Developing Complex Information Systems from Standard Software Components: A Case Study within Manufacturing Resource Planning (MRP II)

Marc Braun
Computer Science Research Group B
University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
Marc.Braun@informatik.uni-erlangen.de

Sybille Möhle
Computer Science Research Group B
University of Erlangen-Nuremberg
Martensstraße 3, 91058 Erlangen, Germany
Moehle@informatik.uni-erlangen.de

Abstract
At our research institute, we explored to what extent it is possible to build complex applications with standard software. Using Microsoft Office packages we established an MRP II-system which contains all important modules. For small to medium-sized enterprises (SME) it might be an alternative to today's planning systems, most of which focusing on organizing and supervising complex manufacturing processes with many options for tuning and adaptation. In this aspect our system is lean and particularly well suited for SME. We describe our engineering concept and present the architecture of our prototype. A balance of standard functions which were used, on the one hand, and functions which had to programmed individually, on the other hand, is discussed. Furthermore, we summarize strengths and weaknesses of our approach, including results of performance tests and adaptations to specific business processes. Finally, we specify the concept of a component configurator which composes an MRP II-system from enterprise-type-specific building blocks.

1 Introduction
In many enterprises, especially in small and medium-sized ones, uncomplicated production processes due to the decentralization of manufacturing have been established. For planning such production processes we also need appropriate manufacturing resource planning (MRP II) systems. Most planning systems of today, however, are not suited for a simplified resource planning: their focus is on organizing and supervising complex manufacturing processes with many options for tuning and adaptation. Configuring these to the planning processes of SME is expensive both in cost and time. In consequence, systems for manufacturing resource planning should also be simplified: with services tailored to the needs of uncomplicated production processes, and composed of standard off-the-shelf software components.

2 The componentware idea
A major problem of adapting standard software systems to specific business processes lies in high parameterization efforts (see e.g. [1], [2], [3]). In addition, many SME only exploit a small part of the whole functionality that available standard systems offer, due to the fact that they are designed to cover a wide range of business processes. Most enterprises expect systems that combine the advantage of both individual and standard software, but which avoid - at the same time - their weaknesses. Therefore new approaches aim at creating applications from universal, standardized fragments on the one hand and from generic, adaptable parts on the other.

Specialized software development companies have started building domain-specific components and business objects on top of standard and quasi-standard platforms [4], such as CORBA (see e.g. [5], [6], [7]) and OLE [8]. Such components and business objects can be composed in a simple way to form applications that meet the needs of specific customers: the underlying platforms provide interfaces for an easy-to-use plug-and-play environment. Furthermore, you can reuse these components in more than one application, for example if they represent business objects that are needed in almost every application for a specific domain or enterprise. It is also expected that the platforms' interfaces will be improved to facilitate the easy exchange of components.

Developing an application, such as our MRP-II system, by reusing existing code, for example code contained in Microsoft’s Office package, and using existing platforms, such as OLE, meets the fundamental idea of compo-
componentware. The objective of this initiative is to allow end-users to handle components with little effort and to avoid individual programming as much as possible (see e.g. [9], [10], [11]).

3 Motivation

At the Computer Science Research Group B of the University of Erlangen-Nuremberg, we wanted to explore to what extent it is possible to build rather complex applications like MRP II-systems with standard software. Our goal was to use well-known software packages which are available on conventional PCs. The tasks of an MRP II-system should be solved with existing functions avoiding individual programming as much as possible. This way we wanted to generate a „lean“ MRP II-solution for SME.

3.1 Lean Production of application systems

- Modular design
- Inexpensive hard-and software
- Rapid Application Design (RAD)
- Reuse of tested software

Figure 1. Advantages of using software components

The advantages of our approach are the following ones: Since we resort to already tested software, the obtained modules promise to be robust and reliable. Therefore, such developments are less risky than comparable software that is designed from scratch. Besides saving time for developing tailored software at considerably lower costs, the period of time needed for end-user training is reduced and the acceptance is high, since the user usually knows the software. At last not only component software, but also the PC hardware is cost effective (see fig. 1).

4 Technical Foundations

The Microsoft Office Suite lent itself as the tool for our intentions. The line of objects and functions within the software packages enables application systems to be assembled in a „lego manner“: In this context, we want to point to the relational database management system Access, the spreadsheet analysis program Excel as well as to the project-planning system Project. Besides their core tasks, these packages provide functions that can be extended in order to partially solve the tasks of an MRP II-system. Linear Programming (Excel) can be used for master production scheduling. The database management functions of Access were used to record master data, the reporting forms are suitable for printing schedules of job operations or stock-keeping forms. The package Project holds scheduling tools and functions for capacity planning. The Microsoft standard functions were adapted and enlarged to components for MRP II-tasks (see fig. 2).

4.1 Technical Foundations

Microsoft’s proprietary component standard, OLE, enabled us to merge these software elements into a complete MRP II-system. With the programming languages Visual Basic and Visual Basic for Applications (VBA), which is part of the Office line, we found comprehensive instruments for the necessary adaptations.

5 Development Process

The process for developing application systems with componentware can be described as follows (see fig. 3, according to [12]):

1. Search/Retrieval
   The problem area must be analyzed. The developer has to select, verify and rate potential components for the specific case. Hereafter the „best-of-breed“-component of the alternatives must be identified.

2. Adaptation
   In general, the selected component cannot be deployed „as is“. Therefore, a programmer adapts it using simple script and programming languages. In the future,
3. Integration
The selected and adapted fragments have to be integrated in existing application systems or frameworks.

4. Out of the constructed system, new candidates for a future reuse can be identified. Hence new components for other cases are emerging.

5. Qualifying candidates have to be adapted to the standards of the existing component libraries concerning quality, documentation and classification. After this, they can be incorporated into the libraries.

6. The architecture of the prototype
An MRP II-system has to solve the following core tasks (see e.g. [13]):

- Master data administration
- Master production scheduling
- Material requirements planning
- Scheduling of orders and capacity adjustment
- Shop order release
- Sequencing of orders

Every single core task is designed as a component of rough granularity which serves as a framework for finer enterprise-type-specific MRP II-functions. The common services of the MRP II-system like administration of API-functions, linking of external database tables, and run-time monitoring are located in an Access front end. This framework activates the interfaces of the MRP II-components and transfers data. Access is the OLE-client which fetches services from the other applications. Graph is a suitable tool for the presentation of statistical data. Via OLE charts are embedded into an Access form. A user does not need to know that an external application is linked. OLE-automation is an appropriate means of controlling Excel objects from Access. Sheets can be generated and formatted with formulas and data. The results again flow back into the Access database. Other data are transferred from Access to Project where scheduling functions are applied. At first, we used a standard database interface provided within Project. But as problems occurred we implemented our own dialog management services based on DDE (see fig. 4). The MRP II-functions are strictly separate from the data they need. The whole master data are located in a central database. The different modules only communicate via this database and are not connected with each other. The design of the MRP II-components and how they embed the Microsoft functions is described in the following sections.

6.1 Master data administration
An MRP II-system must manage several kinds of data, some of which have a complex inner structure:

- Customer and supplier information objects
- Product descriptions
- Manufacturing resources
- Shop orders
• Material plans for products, organized in a layered structure called a bill of materials

In this segment of MRP II, already simple database applications and queries are required which we implemented with Access. The components we used were Access forms and controls. We also could apply standard functions of the package like searching for items in a database field, locking of data records in use, and copying information. As the database applications should be separate from the master data they contain, we linked the external data into the master data administration. Every other MRP II-component is connected to this central database. Access also holds ODBC-interfaces for linking tables of other databases such as dBase, Paradox, and Btrieve.

6.2 Master production scheduling

Master production scheduling is the first step in the MRP II planning hierarchy. It determines the production program for a defined planning horizon (e.g. a week or a month). The production program is composed of certain and uncertain product demands. If a customer orders a defined number of products the manufacturer exactly knows how many parts and subparts he has to plan. Some enterprises also produce for an anonymous market, so they have to estimate how much to produce. Forecasting methods serve to predict future values on the basis of the past. Simple methods like averaging, exponential smoothing, and moving averages assume linear dependencies between the past and the future. More complicated ones such as double exponential smoothing and the Winters method handle seasonal variations. Bicycle manufacturers will sell more products in spring and summer than in autumn or winter.

After the production program of a defined period is determined, a calculated matrix gives an overview of the necessary machine capacity. This tool detects capacity violations already in this early stage of the planning process. In order to bypass such problems capacity can be manually adjusted. The first possibility is to reduce the master production schedule, the second alternative is to switch from one- to two-shift work for bottle-neck machines.

Both the customer order administration and the forecasting methods were implemented with Access. Thereby we could reuse some prefabricated functions like averaging and queries. The progression of the historical product demands of a defined period in comparison to the suggested values calculated by the forecasting functions is illustrated using Graph (see figure 5). Via OLE data of the Access database is transferred to Graph. The Graph chart is then embedded into an Access form. In this case Access is the OLE-client which fetches services from Graph, the OLE-server. The coefficients of the capacity requirements matrix are calculated with Access Basic modules from master data (schedules of job operations and bills of materials). We hoped that Excel with its multitude of mathematical and statistical functions could save us programming effort. But as some of the necessary functions were not available, we had to resort to the programming languages. As Access offered basic calculation facilities we needed for our programming, we avoided the data exchange between Access and Excel at this point.

Linear Programming is a sophisticated mathematical model which can be found scarcely in modules for master production scheduling on the software market. It allows the user to generate a master production schedule which maximizes the contribution margin or the capacity de-

![Figure 5. Forecasting within the master production scheduling (Access - Graph)](image)
mands in manufacturing without violating the capacity constraints.

The Linear Programming for adapting capacity is implemented by an Access-Excel dialog which is based on OLE-automation. The Excel Solver, a tool which holds a lot of functions for this method, reduced the individual adaptations to a minor extent. We load the necessary data from the external database into Access and generate an Excel-sheet object which is formatted for the mathematical model by adding formulas and constraints. After this the Excel sheet is embedded into an Access form. This process is not visible for the user who starts the optimization. The program changes between the applications automatically and the user starts the method. The algorithm for the integer Linear Programming problem, namely a special form of the Simplex Algorithm, is provided by the Excel Solver. Supposing that a valid solution of the problem exists, it is displayed as the required capacity of the machines in the Excel diagram. The degrees of machine utilization are presented in the capacity requirements matrix of the model. Excel uses the Graph component in order to visualize this in a diagram (fig. 6). The provided information serves as a hint for necessary investment decisions, because both bottle-neck machines and under-utilized machines are detected at one glance.

The Access forms, queries, and the Excel-sheet of the Linear Programming are isolated as a stand-alone component. Via our MRP II front end, we control the component.

6.3 Material Requirements Planning

Once the master production schedule is fixed, it is transferred to the material requirements planning. At first, we derive the quantity of each kind of material that is needed in the whole production process. After subtracting the available stock, the net quantities to be ordered from suppliers are calculated, arranged to optimal scales. Lot-size planning is executed hierarchically. The algorithms traverse the layered structure of the bills of materials to calculate product demands and aggregate these to optimal ordering units. It begins with the top layer and is recursively repeated for the lower layers of each top-layer part. As a result we receive a list of orders for parts to be provided by suppliers. On the other side we get a set of roughly scheduled shop orders for the manufacturing process. The system provides several heuristics for this MRP II-task: the Harris/Andler method, the least-unit cost heuristic, and the part period balancing.

For this central task of an MRP II-system, we only could exploit Access forms and queries because none of the components we used provides lot-size planning tools. So it is accomplished by Access Basic modules. The proportion of individual programming compared to reused standard functions is the highest of the components described in this paper (see also section 7.4). The MRP II-task consisting of its forms, modules and queries is plugged into the MRP II front end. It resorts to the external basic data.

Figure 6. Rough capacity planning with Linear Programming (Excel - Access)
6.4 Scheduling of orders

The scheduling of orders assigns the roughly scheduled orders to machine groups more precisely. The earliest and latest start and due dates are calculated. The backward-scheduling mechanism subtracts the duration of every manufacturing step of an order from the due date. Thereby working times of the machines, transport times of materials, and so on are taken into account. The result is a latest starting and finishing time of every manufacturing step in order to meet the deadline. The forward-scheduling mechanism starts in the next planning period and adds the durations. Here we obtain earliest starting and finishing dates for the sequencing of orders. It may happen that the scheduling results in capacity overloads. More orders have to be produced on a machine than is possible within the required time. These violations must be eliminated personally or automatically by a capacity adjustment. The user may postpone single operations or even complete orders, schedule overtime, or change the number of shifts for the affected machines.

Most available MRP II-systems, however, do not provide capacity adjustment heuristics. These are considered to be too complex in order to be adapted to customer-specific needs. Rather, these systems offer user dialogs for capacity adjustment [14]. Thus, our solution represents a step forward towards more user-friendly systems.

The Microsoft Project component turned out to be the most suitable for implementing the required functions. Forward- and backward-scheduling of orders, capacity adjustment, scrolling orders, and the calculation of buffer times could be found within the package. Also graphic tools like the GANTT diagram and the network plan for illustrating predecessor-successor relationships fulfill the needs of a comfortable MRP II-system.

On the other hand, the exchange of master data from Access to Project and vice versa caused problems. In consequence, we had to implement our own dialog management services. At first, the required data are loaded from the external central Access database into Project. The Project interface is embedded into an Access form. Via double-click the user changes from the OLE-client, Access, to the OLE-server, Project. There he can inspect the situation of the machines, move orders, change shifts of the machines, and so on (see fig. 7). The whole Access-Project dialog is designed transparently for the user.

The capacity requirements planning component, including forms, modules, queries, and embedded Project services is plugged into our MRP II-front end.

6.5 Release of orders

In order to meet the due date a manufacturing order has to be released between its earliest and latest start date (scheduled in Project). The module for the release of orders checks the availability of the material that is needed for the production process. After this the orders are released and the material can be taken from the stock. If needed, the module initiates new orders for parts that are disposed by consumption-based material budgeting and that reach the minimum-inventory level. The forms that are necessary for manufacturing orders are printed automatically by the report generator of Access. The technical design resembles other Access-based components described in this paper, such as the material requirements planning in section 6.3.

![Figure 7. Scheduling of orders (Project)](image-url)
6.6 Sequencing

The released orders are competing for the machines on the production floor. When several operations of different orders are assigned to one machine at the same time, a sequencing problem arises. To resolve the conflict we implemented several priority rules in Access: the FCFS rule (First Come First Served), the SOT rule (Shortest Operation Time) or the ST rule (the customer with the shortest remaining Slack Time is served first).

This task was implemented using Access and Project; the latter allowed the illustration of Gantt diagrams and orders of job operations.

After the orders are released and the sequencing is fixed, the operations are executed on the machines. The operational data are recorded. They could be used to analyze whether or not deadlines were met, how much rejects were produced, how long the medium operation time was, etc.

7 Experiences when using Microsoft components

The MRP II-system we developed is tailored for the use in SME. Thus, we avoided implementing services and algorithms that would not be accepted by such clients. Our experiences with this approach are as follows.

7.1 The fit of the Office applications for MRP II-functions

Project. It turned out that Project contains various scheduling functions that widely meet the requirements of an MRP II-system. For example, the network planning technique, the standard and resource calendar as well as the possibility to calculate cost and buffer times could be directly used. Only minor adaptations were necessary, such as the Access-Project dialog management services, and the calculation of the amount of capital tie-up in the manufacturing process.

Access helped us to build the master data administration. Its inherent forms are an especially powerful tool to build Graphical User Interfaces (GUI). Its reporting generator served well to create the documents needed in an MRP II-system, such as schedules of job operations, bills of materials, or material orders. However, to adapt these to our needs much extra programming was necessary. For example, forecasting functions within Access are limited to simple algorithms like averages. Therefore, more sophisticated solutions which incorporate trend and seasonal variations had to be coded. But since the architecture is modular, some functions that were implemented once, like the explosion of the bills of materials, are reusable.

Excel offers a lot of mathematical and statistical functions, some of which are rather sophisticated. The graphical presentation methods are easy to handle. Adaptations were inevitable here in order to match functions to the requirements of an MRP II-system. An advantage of Excel is its considerable acceptance by end-users. Some SME which do not yet have an integrated MRP II-system use the package for single MRP II-functions (see e.g. [15]).

OLE. The OLE-technique serves as the backbone for the integration of the adapted Microsoft Office functions to our MRP II-prototype. In some aspects the technique is documented inadequately. Thus, problems occurred with the Access-Project dialog. OLE helped us with the Linear Programming tool. Access generates an Excel-sheet object, which includes data and formulas of the database. The parameters are prepared for the Solver and then saved under Excel. A user does not take notice of this transaction, since Excel is launched automatically within Access. With the amelioration of the OLE-support the approach might be extended from Microsoft components to software of different producers. When building applications like our MRP II-system, developers have the choice between several alternatives and will select the "best-of-breed" components for every single task.

A considerable benefit of this kind of software development is the dominance of Microsoft users on the software market. The experiences and knowledge of the community is more and more linked by the Internet to the Microsoft Developers Network. Hence several million software users access the various newsgroups about the topics Windows, Access, Excel, Project, Visual Basic, and OLE. A lot of problems that neither the documentation nor the Microsoft hotline could solve, were remedied by consulting newsgroups.

One disadvantage of Microsoft’s OLE-technique is its restriction to Windows environments. The Software AG is transferring OLE to UNIX platforms (see [16]). So the approach may also work on different hardware types in the future. But already today, there are applications, such as Java Beans, which are independent of the hardware environment.

7.2 Performance test

An essential acceptance factor for every system is performance. Therefore, the development of the MRP II-components was permanently accompanied by performance tests. In first laboratory experiments, we evaluated the access and computing times of basic operations like searching, sorting, database queries, and explosion of bills of materials [17]. For our tests we used tables of various sizes: with 1,000, 10,000, 20,000, 30,000, or 40,000 parts. This may serve as a realistic scenario for small and medium-sized manufacturers [18]. The results of the different
tests were satisfying, because the response times were lying within second intervals, for example 1 to 29 seconds for sorting operations, 1.6 to 42.6 seconds for simple SQL queries on a 486DX/33 computer.

We also tested the computing times of more complex MRP II-tasks, such as the lot-size planning. Each schedule of the job order and each bill of materials have to be accessed. For one planning period we generated a demand of 69 different end products (bicycles). The material requirements planning derived

- 292 manufacturing orders with 1,720 job operations,
- 63 purchase orders with 1,195 materials.
- Furthermore 3,376 inventory notes like reservations of materials for orders and correcting the available stock had to be made.

The calculation times differed widely, dependent on the hardware power we used:

- One hour on a 486/DX33 processor
- 40 minutes on a Pentium-90 processor in a network
- 20 minutes on a Pentium-100 processor stand-alone

We expect further improvements when using powerful servers and networks.

### 7.3 Release Management

In a next step, we want to analyze the effort for adaptation with new releases of the Microsoft Office package. Since the start of our research project in late 1994 two new versions emerged. With Office95 we had to redefine API-functions from 16 to 32 bits. The version holds tools for converting old objects to new ones. However, adaptations of the code were inevitable because of syntax changes in the VBA programming languages. With Office97 Microsoft changed the data format for the applications once again. So we had to transform the source code for a new version of the MRP II-prototype. Although the Office97 packages contain tools for converting code adaptations were inevitable. Furthermore, the products, especially Access and Excel, did not work as reliable as old versions used to. One significant barrier for our approach is that we had to manage releases and support for three different releases for various Office platforms, if we were a commercial software provider. A more detailed analysis of the migration will be part of our future work.

### 7.4 Effort for development

The proportion of functions that were provided by the components we used to the ones we developed and adapted was guessed for our MRP II-tasks. Figure 8 shows that for the master data administration 90 % of the needed functions were satisfied by Microsoft’s inherent tools. 10 % had to be added by coding. For the material requirements planning 30 % were embedded in the Office environment, another 70 % were added. We guessed this balance for all our MRP II-functions and weighted it with the effort for development measured in hours. As a result we have 48 % of the prototype that was provided by the components and another 52 % that was added by individual programming.

### Figure 8. Application development with Microsoft components

The prototype was developed by two research assistants and five graduate students. The human resources that were invested meanwhile amount to about 65 manmonths. This turned out to be a rather modest effort for an application system of this size.

### 8 MRP II-components for alternative enterprise types

#### 8.1 Two different solutions for the material requirements planning

To evaluate the adaptability of the system, we developed two different material requirements planning modules. For serial producers we needed the version described in the previous section. For enterprises that produce only single-items, we can provide a more simple version. The module for planning the secondary requirements places orders for materials under the assumption that no lot-size planning is necessary. Only predecessor-successor relationships are taken into account. The result of the planning process is then transferred from Access to Project. For the exchange of two components, no adaptations in the source
code is necessary. We only have to make some entries in the registration tables of the MRP II-framework.

We plan to implement further alternate MRP II-components for various enterprise types. This demands detailed knowledge concerning specific requirements. The Forschungsinstitut für Rationalisierung (FIR) at the Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen, Germany, developed a morphologic scheme which makes it possible to assign enterprises to an enterprise type (see e.g. [19]). It also provides an extensive database of MRP II-systems on the marketplace and tools for the rating and selection of solutions (see [20]). Within a cooperation project we want to use this scheme to derive intrinsic software functions. The assignment of software functions to specific characteristics of a small enterprise will be part of our Component Configurator.

8.2 The Component Configurator

To support the adaptation of the system to different customer-specific needs, we provide a so-called Component Configurator. With its help the system can be configured with specific components easily, for example with the appropriate version of the material requirements planning module. A first prototype of the Component Configurator is available. The administrator has to select different items of a property list. The enterprise is first characterized as a single-item producer, a small-batch producer, a make-to-order producer, or a serial producer. After this, different components can be selected. The result is a flow chart of the different steps to configure the specific MRP II-solution.

A future version will contain the morphologic scheme described above (see fig. 9). MRP II-solutions could be assembled from enterprise-type-specific components. By characterizing the kind of enterprise and its production process, the system administrator can configure the MRP II-system to the customer's needs. Appropriate components are selected and linked to the system automatically. An important feature is that you do not select components explicitly. Rather you specify the configuration in terms of the domain. By which components the desired services are implemented is not exposed. This allows you to configure the system even if you are not a software developer.

With the enterprise-type-specific components we want to evaluate if it is possible to build components of finer granularity with the Microsoft Office line. Single functions like a forecasting method, a lot-size planning heuristic, or the explosion of the bills of materials, could then be assembled to more complex components, which are convenient for special enterprise types.

![Figure 9. Component Configurator](image)

Implementing fine MRP II-components in order to combine them to components of rough granularity will increase the number of interfaces. So the buildup process may become complex. It has to be investigated whether it is easier to adapt components of rough granularity by individual programming or parameterization.

In order to provide generic MRP II-functions concerning input and output data we start building data dictionaries. This strategy helps to avoid adapting source code for changes of database table and field names.

9 Conclusions

The tasks of an MRP II-system can be satisfied astonishingly well with components embedded in standard software. Even sophisticated algorithms, which by far are not part of all standard MRP II-systems, were implemented. Concerning the functions, the system can compete with a conventional MRP II-system of medium-range complexity. The prototype also withstands critics concerning performance despite the use of conventional PCs. Further benefits are expected with respect to the introduction of the system and training, because end-users access well-known software.
A considerable benefit of this approach is the short time to accomplish a customized solution measured in human resources. MRP II-components such as material requirements planning or master production scheduling could be adapted to specific requirements with little effort. In the future, encapsulated industry- and enterprise-type-specific components of different granularity shall be provided.

The Component Configurator resorts to a morphologic scheme, which assigns an enterprise to a specific industry type. Components will not be selected explicitly. Rather one specifies the configuration in terms of a domain. This allows to configure the system even without being a software developer.

Software providers such as Microsoft could make component-based system development easier by improving the handling of OLE-techniques. Until now using OLE-automation requires a rather high experience level. The improvement of the documentation could contribute a great deal to the aim of componentware: software users rather than developers shall merge existing functions such as Office packages to an application using simple modifications. For tasks where the gap between required and existing standard software functions is too big, a component market could evolve.

For material requirements planning or job shop scheduling we only could exploit a minor part of standard software functions. For these tasks it is conceivable to encapsulate ActiveX.

With the emerging component standards, such as JavaBeans and ActiveX, and design patterns [21] the trend to develop components will continue undoubtedly. Software enterprises will deliver exchangeable components in the area where they have their core competencies. Those will be traded over shareware servers, software stocks, or Component Warehouses over the Internet [22], [23]. Assembling application systems from „best-of-breed“ components will become even more popular.

10 References