AI Magazine Volume 16 Number 1 (1995) (© AAAI)

Routine Design for Mechanical Engineering

Axel Brinkop, Norbert Laudwein, and Rüdiger Maasen

■ COMIX (configuration of mixing machines) is a system that assists members of the EKATO Sales Department in designing a mixing machine that fulfills the requirements of a customer. It is used to help the engineer design the requested machine and prepare an offer that's to be submitted to the customer. COMIX integrates more traditional software techniques with explicit knowledge representation and constraint propagation. During the process of routine design, some design decisions have to be made with uncertainty. By including knowledge from process technology and company experience in the mechanical design, a sufficiently high degree of flexibility is achieved that the system can even assist in difficult design situations. The success of the system can be measured by the increase in the quantity and the quality of the submitted offers.

The system described in this article is currently working in the field at the Sales Department of EKATO, one of the world's most successful manufacturers of industrial mixing machines. It was developed in close cooperation with the Fraunhofer Institute for Information and Data Processing (IITB) during a three-year period. COMIX (configuration of mixing machines) is a system that assists members of the Sales Department in designing a mixing machine that fulfills the requirements of a customer. Because mixing machines have to be tailored individually to meet a customer's requirements, this design process is time consuming. COMIX is used to help the engineer design the requested machine and prepare an offer that's to be submitted to the customer.

COMIX integrates more traditional software techniques, such as databases, algorithmic computations, and iterations, with explicit knowledge representation and constraint propagation. During the process of routine design, some design decisions have to be made with uncertainty. Therefore, the process is guided by a combination of several strategies to help resolve the conflicts. For example, variable ordering, dependency-directed backtracking, and knowledge-guided backtracking are used. Nevertheless, all main design decisions are left to the user.

By including knowledge from process technology and company experience in the mechanical design, a sufficiently high degree of flexibility is achieved that the system can even assist in difficult design situations.

The success of the system can be measured by the increase in the quantity and the quality of the submitted offers. The average time needed to prepare a sales quote for a mixing machine was reduced by two hours. This reduced design time will lead to an amortizing of the investment in COMIX within 2.5 years of its introduction into the field. In addition, the amount of offers that led to contracting rose 20 percent. These measures show that the use of COMIX has been beneficial for EKATO.

The Domain: Industrial Agitators

Industrial mixing machines, better known as *agitators,* are widely used in industrial manufacturing. They are especially useful for the chemical and pharmaceutical industries, food production, and biotechnology. The basic structure of an industrial agitator is shown in figure 1.

The following aspects of the domain are important: First, agitators are individually tailored to meet the special requirements of a customer; so, they cannot be mass produced. Deciding whether a given agitator can perform a special mixing task can only be solved by comparing the new task with previously solved mixing tasks. Second, the space of possible configurations is large.

The basic structure of an agitator is simple,

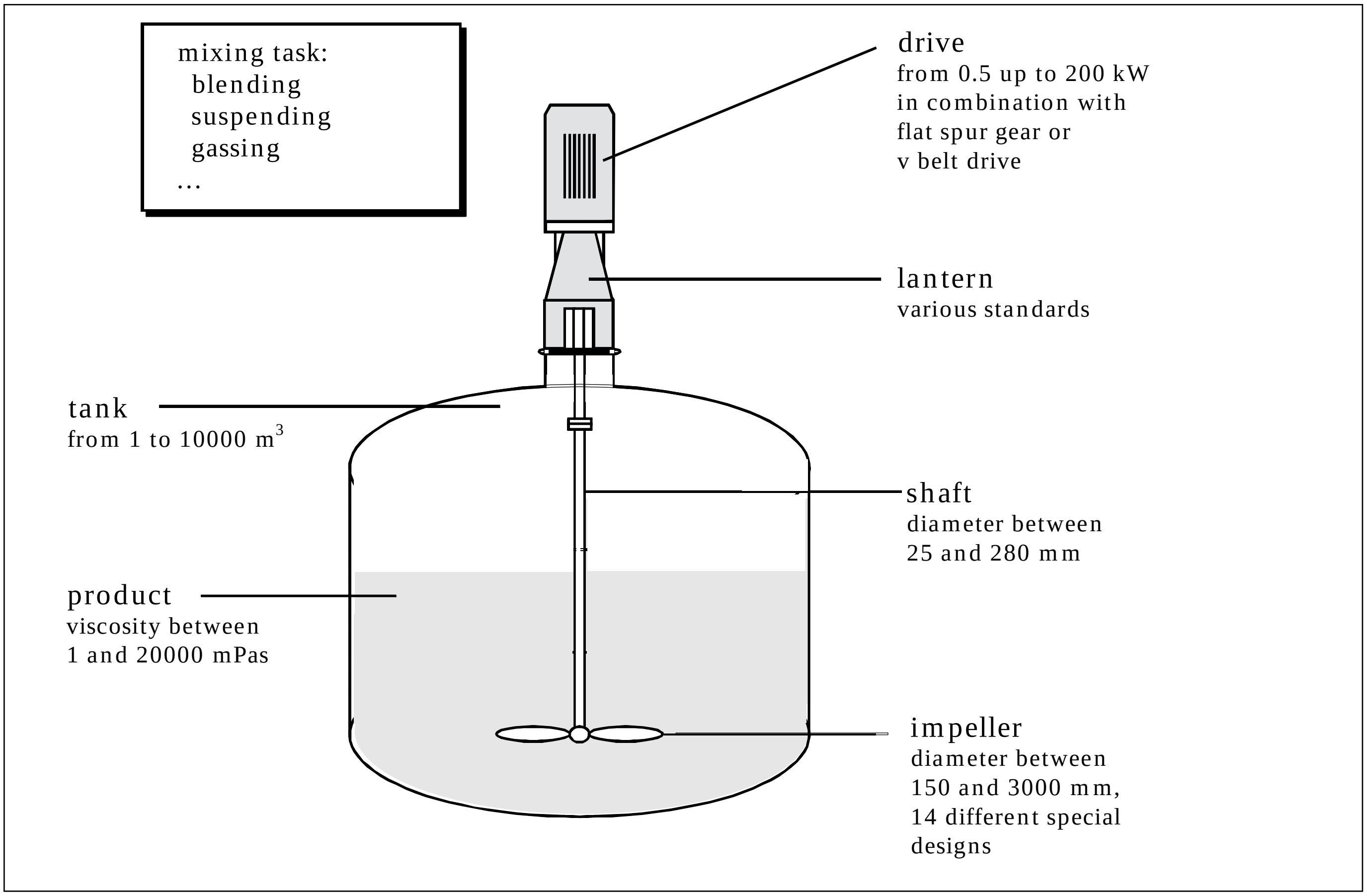


Figure 1. Basic Structure of an Agitator.

but each component can be constructed using a variety of different parts. For example, there are not only propellers but also various special forms of impellers; different shafts are available; and thousands of direct motors, providing power from 0.5 kilowatts (kW) to 200 kW, produced by several manufacturers can be used. The direct motors can be used with flat helical gearboxes, v-belt drives, or a combination of both. There are also some optional components, such as a sealing between lantern and tank or an additional bearing at the bottom of the tank. It is also possible to add some fixtures, called *baffles*, to the tank.

In addition to the wide variety of parts for each component, the design task also has to cope with several different mixing tasks, such as *blending*, that is, making a homogeneous liquid out of two different solvable liquids, or *suspending*, that is, achieving a relatively uniform distribution of solid particles in a liquid, a mixing task common in lacquer production. Another task is *gassing*, that is, bringing gas into liquid, a well-known method in the field of wastewater treatment or flue-gas desulfurization.

Main Objectives

For a long time, EKATO has used computers to aid in the design of the mechanics of the agitators. About 15 years ago, the first programs were developed at EKATO to assist in the mechanical design of agitators. These programs have been in continuous use since then and have repeatedly been revised and extended to reflect changes in manufacturing and process technology. Recently, however, it was determined that these systems could no longer be maintained. Thus, a new system was needed with configuration knowledge explicitly represented so that maintenance would be easier.

Analysis of the strengths and weaknesses of the old system led to expanded functions: (1) integration of process-engineering knowledge; (2) high flexibility; and (3) assistance even in difficult design situations, for example, when the number of revolutions of the configured machine meets the frequency of natural oscillation.

By including industrial process-engineering knowledge, two objectives could be achieved: First, knowledge from different company experts could be preserved. Second, the software used by the sales staff for designing the machines could be run on a standard platform.

More flexibility in the configuration enabled the reduction in the variety of parts of some of the key components. This reduction led to the use of a higher number of premanufactured parts, which, in turn, led to considerably reduced manufacturing costs.

Assistance in difficult design situations, such as conflicting number of revolutions and natural oscillation, saved design time and increased the quality of the design.

Tasks

COMIX has to perform a set of different tasks depending on user requirements and information given by the customer. In the agitator domain, it is difficult to decide whether a given agitator fulfills a given mixing task. The main problem is the design of the impeller system. The phenomena in agitated vessels defy a description by equations in an exact physical way; so, the problem can only be solved based on experience and a comparison of the task with previously solved mixing tasks. If the mixing task is somehow new, experiments with scaled-down mixing machines have to be done.

However, once the impeller system is specified, then the mechanical design of the agitator can be treated as a configuration problem. Given a well-defined set of available components, the system only needs to find a combination of single components that will work together as intended.

COMIX performs the following tasks: First, given the user requirements, the system derives a specification of the impeller system for several basic process-engineering tasks. Second, given the specification of the impeller system, COMIX finds a correct and complete configuration of the agitator.

Deriving Structure from Function

Usually, a customer presents a set of requirements for an agitator and asks the manufacturer to submit a bid. A responsible engineer within the Sales Department has to design a mixing machine that meets the particular requirements of the customer. Given a desired functioning of the machine, the system has to derive a specification of a machine with the intended functions. Based on knowledge from process technology and prior experience, the system is able to handle the following basic process-engineering tasks: (1) blending two different solvable liquids; (2) suspending solid particles in a liquid; (3) transferring heat, that is, cooling down or heating up a liquid; and (4) gassing.

The agitator is specified by the design of the impeller system, that is, number, design, and diameter of impellers; their positions in the vessel; and the number of revolutions. This design also includes the number and size of the baffles in the vessel. Given a specification of the impeller system, it is possible to derive the necessary components of the agitator.

Configuration

Given the specification of the impeller system, COMIX has to find a detailed and complete configuration of an agitator that will work as intended. The set of available components is well defined. For some types of component, only a small set is available. For others, thousands of choices are available. Some parts even fall into a continuous definition set. Laws of mechanics, standards and quality guidelines of the manufacturer, and industrial standards, as well as intercomponent constraints, have to be taken into account. Additionally, constraints from process technology have to be satisfied to guarantee the intended function. The result consists of a detailed specification that includes price information, information for manufacturing, and a dimensional sketch with the basic geometric measurements.

Redesign and Case-Based Redesign

COMIX has to perform the *redesign task,* that is, the adaptation of an agitator to changed prerequisites. Often, a customer is satisfied with an existing machine and would like to have it rebuilt, or similarly, during contracting, it turns out that the mixing task is slightly different from the original prerequisites for the design. In this case, the agitator's current configuration can be loaded and modified. The system will then compute all the necessary changes, taking into account the actual parts available and their properties.

Another possibility during routine design is that the problem is similar to a previously solved problem. For example, the mixing task could be similar to a previously solved mixing task, or the design of the impeller system is closely related to a previously designed agi-

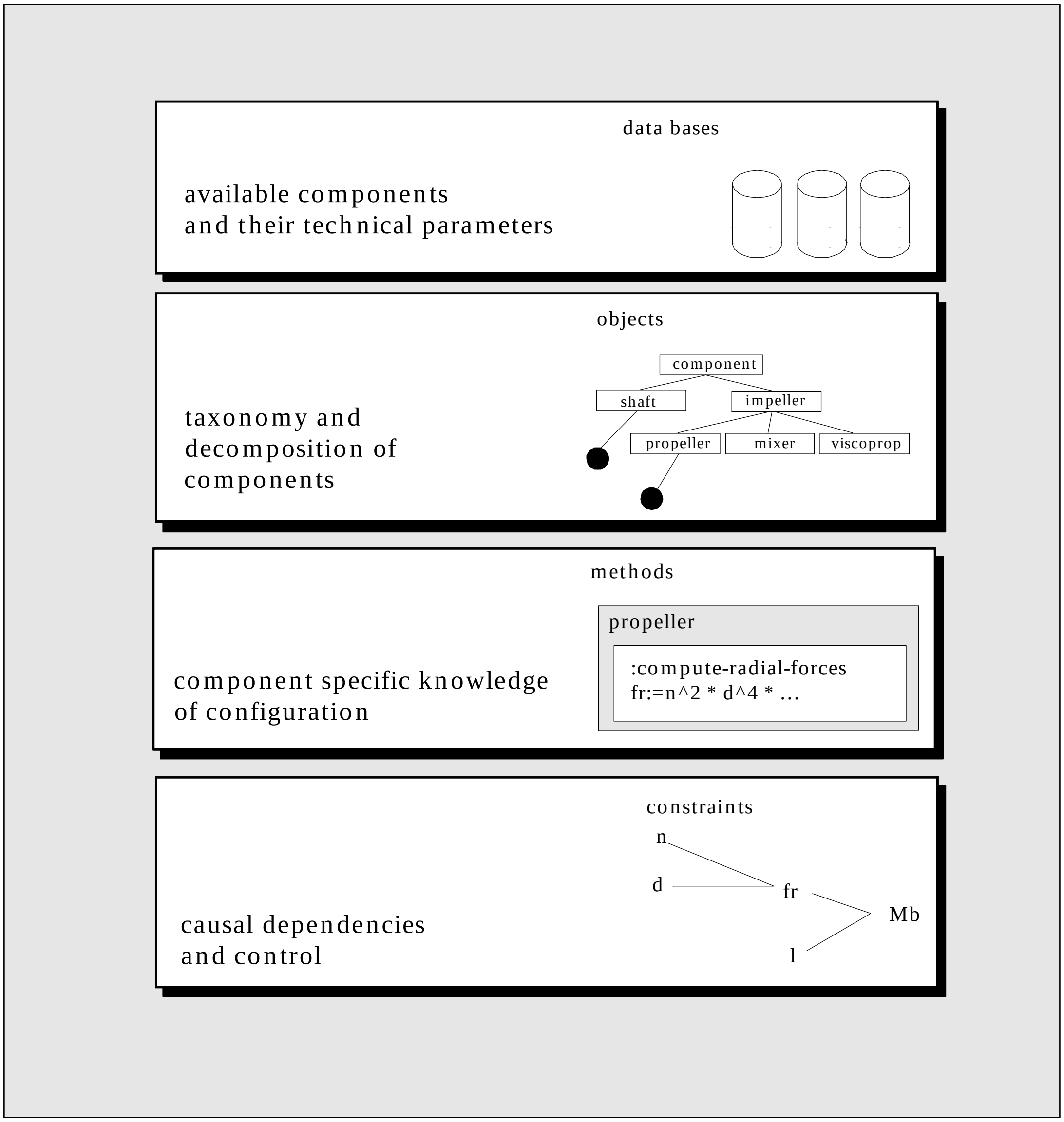


Figure 2. Knowledge Types and Their Representations.

tator. The search for similar mixing tasks or similar agitators is performed by searching databases, which contain information about previously configured agitators. The criteria for search are left open to the user, so that the user can define his/her own similarity criteria.

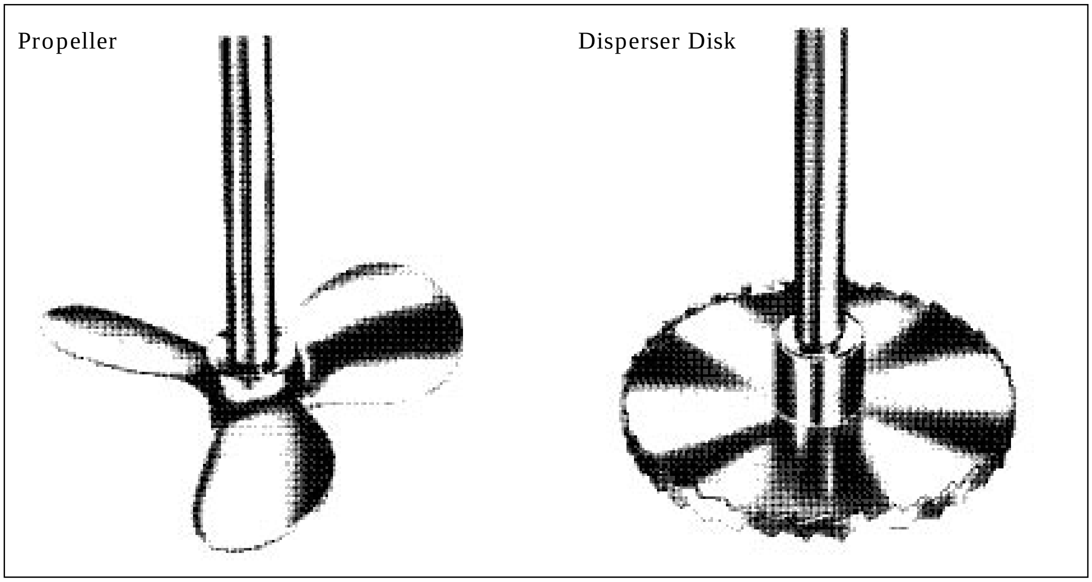
Nonroutine Design Assistance

COMIX can also assist in nonroutine design by looking for partial configurations with regard to the available information. When designing a special-purpose agitator, such as a specialized impeller that is not in the system knowledge base, it is possible for the system to look for a shaft and a drive that will handle the forces placed on the impeller. To perform this operation, the user has to know some special characteristics about the impeller and enter them into the system.

The System: Knowledge Representation

In this application domain, four types of domain-specific knowledge can be identified. Each type is represented differently in COMIX. Figure 2 provides an overview of these knowledge types.

First, available parts and their technical data are kept in various databases. Some components, such as motors and gears, are not manufactured on site but are bought from



.Figure 3. Two Different Designs of Impeller.

other manufacturers. Data about such components, such as pricing, frequently change. It is important to make updating simple. By keeping the data in databases, the data are easy to maintain without changing anything inside COMIX.

Second, knowledge about structure and classification of components is represented in different part-of and is-a hierarchies in an object-oriented manner. This approach enables maintenance with little work in case of changes in the types of manufactured part. During the development of COMIX, this kind of maintenance was often necessary and could easily be done.

Third, component-specific configuration knowledge is the knowledge about how to determine a special component or a single parameter of a component or how to satisfy a related constraint. This information is often given in algorithmic or computational form and is coded in a functional way. It is attached to components by methods, which enables message passing to objects without having to worry about the component's type. For example, the system can send the message "compute-your- weight" to the impeller of the agitator without checking if it is a marine propeller or a dispenser disk. These impellers are different and use completely different methods to compute their weight. They are shown in figure 3.

Fourth, knowledge about causal dependencies is represented using directed constraints. These constraints are dependencies in the sense of design knowledge, not physics, that is, which component depends on the design of another component. In combination with a so-called standard agenda, this agenda reflects the order in which an experienced engineer would configure the machine. This knowledge is used to direct the process of configuration.

The constraints are controlling the design process. In each construction cycle, the constraints that can be applied are determined. To satisfy the constraint, the corresponding constraint function has to be applied. In each constraint function, the configuration knowledge about how to design a component is represented. Each design decision is made by sending messages to the related components. The configuration knowledge is attached to the components by methods that are activated by messages. The configuration knowledge also includes information about how to determine lower and upper boundaries for parameters, which are used as criteria for database searches.

Process of Routine Design

The design process is accomplished in two stages: First, a specification of a machine with the intended functions has to be derived from information about the mixing task. This step differs depending on the basic process-engineering task, that is, blending or suspending. Thus, different data are needed. For example, in the case of blending, only density and viscosity of the product have to be known to determine the impeller system, whereas for suspending, particle size and solids volume concentration also have to be known. This process knowledge includes complex algorithms, scaleup computations, and iterations. It is hand coded for each basic mixing task for the sake of performance.

Second, the detailed mechanical design of the machine has to be determined. This design is based on a constraint propagation. The constraints used in COMIX can be characterized as (1) *directed,* the values of all but one of the parameters have to be known to determine the missing one; (2) *simple,* the set of possible values of a constraint parameter contains at most one element; (3) *functional,* the definition of the constraint relation is given by a function in contrast to a characteristic predicate; and (4) *constructive,* given all but one of the values of the constraint parameters, the missing value to satisfy the constraint can be determined constructively.

For each constraint, the following are explicitly stated: (1) which parameters have to be known; (2) which parameter can be determined; (3) which additional preconditions have to be satisfied; and (4) where to find the definition of the constraint relation, that is, which function to evaluate to satisfy the constraint.

The chosen algorithm for the propagation is based on the use of local consistency. A solution is found when an overall local consistent value assignment is found, which also leads to global consistency using only simple constraints.

During the design process, some decisions have to be made under uncertainty. Backtracking cannot be avoided. Several strategies are implemented to guide the propagation to reduce backtracking as well as manufacturing costs.

Variable Ordering for Backtracking Reduction

A good order of design decisions can often lead to a reduction in, or even an avoidance of, backtracking. The ordering knowledge can come from experience as well as from dependencies between decisions. Using constraint propagation results in an order of design decisions based on dependencies. Conflicts can arise when there is more than one constraint ready to fire at the same time. To resolve this kind of conflict, a standard agenda is introduced. All relevant parameters of the design process are listed in this agenda in a fixed order. This agenda reflects the order an experienced engineer would follow during the design. In case more than one constraint is ready to fire at the same time, this agenda is taken into account when determining which constraint should fire first. The use of this agenda leads to considerably reduced backtracking in many cases.

Dependency-Directed Backtracking

The process of finding a suitable configuration can be regarded as a search process in the space of all possible combinations of available components. In this domain, the parameters are highly interdependent, and there are many dependency cycles; so, the use of guessing and backtracking in the search process cannot be avoided. The search process is guided by the domain-independent strategy of dependency-directed backtracking that is similar to EL (Stallman and Sussmann 1977). For management of the dependencies, a mechanism such as the justification-based truth maintenance system (JTMS) (Doyle 1979) is implemented with special features to handle dependency cycles and to act with parameters that are represented in an object-oriented manner. For more details on the treatment of dependency cycles, please refer to Brinkop and Laudwein (1992).

Knowledge-Guided Backtracking

Generally, in COMIX, the satisfaction of a constraint implies the intended functions, but in one critical situation—when the number of machine revolutions is too close to the natural oscillation of the machine—the proper function is no longer guaranteed. Although the number of revolutions has to be fixed in an early state, the natural oscillation of the machine can only be determined when all parts of the machine are known in detail. Nearly all parameters depend, either recursively or directly, on the number of revolutions. In this case, even dependency-directed backtracking is too inefficient. To solve this crucial problem, the backtracking is guided by domain-specific knowledge combined with lookahead. When the number of revolutions is too close to the natural oscillation, the system has two possibilities—change the former or change the latter. Past experience demonstrated how to change some properties that were already selected to achieve an increase or a decrease in the natural oscillation without major influence on the overall behavior. Some of these techniques are tried by the system and tested to see if they would succeed. The succeeding activities are then suggested to the user, who can decide to act on the suggestions or choose another strategy to solve the problem.

Main Decisions by the User

The process of finding a configuration can also be regarded as a model-based search problem. It starts with a coarse model of the machine, the one implied by the specification. During the configuration process, the model is refined stepwise until all the parameters have been determined. In this process, all main design decisions are left to the user. COMIX determines the consequences of design decisions, evaluates constraints, and makes suggestions for possible choices. "This division of labor seems to build on the strengths of each party, making the computer responsible for completeness and consistency and the human responsible for strategy" (Steinberg 1987, p. 830).

Preferences

Another control strategy used in COMIX is the use of different levels of preference for key components so that manufacturing costs can be reduced. Premanufactured parts, which can be held in stock, are preferred as long as possible. Because these parts can be manufactured in large numbers, the manufacturing costs for the mixing machine are lowered. The use of premanufactured parts also helps to reduce the manufacturing time for a complete agitator.

For example, impeller-hub combinations are classified into three levels of preference: Level 1 is a small set of fixed impeller-hub combinations already manufactured. Level 2 is a larger set of impeller-hub combinations with existing blueprints for manufacturing that are manufactured as needed. For level 3, depending on the type and the diameter of the impeller, the diameter of the shaft, and occurring forces, the needed hub is computed. The combination has to be drawn in the Design Department and is manufactured as needed.

COMIX tries as long as possible to use the level-1 combinations. Only when it is no longer possible to use a combination at this level, the system asks the user if it should switch to a level of lower preference or revise a previous design decision.

Related Work

COMIX is designed to solve problems in the special domain of agitators. It was not designed to be a tool for configuration or design. Many ideas from other systems were taken and used when they were appropriate. The combination of these ideas led to a system that is now successfully used in the field.

The architecture of COMIX is similar to PLAKON (Cunis, Guenter, and Strecker 1991; Cunis et al. 1989), a tool for planning and configuration tasks. In COMIX, only propagation of directed constraints, in combination with JTMS techniques, is used, whereas PLAKON can handle nondirected constraints and offers different levels of control, from backtrack free to assumption-based truth maintenance system (ATMS) techniques (de Kleer 1986).

One of the best-known systems using the domain-independent strategy of dependency-directed backtracking is EL (Stallman and Sussmann 1977). However, COMIX is not able to handle EL's symbolic expressions.

Knowledge-guided backtracking and different levels of preference can also be found in VT and SALT, respectively (Marcus and McDermott 1989; Marcus, Stout, and McDermott 1988). The difference between VT and COMIX is related to the kind of constraints and the way they are used. Predicative constraints, in combination with procedures, are used in VT, but constructive constraints are used in COMIX. In VT, the strategy of knowledge-guided backtracking is used exclusively, whereas in COMIX, this strategy only takes place in special situations. Also, preferences are defined in VT for each parameter, whereas in COMIX, they are only defined for some special key components.

VEXED (Steinberg 1987), a design aid for NMOS digital circuits, is also based on top-down refinement plus constraint propagation. However, in VEXED, all control decisions are left to the user, including the order of design decisions and the choice of the way to proceed. In COMIX, only the main design decisions are left to the user.

Refinement techniques, in combination with constraint handling, can also be found in ALL-RISE (Sriram 1987), an extension of HI-RISE, which is a system for the design of high-rise buildings. In ALL-RISE, all possible solutions are explored, whereas COMIX stops at the first solution found.

Integration

COMIX is integrated into EKATO's information system. It can produce different printed documents and exchange data with the CIM system.

COMIX is implemented in GOLDWORKS III (GoldenCommonLisp, WINDOWS 3.1, DOS) and is running in a network of PCs using Novell NETWARE 386 for the local area network (LAN). The LAN is connected by gateway to the CIM

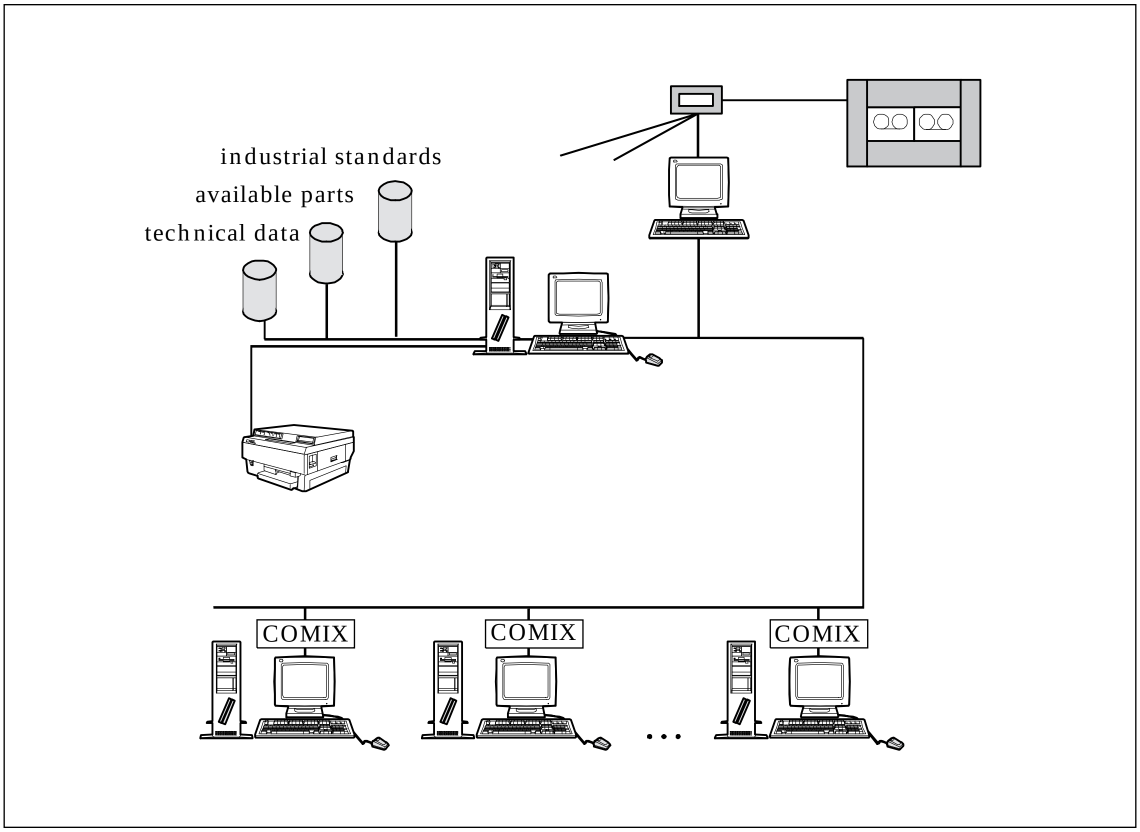


Figure 4. Overview of the Network.

system running on a mainframe. An overview of the network is given in figure 4.

Several printed documents are needed for the information system. One of them, which contains details of the configuration, including price information, is needed for documentation in an internal file. Another document, containing part-identification numbers of the CIM system and hints for the assembly of the parts, is used to guide the machine during the entire manufacturing process.

Additionally, a document presenting the final offer, including a drawing of the proposed machine, is produced. The offer is created by combining predefined text parts that depend on the actual configuration of the machine. With an integrated editor, the user can change the vocabulary of the offer, add special information, or change standardized formulations to get a document ready to send to the customer. The drawing of the machine is stored in a POSTSCRIPT document, with symbolic representations of the machine components in scale. It shows the tank as well as the agitator. Therefore, it is possible to get an impression of the size of the machine and its components.

Benefits

Integrating knowledge from process technology led to higher flexibility in the design of agitators and enabled the development of series of standardized parts. By building a large number of mixing machines with a small number of standardized parts, manufacturing costs are considerably reduced. Placing the agitator design on a standard platform allowed the comparison of different designs and reduced cases of malfunction or even nonfunction, resulting in a considerable reduction in design time and reclamation costs.

After more than one year of being in the field, some benefits are quantifiable. The total time needed for an offer lies between 4 and 12 hours, from first contact with the customer to the completion of the offer for the customer. The average time to prepare an offer was reduced by 2 hours, resulting in a savings of US$450,000 (DM 660 000) a year out of a volume of US$1,500,000 (DM 2 300 000) a year of total costs for producing offers. Comparing it with EKATO's costs of US$900,000 (DM 1 300 000) for the development of COMIX, the system will be amortized after COMIX is in the field for 2.5 years.

This reduction in design time was not achieved simply by the high performance of COMIX. With COMIX, it is possible to design a machine in less than 10 minutes, but the older system took 45 minutes. However, the main reason for the reduction in average design time is the flexibility of COMIX. It can even assist in difficult design situations with suggestions about how to solve a problem, such as when the number of revolutions approaches the critical speed. Currently, approximately 80 percent of all inquiries can be handled using COMIX.

The redesign ability of COMIX allows the user to try alternative designs to solve the mixing task and meet the customer's needs. Therefore, an increase in quality can be achieved, which was shown by the number of successful offers. Under COMIX, the percentage of offers that led to contracts increased from 27 percent to 32 percent. Thus, EKATO saw a 20-percent increase in the number of offers that led to contracts.

During the development of COMIX (three years), many maintenance activities were necessary. These activities were analyzed and incorporated into COMIX's design, resulting in a high degree of maintainability.

Development Phases

In this section, we look at basic concepts, the development process, and deployment of the system.

Basic Concepts

Most of the knowledge given by the experts, such as how to configure an agitator and when to choose a specific component, is specified using just a few parameters. The consequences of decisions made during the configuration of an agitator are not easy to judge; so, we have to cope with the problem of aggregating local information to find a global solution. For this reason, the configuration of agitators was based on constraint propagation.

Constraint propagation is also flexible. Depending on which kind and what subset of parts are used, the configuration process has to proceed differently. When using a sealing, the drop of power caused by friction at the sealing has to be taken into account when computing the minimal power of the drive. These alternatives can be expressed easily when using a constraint mechanism: Two alternative constraints have to be established, determining how to process with or without sealing.

The flexibility and extendibility also helped to increase the ease of maintenance and expand the system. It is easy to add new components or change dependencies between parameters. This kind of maintenance occurred often during COMIX's development.

It was planned to base the propagation on nondirected constraints. This decision failed because of the way the configuration knowledge was provided by the experts. It included formulas for computing parameters, complex algorithms for determining parameters, and database searches depending on lower and upper boundaries of parameters.

To derive a nondirected constraint from an equation, the equation had to be solved for each occurring parameter, which was not possible in some cases. Also, the ability to derive nondirected constraints from complex algorithms or database searches failed completely; so, the decision was made to use only directed constraints. Luckily, the restriction on directed constraints led to a simpler and, therefore, more effective mechanism for the evaluation of the constraints.

Development

The project started in the middle of 1989 with a preliminary project to investigate whether the problem could be solved by a knowledge-based system and to evaluate the hardware and software requirements. Because EKATO employees had experience with the PC using DOS, it was chosen as the hardware platform.

The selection of an appropriate tool was based on the following criteria: (1) it was PC based (DOS), (2) object-oriented representation was possible, (3) an interface to a high-level programming language was available to implement complex algorithms, (4) an interface to the database was available, and (5) there was support for the efficient development of a graphic user interface.

GOLDWORKS II was chosen, which was based on GoldenCommonLisp under WINDOWS 286 on DOS. During development, the platform was changed to GOLDWORKS III under WINDOWS 3.1.

The main project started in 1990. During the entire project, two persons at IITB and one person at EKATO were continually involved.

The knowledge-acquisition process started at IITB with an analysis of existing programs and printed documents. The basics about mixing technology were learned to facilitate speaking with EKATO experts.

Close contact between the developer and the experts was a characteristic of the main project. The knowledge-acquisition process ran in a permanent loop: acquisition, implementation, validation, and revision, with a two- to three-week frequency. For validation, single agitators were configured with the newest version of the system and were hand checked by the experts.

The development strategy was to construct a system as early as possible that could be given to EKATO for testing. Different stages of development and installation can be identified as follows: First, a stand-alone system was developed and tested in the Research and Development Department. Second, this system was extended to run on a LAN. This version was installed for testing in the Research and Development Department as well as in the Data-Processing Department. Third, COMIX was installed on several workstations in the Sales Department in mid-1991.

During development, it became obvious that our choice of DBASE as the format for the databases was a fortunate one. The databases had to be filled, for example, with data about thousands of motors and hundreds of gears. The data had to be checked and permanently investigated for errors. A lot of tools can handle files in DBASE format; so, everyone was able to choose the tool he/she was the most comfortable with for editing the databases. Therefore, the design and data-entry tasks could be done independently.

Deployment

The deployment started with the installation of COMIX in 1991. Only a small amount of user training was needed: Thanks to the graphic user interface, only 30 minutes of individual introduction were needed to train members of the Sales Department on the system.

The first version had nearly all the functions, but only a reduced set of components was available. The reduced set of available components provoked the majority of the criticism and led to low user acceptance: Only a few members of the Sales Department would work with the system.

However, the work of these few was worth a lot because they used COMIX for their daily work with great enthusiasm and patience. By carefully investigating each configuration done by COMIX, many of the remaining bugs were detected and could be fixed. Also, a lot of advice was given on how to increase user support and, therefore, the usefulness of the system for daily work.

Performance measurements showed that the system spent most of its time investigating constraints and controlling dependencies. To increase the performance, a new kernel was developed with efficient control of constraints and their dependencies. The new kernel led to a threefold performance increase.

Work also concentrated on extending the set of available components. For example, in the first version, only direct motors and helical geared motors were available. Later, the use of flat-spur gears, v-belt drives, or combinations of both were allowed. Figure 5 shows the possible combinations of gears and motors now available in the system.

Fast bug fixing, extended component sets, increased performance, and user support led to an increase in acceptance. Since mid-1992, the system has been in wide use in the Sales Department as the primary tool for agitator configuration and offer preparation.

In early 1994, the system was installed on 5 workstations in the Sales Department and is now used by 15 engineers. For maintenance and testing, the system is also installed on two workstations in the Data-Processing Department and on one workstation in the Research and Development Department at EKATO. One system is also installed in EKATO's American subsidiary. Currently, about 300 machines are designed monthly with COMIX at EKATO's facility in Schopfheim, Germany.

Maintenance

It was planned to turn over system maintenance to EKATO after the initial system was developed at IITB. To facilitate this changeover, courses started early in the project that covered Lisp, object-oriented programming, and the implementation of COMIX. The courses were held by the original developers of IITB for two members of the EKATO Data-Processing Department. This training was extended; modules of the system were jointly designed; and since mid-1992, modules have been implemented independently at EKATO. The main project between IITB and EKATO ended in 1993. Since this time, the system has been maintained and extended by EKATO.

During development, it was determined that maintenance activities could be divided into four different levels: (1) changes in price and technical component parameter (such changes happen frequently, and maintenance can be done by updating database fields), (2) changes in the manufacture of components

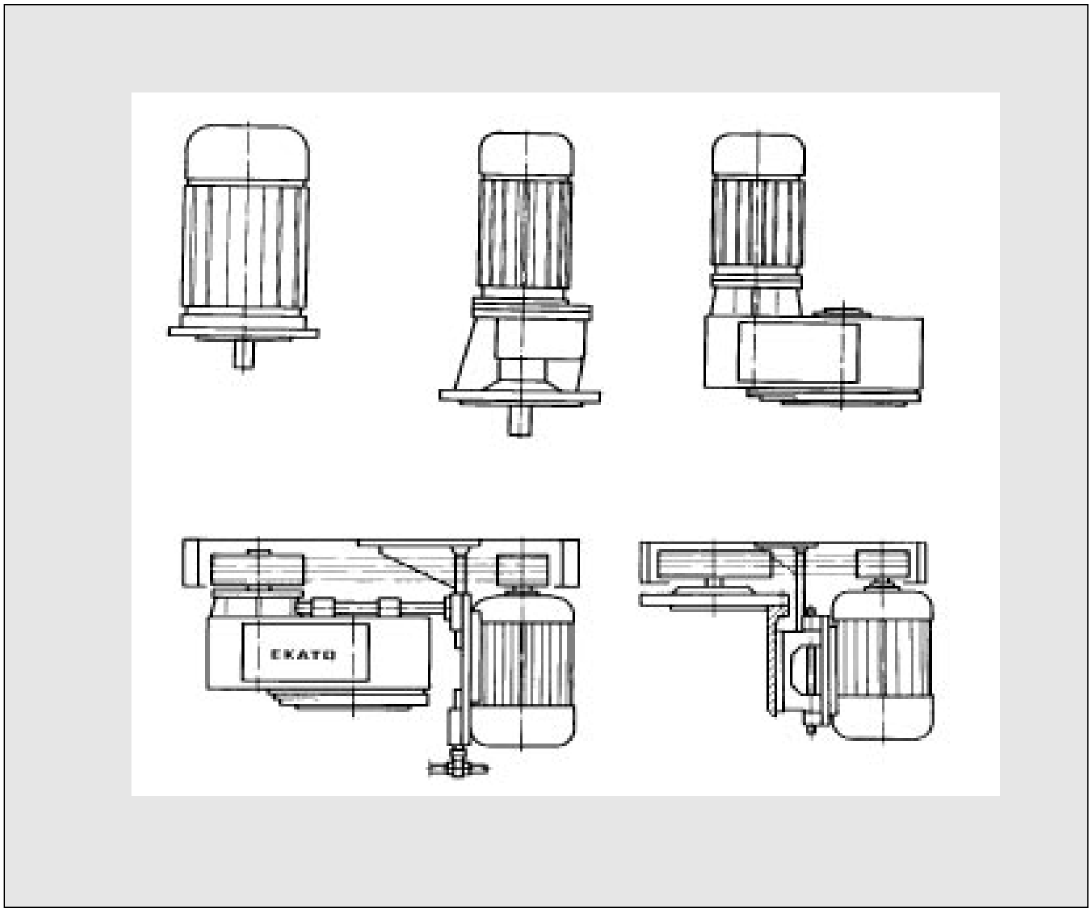


Figure 5. Different Types of Drives.

(maintenance can be done with little work by making changes in the hierarchies of components or in the databases [was often necessary during development]), (3) changes in the configuration knowledge (such changes seldom happen, and maintenance can only be done by editing Lisp code), (4) new types of component (objects have to be added to the hierarchies, configuration knowledge has to be coded, and related constraints have to be added to the constraint network, requiring experience with the COMIX architecture and the use of good programming skills, which experience has shown is also feasible).

During the last year, the system was extended by EKATO to include a new series of mixing machines and the ability to use a larger number of different impeller types. EKATO plans to develop an interface to the computer-aided design system AUTOCAD, which is used in the Construction Department.

Time will show whether the chosen architecture will lead to a long life for COMIX.

References

Brinkop, A., and Laudwein, N. 1992. Configuration of Industrial Mixing-Machines—Development of a Knowledge-Based System. In *Proceedings of the Fifth International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-92),* 432–440. Paderborn, Germany: Springer.

Cunis, R.; Günter, A.; Syska, I.; Peters, H.; and Bode, H. 1989. PLAKON—An Approach to Domain-Independent Construction. In Proceedings of the Second International Conference on Industrial and Engineering Applications of Artificial Intelligence and Expert Systems (IEA/AIE-89), 866–874. New York: Association of Computing Machinery.

Cunis, R.; Guenter, A.; and Strecker, H. eds. 1991. *Das PLAKON-Buch* (The Book of PLAKON). Informatik Fachberichte 266. Berlin: Springer Verlag.

de Kleer, J. 1986. An Assumption-Based TMS. *Artificial Intelligence* 28:127–162.

Doyle, J. 1979. A Truth Maintenance System. *Artificial Intelligence* 12:231–272.

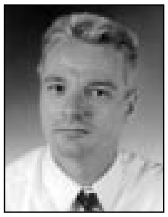
Marcus, S., and McDermott, J. 1989. SALT: A Knowledge-Acquisition Language for Propose-and-Revise Systems. *Artificial Intelligence* 39:1–37.

Marcus, S.; Stout, J.; and McDermott, J. 1988. VT: An Expert Elevator Designer That Uses Knowledge-Based Backtracking. *AI Magazine* 9(1): 95–112.

Sriram, D. 1987. ALL-RISE: A Case Study in Constraint-Based Design. *Artificial Intelligence in Engineering* 2(4): 186–203.

Stallman, R. M., and Sussmann, G. J. 1977. Forward Reasoning and Dependency-Directed Backtracking in a System for Computer-Aided Circuit Analysis. *Artificial Intelligence* 9:135–196.

Steinberg, L. I. 1987. Design as Refinement Plus Constraint Propagation: The VEXED Experience. In Proceedings of the Sixth National Conference on Artificial Intelligence, 830–835. Menlo Park, Calif.: American Association for Artificial Intelligence.



Axel Brinkop is a scientific collaborator at the Fraunhofer Institute for Information and Data Processing in Karlsruhe, Germany. He recieved his Masters in computer science in 1989 from the University of Kaiserslautern. Since 1989, he has been working at the Fraunhofer Institute in the Department of Expert Systems for Technical Applications. From 1990 to 1992, he worked on COMIX, a system for the knowledge-based design of industrial agitators. His current work concentrates on the modeling of application domains for simulation- and model-based diagnosis.

Norbert Laudwein is a scientific collaborator at the Fraunhofer Institute for Information and Data Processing in Karlsruhe, Germany. He recieved his Masters in computer science in 1989 from the University of Kaiserslautern. Since 1989, he has been working at the Fraunhofer Institute in the Department of Expert Systems for Technical Applications. From 1990 to 1993, he worked on COMIX, a system for the knowledge-based design of industrial agitators. In 1993, he (with Axel Brinkop) received the Fraunhofer Prize for the development of this system.

Rüdiger Maassen is an employee of EKATO Rühr-und Mischtechnik GmbH in Schopfheim, Germany. He studied process technology at Düsseldorf, Germany, where he received his bachelors in process technology (Verfahrenstechnik) in 1989. Since 1989, he has been working at EKATO, first in the Research and Development Department and then in the Data Processing Department. Since 1990, he has been working on COMIX, a system for the knowledge-based design of industrial agitators. His current work concentrates on the maintenance and extension of this system.