work, groupware is emerging. For sales representatives, **customer interaction software** becomes mature. And so on.

Is ERP future proof? That is, will an investment in ERP last for some five or ten years? The answer is probably “yes” for internal processes. For inter-organizational collaboration there is more doubt. The scenario sketched in Section 6 will perhaps turn the doubt into a prudent “yes”.

Is ERP indeed cheap and flexible? The software itself is cheap, no doubt. The costs of implementation are considerable, as will be discussed in Section 5. There we will argue that the software-as-delivered is flexible, but the software-as-implemented is not.

Will ERP remain important? Of course! ERP represents the backbone of business processes and databases. Future generations cannot envisage an organization without it!

4 THE ROLE OF TECHNOLOGY

In the above discussion, the role various technological “waves” has already been touched at several places. In this section, we will highlight this role and discuss the current technology wave in more depth. Technology changes at different levels. Roughly speaking, the levels are: hardware, operating systems and communication standards, middleware, and tools for application development.

Hardware

The advent of online processing in the seventies created a revolution in business information systems, as described in Section 2. Similar roles were played later by minicomputers, PCs, and high-speed communication networks. Other hardware innovations such as barcoding, mobile phones, and LANs were not described, but had in related fields considerable impact.

Operating systems and communication standards

Again, the role of open operating systems such as Unix has been considerable in the past. They created a simple de-facto standard which made application programs portable. Windows NT plays the same role right nowadays. Communication standards such as Corba or DCOM have a similar effect on interfaces between application programs.

Middleware

The term “middleware” is a collective name for database management systems, communication software (e.g. TCP/IP), security management, GUI-drivers, hardware monitoring and optimization tools, and many other pieces of software which can be used by application programs at runtime. In section 2, we have seen that databases and GUIs had a major impact on ERP systems. Workflow technology is another example.

Application development tools

Probably the most important technologies are tools, which simplify the life of the application programmer, the so-called lower case tools. These tools comprise higher level programming languages and code generators, libraries with utilities,

debuggers, documentation and testing tools, reusable services, repository management, etc.

All these technologies have had considerable impact on ERP systems. In fact, one of the major reasons for productivity improvement through ERP is the thorough use of new technologies. This is also one of the reasons why home-grown systems cannot be maintained, as was argued in the previous section.

Moreover, the visionary and innovative ERP-vendors are the vendors who adopt new technologies steadily and in close relation to their maturity stage. When a new technology emerges, it should be incorporated in the ERP-product. However, when standards are established and mature specialized components are offered, the ERP vendors should also be prepared to buy the new technology rather than to continue building components by themselves. No doubt internet and object-orientation will have considerable impact in the near future.

5 THE DIFFICULTY OF IMPLEMENTATION

Despite of the growth of ERP license sales, the implementation of ERP systems is often a cumbersome process, especially in large organizations. Typically, the expenses covering the software and hardware together are less than the expenses for consultancy and support, in larger implementations. The costs for training are also considerable. There are many reasons for this.

First of all, there is the complexity of the ERP software. As has been explained in Section 2, there are many parameters to be initiated when implementing standard software. There are thousands of tables to be initiated before an implementation of a large system is finalized (Bancroft e.a. (1997)).

*More specifically, the flexibility of the software allows many different ways to support business processes. But the reverse side of flexibility is complexity. Thousands of parameters are extremely difficult to master. Therefore, several approaches towards support of parameter setting have been propagated. These approaches range from the use of **artificial intelligence** (expert systems, case-based reasoning) via the use of **templates** (specific parameter settings for certain vertical markets) to the use of **enterprise modeling tools** for setting parameters. Enterprise modeling approaches typically describe business processes and control structures in various levels of detail. The idea is, that parameter settings can be linked to these descriptions ("models"). In the next section we will therefore elaborate on enterprise modeling in the context of implementation.

Secondly, the implementation of ERP requires a considerable amount of training. End users need intensive training on a limited amount of functionality, but key-users and others (consultants!) need considerable training in many aspects of packaged software.

Thirdly, the implementation of ERP is often combined with BPR (see Section 3). In today's world this means the functional organization is partly replaced by an organization which is more oriented towards the company's work flows. The

change from the so-called **functional silos** in organizations towards business units often goes together with ERP implementation (see Appleton (1997)). It is currently not clear whether this organizational change is a major cost driver, which will ultimately disappear.

Fourthly, an important cost driver of ERP implementations is the fact that **future adaptations** are not easy. Therefore, some vendors provide in an enterprise modeling systems tools for describing planned changes to the implementation (see next Section). However, the flexibility of the initial implementation is not easily maintained when a system is up and running. Many alternatives which are an option in the initial implementation cannot be seen as an option afterwards.

One of the reasons is, that a new setting of parameters may lead to a different data-structure diagram. This requires specific conversion programs to be written, in order to be able to continue the use of historic data. But a more important reason is the fact, that different **versions** of related application programs in an ERP-suite do not cooperate. Even within the same ERP-suite, all users of application programs have to use the same version of whatever application. This will become unacceptable in the near future.

6 ENTERPRISE MODELING

From the early days of Computer Integrated Manufacturing it has been argued, that **modeling** of business processes and ICT-solutions is the only way to master complexity. The term **modeling** has many meanings. But in the context of our discussion, techniques such as IDEF or SADT were forerunners of efforts such as CIMOSA (1993) or ARIS (See Scheer (1994)).

The basic idea of enterprise modeling is a combination of powerful expression of business requirements with precision. **Powerful expression** of business requirements has been pursued by powerful description of business processes and resources. For example, business processes can be described at various levels of detail and can be depicted graphically. Resources can be modeled by complex data structures. Precision has been obtained by formalisms such as Petri-nets or UML as object modeling language. However, the objective to generate code from precise descriptions is not yet practice.

The aim of **enterprise reference architectures** (See Williams et al. (1994), CEN/CELENEC (1994)) is to support not only descriptions of ICT-systems in their business context, but also to support the design process as such. In the context of ERP-packages, reference architectures require adaptation because the business knowledge to be specified, is already available in the package. Therefore, the nature of enterprise modeling changes from a one-of-a-kind activity towards a modify-the-template activity.

Enterprise modeling tools such as the ARIS tool set (Scheer,1994) or Baan's Dynamic Enterprise Modeler (Van Es and Post (1996)) are contributing considerably to implementation effort of ERP systems. For the first time, enterprise modeling is being applied at large scale, and experience is captured through

reference or **template** models. Baan's DEM can be integrated with the Baan ERP-suite, thus approaching the idea of executable models. However, the integration of enterprise models with an ERP-suite has also its drawback. In particular, current template models have to be updated for a new version of the software.

Furthermore, the current modeling constructs are focussed on **enterprise** modeling, and therefore they have an inherent weakness: these modeling languages provide no support in modeling the functionality and quality of applications, nor the properties of ICT-infrastructures.

Modeling an ERP-solution in the future will require not only modeling of the business environment, but also modeling of the functionality of an application. More specifically, if components from different versions of an ERP-suite have to work together, then these components need **models** of the functionality of their partners in the same band.

Users have wishes reaching further than this. The user community of ERP software will expect close cooperation between components from different vendors. This requirement asks much more, viz. that components from one ERP suite can cooperate with components from another ERP suite. This requirement will also be enforced by cooperation within supply chains or virtual organizations. Modeling functionality is required, but not enough. In addition, properties such as reliability, portability, response time, scaleability should be modeled and communicated.

7 A WINDOW IN THE FUTURE

Client-server architectures are essentially monolithic. There is a central database, with a common conceptual schema for all applications. Even in the case of so-called distributed databases, the **conceptual** schema remains monolithic. ERP-systems cannot remain like this, for several reasons.

First of all, larger organizations cannot be forced to use the same version of the software everywhere. An automotive assembler with hundreds of factories will never obtain a situation where all factories use the same software.

Secondly, internet will force cooperation and coupling between different versions of the same vendor, but also between versions of different vendors.

Thirdly, intranet technology and componentization of software in other areas will teach users to go for "best of breed" software.

But the most fundamental reason for **conceptually** distributed and loosely coupled systems is the necessity for change. Change in software requires change in datastructures, and therefore a monolithical conceptual database is highly unlikely.

Actually, replication mechanisms such as new available in systems such as Lotus Notes or Microsoft Outlook will probably become the usual way of working in ERP-systems and PDM-systems. Large applications will be broken up into components which run asynchronously and replicate essential data periodically. Shop floor automation has always led to a massive amount of data which in an

aggregated form finds it's ways to towards databases. Here too, an asynchronous replication scheme is used.

Loosely coupled systems constitute a new area of research from a technical point of view. But also the fact that future enterprise systems have to be **configurable** constitutes an enormous challenge. It means that **enterprise modeling** needs renewed interest, but in combination with modeling of application's functionality and with other properties of ICT-components.

Loosely coupled, component-based systems require an architecture of interfaces, which will remain much longer valid than the individual components. This architecture will be based on a decomposition of the industrial engineer's view of the world. This architecture should be sufficient to enable workflow and enterprise modeling in the context of production management. It requires probably better insight in the role of the subsystems and business processes than we have today.

8 CONCLUSION

Enterprise Resource Planning is causing quite some renewal in production and logistics. It serves more and more as a transaction processing backbone for other applications, and is indispensable for modern financial accounting, quality control, and operations management.

ERP systems become more and more a "must" in modern business world. Nevertheless, implementing ERP is not easy. The costs and expenses are considerable. Training effort are often underestimated. ERP systems represent the "best of breed" solutions nowadays available. As such, they may also become a hindrance in distributing business knowledge.

Implementing ERP systems should be closely connected to continuous improvement of business processes. In today's situation, this is not yet the case. However, enterprise modeling techniques seem promising in attaining a goal of ongoing optimization.

Future systems will be based on loosely coupled component architectures. Such architectures are more robust in case of change. They require additional modeling capabilities of the functionality and quality of ICT components in relation to the business processes and other aspects of the functioning of enterprises. Internet will only increase the speed of such interoperable multi-vendor component-based systems.

9 REFERENCES

- Appleton, E.L. *How to survive ERP*. Datamation 43 (3), pp. 50-58, 1997.
 Bancroft, N.H., H.Seip, and A. Sprengel *Implementing SAP R/3:How to introduce a large system in a large organization* (2nd ed.). Manning, Greenwich, CT, 1997.

- Blumenthal, S.C. *Management Information Systems: A Framework for Planning and Development*. Prentice-Hall, Englewood-Cliffs, N.J., 1969.
- Burbidge, J.L. *Production Flow Analysis*. Oxford, 1989.
- Champy, J. *Packaged Systems: One Way to Force Change*. Computerworld 61 (1997).
- CEN/CENELEC WG ARC(1994) Evaluation report of constructs for views according to ENV 40 003, *Computers in Industry* 24/2-3, 159-236.
- ESPRIT consortium AMICE (Eds.) *CIMOSA: Open System Architecture for CIM*. Springer, Berlin, 1993.
- Fogarty, D.W., Blackstone jr, J.H., and Hoffmann, T.R.. *Production & Inventory Management*, 2nd ed. South-Western/APICS, Cincinnati, Ohio, 1991.
- Orlicky, J. *Material Requirements Planning*. McGraw-Hill, 1975.
- Scheer, A.-W. *Business Process Engineering* (2nd ed.) Springer, Berlin, 1994.
- Shingo, S. *Study of Toyota Production System from industrial engineering viewpoint*, Tokyo, Japan Management Association, 1985.
- Van Es, R.M. and Post, H.A. (1996). *Dynamic Enterprise Modeling: linking business and IT*. Kluwer, Deventer, The Netherlands.
- Van Rijn, T.J.M., Schyns, B.V.P., et al *MRP in Process – the applicability of MRP-II in the semi-process industry*. Van Gorcum, Assen (Netherlands), 1993.
- T.J. Williams, Bernus, P., Brosvic, J., Chen, D., Doumeingts, G., Nemes, L., Nevins, J.L., Vallespir, B., Vlietstra, J., and Zoetekouw, D. (1994). Architectures for integrating manufacturing activities and enterprises. *Computers in Industry* 24/2-3, 111-140
- Wortmann, J.C., Muntslag, D.R., and Timmermans, P.J.M. *Customer Driven Manufacturing* Chapman and Hall, London, 1997.

10 BIOGRAPHY

Hans Wortmann studied industrial engineering and management science at Eindhoven University of Technology. His PhD work described shop floor control systems in semiconductor manufacturing. He specialized in information systems for production and logistics, and was active in several ESPRIT projects. He is editor-in-chief of *Computers in Industry*. He has a long collaboration with Baan Company. Since 1997, he is chief scientific officer of Baan Institute, while still being a professor at Eindhoven University.