

VOLUME 27, Number 1
September 1995

ISSN 0166-3615
CINUD4 27(1) 1-94 (1995)

COMPUTERS IN INDUSTRY

An international application oriented research journal

Editor: Prof. H. Wortmann



COMPUTERS IN INDUSTRY

Volume 27, 1995

Editor-in-Chief
Hans Wortmann

Co-editor North America
Colin L. Moodie

Co-editor Industrial Acquisitions
Gloria Karlmark



ELSEVIER

Amsterdam – Lausanne – New York – Oxford – Shannon – Tokyo

Configuring computer systems through constraint-based modeling and interactive constraint satisfaction

S.M. Fohn^b, J.S. Liau^{a,*}, A.R. Greef^b, R.E. Young^a, P.J. O'Grady^a

^a Group for Intelligent Design in Manufacturing, Department of Industrial Engineering, North Carolina State University, Raleigh, NC 27695-7906, USA

^b IBM Corporation, Network Application Services Division, Thornwood, NY 10594, USA

Received 22 November 1993; revised 22 September 1994

Abstract

A major problem in the computer industry, as with many other industries, is ensuring that the products ordered by a customer are viable products and that they can be delivered at the quoted price. The inability of companies to solve this problem results in an enormous expense for a company. In this paper we present an approach to solving this problem in the computer industry through constraint-based modeling and interactive constraint satisfaction. PC/CON, a Personal Computer Configuration System, has been built within a computer-based environment called Saturn. The development of PC/CON and related work is discussed. Saturn's architecture is presented and an example session with PC/CON is demonstrated. We show how Saturn and PC/CON have the features and capabilities necessary to support the extremely large variety of products and to support the maintenance of a configuration system for a rapidly and continuously changing product line. This approach would also appear to be applicable to other industries where there is a wide product variety.

Keywords: Computer configuration; Interactive constraint modeling; Constraint-based reasoning; Concurrent engineering

1. Introduction

A major problem in the computer industry, as with many other industries, is ensuring that the products ordered by a customer are viable products and that they can be delivered at the quoted price. The computers must be configured so that they can be readily manufactured and that they can support the desired software. The software must be configured for compatibility between hardware and other software so as to achieve good performance. Al-

though a seemingly structured problem, the variety of options available for a computer system allows for billions of feasible product configurations making this problem difficult to solve consistently.

Trilogy [1] estimates that in the personal computer (PC) industry, between 30 and 85 percent of all orders are incorrectly configured. The result is an enormous expense for a company, especially when realizing that infeasible product configurations made at the point of sale may only be discovered much later in manufacturing, or, worse yet, after delivery to the customer. The consequences of late discovery include having to change a customer's order (thereby delaying manufacturing and increasing the lead-time),

* Corresponding author.

forcing a customer to return a shipment of faulty orders, and the loss of sales. Furthermore, products that are incorrectly configured are most likely incorrectly quoted. As a result, a manufacturer may be required to give away free components which were omitted from the sales quote yet are needed to supply the customer with the promised functionality. The costs involved, both monetary and the loss of customer good will, can be very high.

In order to reduce the impact from the sale of infeasible products, companies have attempted to include product configuration checks both at the point of sale and between the point of sale and start of manufacture. Configuration checking is performed either manually or automatically by computer programs. Programs like XSEL [2] and Trilogy [1] check orders at the point of sale while XCON [3] and Proof-Plan [4] check orders after a sale and before manufacture. The effectiveness of point-of-sale checking is dependent on the experience of the sales representatives and on the variety of the customers. An inexperienced sales representative dealing with a large number of different concerns from a diverse clientele may not be able to answer all questions immediately. This results in postponing sales until more information can be gathered to address the needs of a client. With post-sale checking, any detected configuration conflicts are resolved by calling customers and essentially cycling back to the point of sale before a valid work order is released to manufacturing. This cyclic approach can have a negative affect on a customer who is forced to endure long lead times, frustrating product configuration changes, and the prospect of returning products that do not meet customer expectations.

The computer product configuration problem can therefore be summarized as follows:

- 30% to 85% of all orders are incorrectly configured.
- Late discovery of configuration errors increases manufacturing lead times and produces incorrect orders.
- Configuration errors can result in a loss of confidence by customers and can result in a loss of sales.
- Incorrectly configured products can be quoted inaccurately.
- The configuration problem can be very large,

taking it beyond the bounds of human ability to solve.

A system that solves the computer configuration problem must address these issues.

We present an approach that addresses the issues and solves the computer product configuration problem in the computer industry through constraint-based modeling and interactive constraint satisfaction. "Computer Product Configuration" is defined as the generation of a computer hardware and software configuration that meets the price and performance constraints imposed by a customer, and the compatibility constraints between hardware and software. This approach aims at avoiding the cycling between a manufacturer and its customers and focuses on interactively assisting order taking from a diverse clientele. It furthermore is capable of supporting a diverse number of personnel in fields such as manufacturing and marketing all of whom have an interest in computer product configuration. To accommodate this approach, we also introduce a Constraint Modeling System called Saturn, which is an interactive constraint satisfaction system that can be used to implement a computer configuration system. Saturn implements a logic-based, ATMS-supported, interactive constraint satisfaction system tightly coupled with a relational database [5]. It has a spreadsheet-like development interface that, in conjunction with a database manager, facilitates the construction of a constraint-based model representing a product line of feasible computer hardware configurations, supported software configurations, and pricing information.

In this paper, the development of a computer product configuration system is discussed, beginning with the salient features and capabilities that a computer-support environment for computer configuration should have. Next the related work on configuration systems that have addressed similar problems are presented and compared to our constraint modeling approach. This is followed by a discussion on the nature of the computer product configuration problem and a brief description of Saturn's architecture. In this context, the implementation of PC/CON, which is a personal computer configuration system built within Saturn, as a constraint-based model is shown. An example session with PC/CON is then described to show how Saturn's interactive constraint satisfac-

tion capabilities are harnessed by PC/CON to perform PC configuration. Finally, it is shown how Saturn and PC/CON have the features and capabilities necessary to support computer product configuration and to support the maintenance of a configuration system for a rapidly and continuously changing product line.

2. Embracing the task of computer product configuration

Product configuration is a critical yet a complex task in most industries. This is particularly true for the computer industry. To deliver a viable product, the computers not only need to meet the customers' functional requirements, they also must be configured so that they can be readily manufactured and they possess compatibility between hardware components and between hardware and the desired software. Although a seemingly structured problem, the variety of options available for a computer system allows for billions of feasible product configurations, making this problem difficult to consistently solve. As an illustration, we explore the computer product configuration problem in the personal computer (PC) industry. Compared to a supercomputer or a minicomputer, a PC typically has a simpler operating system, a smaller amount of memory and primary storage space, limited input and output devices, and is a single user system. However, as we will show, the product configuration problem for a PC is still a complex and challenging task.

In earlier years, manual PC product configuration was viable. This is readily seen by examining the number of feasible hardware configurations possible for the IBM model 25, with an 8086 8 MHz processor and an 8 bit PC bus architecture. This machine had only 240 feasible configurations and could run all software products. Today, consistently configuring PC orders manually is nearly impossible. The IBM model 95, with an 80486 33 MHz processor and a 32 bit MCA bus architecture has 467 billion feasible hardware configurations without considering any of the 115 cards that could plug into any of the 8 slots in its backplane. Table 1 contains the number of feasible product configurations for the PS/2 product line for 1991/1992 and for 1993. The numbers are computed from [6] and [7].

Table 1

The number of feasible configurations for the PS/2 product line

Model	Number of configurations	
	1991/1992	1993
25-086	240	240
25-286	768	768
30	21,760	11,520
35	145,317,888	238,132,224
55	835,584	2,151,552
56	235,929,600	2,615,328,768
57	115,605,504	308,625,408
70	258,048	1,347,840
80	23,181,312	131,843,712
90	5,138,022,400	2,082,103,296
95	31,085,035,520	467,362,415,520
Total	36,744,208,624	472,741,960,848

From the table, we can see that the number of configurations increased by more than 400 billion in just one year. The explosion of software products and their inherent incompatibility problems and platform-specific requirements has added to the complexity of computer product configuration. Compounding the problem is the weekly release of new products and changes to existing products. Major product introductions that require significant restructuring of the entire PC model line are now occurring in a 3 month cycle. In the PC industry it is not only essential to support the task of PC product configuration but also to support the development and maintenance of the information upon which that support is based. This calls for a configuration environment that can accommodate growing product complexity stemming from technological progress and from consumer demand.

To meet this challenge we consider computer product configuration to be a highly interactive, non-monotonic, step-wise process. A user can specify any permutation of product/price attribute values. The output is made up of consistent values for locally related attributes whose values were previously unspecified or of feedback information for resolving inconsistent partial configurations. Once all of the attributes have values and there are no inconsistencies among them, the computer product configuration is a valid working product that can be readily manufactured.

This research is focused on embracing the task of computer product configuration with respect to two aspects of the problem. First, is the development of a computer tool that allows product configuration systems to be built by non-specialists and exercised by personnel with diverse skills. Second, is the creation of a model that can represent the complex interrelationships among its elements while retaining an organizational simplicity such that people with no specific training in a computer-related field can perform development, maintenance, and expansion. The emphasis placed on the above research topics are a response to the new character of contemporary, highly volatile markets and have not yet been addressed by other researchers in the field.

3. Related work in automated product configuration

The most well known computer system used for sales support is XSEL and for computer product configuration is XCON (R1). Both are rule-based expert systems used by Digital Equipment Corporation to configure their product set [2,3,8]. XCON ensures the technical accuracy of customer orders and directs the assembly of those orders. XSEL is used interactively to select saleable items for a customer's order and provides pricing information. PRIDE [9] and COSSACK [10] are frame-based expert systems which were built to support the design of paper handling systems inside copiers and to configure micro-computer systems. COSSACK was adapted from the PRIDE system specifically to solve the computer configuration problem. Lowe [4] proposes the technique of proof-plans, a technique used in induction-based theorem-proving systems, to check orders previously taken by sales representatives. Trilogy [1] offers an object-oriented package called the SalesBUILDER to support computer product configuration at the point of sale.

Technically, all of these systems can perform computer product configuration. What we are interested in, however, is their ability to support the computer-product configuration task from our perspective. As discussed above, first, do they support the construction of product configuration systems by non-specialists that can be exercised by personnel with diverse skills. Second, do they represent the

complex interrelationships among the elements of the model while retaining an organizational simplicity such that people with no specific training in a computer-related field can perform development, maintenance, and expansion. These are capabilities that we consider to be of prime importance for application in the computer industry.

XSEL and Trilogy somewhat support computer-product configuration as outlined in Section 2 in that they are employed at the point of sale and not after an order has been taken as in XCON and Lowe's proof-plans. The degree of flexibility provided to the user of these products is not clear from the literature. However, since XSEL is a rule-based system and Trilogy is an object-oriented system, their model (knowledge) must comprise a hierarchical representation of all of the solution paths possible for all feasible configurations. These approaches to modeling tend to impose one particular problem solving approach on a user that must be strictly adhered to at all times. This does not accommodate the need to support a number of diverse users with varying degrees of experience [11]. A novice sales representative may need a chronological paradigm to help determine a computer product configuration. But more experienced users will use both their product experience and their experience with the configuration support system to bypass many of the chronological steps made redundant through their experience. By supporting such diverse users, the usefulness of a configuration support system is ensured both as an aid for novices configuring computer products and as a configuration checker for more experienced users. This spectrum of support is obtained through the use of constraint-based modeling, interactive constraint propagation, and dynamic interaction with a relational database.

Again considering the volatility of the computer market, it is evident that configuration applications required today may be obsolete in as little as one year's time. In the interest of time and money, it is better to train personnel from the field in information modeling and the use a software package than to train Artificial Intelligence (AI) specialists in the field. Hence, an environment is demanded that will enable rapid model building by product and manufacturing engineers, and sales representatives. This approach does not appear possible with rule-based

expert systems as seen by DEC's XSEL and XCON, both of which are burdened by their size and complexity. Together, these systems have just under 14,000 rules, require a staff of 60 to maintain, and require a programmer/analyst to serve a 12 month apprenticeship in order to become proficient at updating the system [2]. A similar level of proficiency is required to construct an initial product configurator in Trilogy as it can take up to 12 months to complete. COSSACK and Lowe's proof-plans have no measure of the time it takes to build configuration applications as this was not their focus of interest; but, based on their implementation descriptions, they appear to have similar limitations. Rapid development of configuration applications such as PC/CON results from two implementation aspects. First, the large quantity of rapidly changing yet stable interrelationships between product components were modeled in a relational database. Second, the slower changing and more general constraints were modeled in Saturn. These results are presented later in the paper.

One of the most important issues in computer product configuration is the maintainability and expandability of existing configuration information models. The results from XSEL and XCON show that maintenance of these systems requires an enormous effort. In addition to the specialized personnel involved, it is extremely difficult to ensure that the alteration or addition of a rule does not inadvertently create a contradiction with another rule somewhere else in the system. Thus, extensive testing must be done to ensure that rule changes maintain system consistency.

Trilogy hard-codes configuration rules and constraints as methods that are part of objects in a class hierarchy. Trilogy emphasizes the ease of system maintainability in their approach since most new products can be added to the configurator in terms of the existing products. Yet it is when the existing products change and when new products, independent of the existing ones, are introduced that maintaining and updating the system becomes an issue. Such change is not nominal since it involves rewriting methods without introducing any inconsistency into the system, a difficult task without a consistency checking mechanism. Although Trilogy has stated that the product information could be stored in a

relational database and accessed by their SalesBUILDER, they have not yet built such a system. Consequently, they could provide no estimate as to the time required to build the system. In addition, the cost of SalesBUILDER seems high at an average cost of \$1 million per customer [12], not including maintenance cost and system updates.

Maintenance issues are difficult to assess in PRIDE and COSSACK since these systems were never used in a production environment nor was maintenance specifically addressed in their prototype form. However, since the product information is hard-coded in the frames and class hierarchy, changes to the product information would require altering or rebuilding the basic system structure. This again requires an individual with advanced training and skill in artificial intelligence to be involved in the tedious process associated with the model maintenance. Maintenance is similarly difficult to assess in Lowe's proof-plan system but since the issue was not addressed, it was most likely not solved.

In summary, the above systems, while all able to support configuration in some form, fall short of dealing with a spectrum of users and attending to the needs of a diverse market in the computer industry which is characterized by a continuously changing and expanding product line. This becomes more apparent when considering the nature of computer product configuration.

4. The expansive nature of computer product configuration

The most distinguished characteristics of computer product configuration in the computer industry are change and the number of different perspectives which should be represented in the configuration system. Change is found in all aspects of the industry: products, prices, discounts for special configurations available for a limited time, software applications, components, and the market focus. Such rapid and continuous change affects many areas in a company, and each area can view the configuration problem from their own perspective. Each perspective places a different emphasis on what aspects are considered to be more important in a configuration. In summary, both the change and the diverse inter-

perception of a constraint is any interrelationship among elements of a model; for example, a constraint can be a rule, equation, relation, rows in a database table or any data structure. Saturn provides an environment that facilitates rapid model building and database interfacing, and provides user assistance when attempting to determine a feasible solution to these constraint-based models. Constraint satisfaction is achieved through automatic theorem proving techniques that operate with propositional default logic and order-sorted, first-order predicate logic (FOPL) [14]. Since Saturn supports user interactive constraint satisfaction, constraint variables can receive values in either of two ways: user assertion or value propagation. Through the selection of suitable variable values, Saturn will propagate values to other interrelated variables. If variable values are selected such that they are inconsistent with the constraints, then the affected constraints are said to be violated. The minimum set of variables that contributed to the violations are determined and presented to the user who then attempts to alleviate the violations either by adjusting the constraints or by selecting alternative variable values.

Consequently, the nature of computer product configuration cannot be seen simply as a process of combining compatible components into feasible configurations. It must be seen as an interactive, dynamic process whereby answers to a large number of diverse questions from different perspectives can be found in the context of continuously changing product lines and product information. This demands that a product configuration support environment must have an architecture to accommodate these characteristics. The Saturn constraint-system shell and its integrated coupling to a relational database provides a unique architecture from which such environments can be built.

Saturn is a constraint-system shell in which problems can be modeled as constraints and exercised through interactive constraint satisfaction [13]. Our



Fig. 1. The *constraint-sheet interface* serves as the environment interface. Each cell in the sheet may contain either a domain, a logic constraint, a macro, a constraint-sheet formula, or a text string that can be used to annotate an application. A cell with a domain represents a constraint-variable and may also have a name and value. Cells containing logic constraints (logic sentences may contain conjunction, disjunction, implication, etc.) are displayed with respect to their evaluation state denoted by #T (*true*), #F (*false*), and #U (*unknown*). Cells containing Prolog programs and database access declarations are reflected in the constraint-sheet with the symbol #MAC (Macro). Cells containing constraint-sheet formula display the evaluation result of the formula. These sheet elements are entered into the constraint-sheet through the cell editor. The cell editor is cell specific and accommodates model development by providing a mechanism through which the sheet's contents are entered and altered.

A cell's domain defines the range of values that can be accommodated by the cell. Domains may be explicitly enumerated in any constraint-sheet cell or defined over a column in a database. When a domain is entered into Saturn, the *well-formed formulae (WFF) compiler* performs a number of checks to ensure that the domain is syntactically correct and that any references either to the database or a domain defined in another cell, are correct. When a value is entered into a cell, the WFF compiler performs a syntax check to ensure that the value either exists in an existentially defined Saturn domain or in a referenced database table's column. All database interaction is performed via the *Relational Database Interface* as shown in Fig. 1. Similar syntax checking is performed whenever a logic constraint, macro, or constraint-sheet formula is entered into the constraint-sheet. All syntax errors are reported to the user and all well formed, domains, values and constraints are translated internally into sentences of an order-sorted, first-order, predicate logic theory (see [15] for the formal theory).

Variable values and normalized constraints are passed on to the *constraint engine* where they are used during theorem proving. The constraint engine implements hyper-resolution, refutation-based resolution, mathematical equation evaluation and database query evaluation to prove the constraints as either

satisfied, or unsatisfied, and to perform value propagation. Satisfied constraints are those that have been determined to be *true*. Conversely, violated (or unsatisfied) constraints are those that have been determined to be *false*. If the constraint engine is incapable of determining the satisfiability or unsatisfiability of a constraint, the constraint remains *unknown*. Constraint propagation is achieved if variable values can be found by the constraint engine such that *unknown* constraints can be proved *true*. Database constraints are first transformed into equivalent SQL (Structured Query Language) sentences by the constraint engine's theorem prover. Theoretically this is achieved by taking a proof-theoretic view of the relational DBMS as discussed extensively by Reiter [16]. These constraints are then passed to the relational DBMS query processor, via the tightly coupled *Relational Database Interface*, for evaluation. The other constraints and macros are used by the standard constraints for data processing purposes.

The *Assumption-Based Truth Maintenance System* (ATMS) [17,18] is a non-monotonic theorem prover based on propositional logic that functions as an archive for all inferences performed and contradictions found by the constraint engine. The archived dependencies allow a user to view constraint violations and resolve conflicts by *rolling back* to an application's previous search state. Additionally, the ATMS can provide justifications for all variable values by presenting the solution path from which a cell's value was determined. It is this component of Saturn, together with the visual truth-values denoting the state of the constraints in the constraint-sheet that provides feedback to a user when performing constraint satisfaction.

The above architectural description of the Saturn constraint-system shell provides an overview of the features and capabilities available for performing constraint-based modeling and interactive constraint satisfaction. Application developers are supported by a familiar spreadsheet-like interface that enables more emphasis to be placed on modeling and not on programming nor on the difficulties of logic theory. The tightly coupled database interface enables large constraint models to be built in a structured manner while still retaining the organizational simplicity of interrelated tables. In addition, Saturn's ability to generate dynamic SQL enables constraint models to

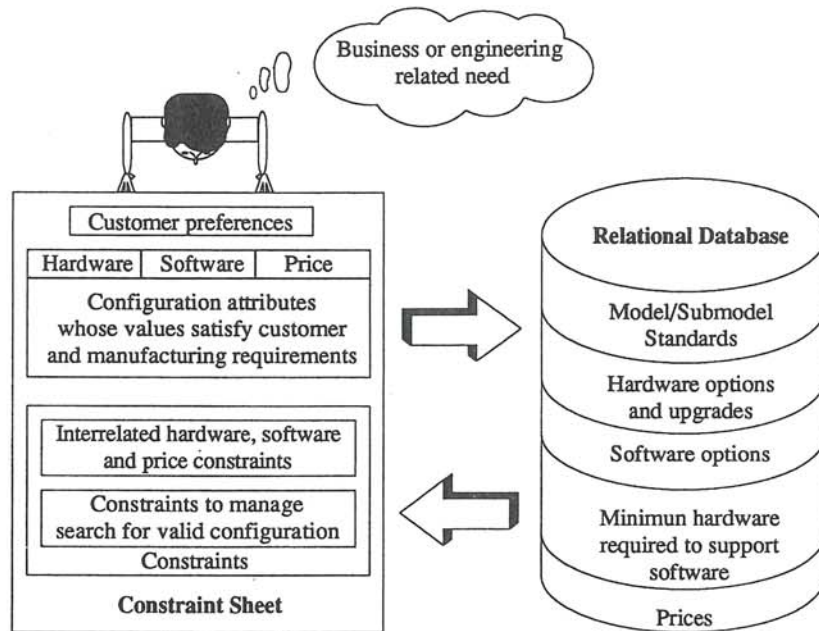


Fig. 2. Architecture of the PC/CON constraint-based model.

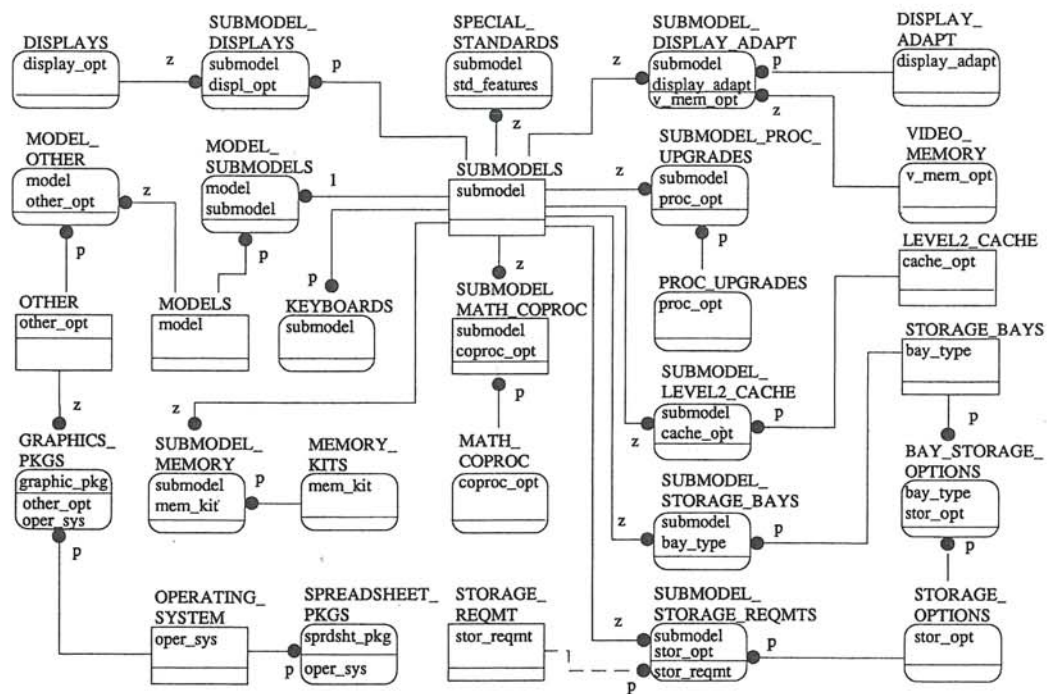


Fig. 3. Entity-relationship model for PC/CON.

be built that represent the feasible interrelationships between elements of the model. This contrasts with other approaches that hard-code all search paths representing feasible configurations. Saturn is able to dynamically alter its search space leading to a significant reduction in the size and complexity of models [19]. A diverse number of users are supported by Saturn's use of a constraint sheet in conjunction with the non-monotonic ATMS. This combination gives users the flexibility to approach a problem's solution from an extremely wide number of directions. The constraint sheet allows a user to begin at any one or number of points in the problem and the ATMS enables a user to fall back to any previous point in the problem. In the following section we demonstrate the utility of Saturn by showing how a computer configuration system is implemented in Saturn and a relational database.

Saturn, and its predecessors [20], have been applied to a number of concurrent engineering tasks, including design for assembly [21], rotational parts [22,23], testability of electronics assemblies [24], process planning for printed circuit boards assembly [25], and the design of automated storage and retrieval systems [26]. Extensions have been proposed in the areas of analysing large-scale systems [27], handling imprecision [28], using genetic algorithms [29], and hierarchical modeling [30].

6. Implementing PC/CON in Saturn and a relational database

In this section, we present an application called PC/CON that is implemented in Saturn and a relational database to demonstrate that Saturn can be used to build an interactive computer product configuration system to accommodate computer product configuration tasks. PC/CON has the features and capabilities necessary to support sales representatives in personal computer product configuration and to support the maintenance of a configuration system for a rapidly and continuously changing product line.

Fig. 2 depicts the structure of PC/CON as it is implemented in Saturn and a relational database. The relational database contains product data taken from the IBM Sales Advisor [6,7] catalogue and reorganized into the information model represented by the IDEF1X entity-relationship diagram shown in Fig. 3.

The database holds all computer model and sub-model, hardware options and upgrades, software options, minimum hardware requirements to support certain software, and all pricing information. This data represents specific interrelationships (constraints) between the elements of the information model and can be frequently updated using OS/2's Database Manager. At present, the database holds the following interrelationships:

- Physically compatible components (i.e., components that can be assembled into a product).
- Products and components that give specific hardware functionality.
- Products and components that give specific software functionality.
- Standard offerings that contain specified components and configurations.
- Catalogue offering that contain specified components and configurations.

Part of the constraint sheet shown in Fig. 4 holds the final computer configuration as groups of attributes that display a hardware configuration, software configuration, and price that satisfy both customer and manufacturing requirements. Another region of the constraint sheet holds constraints that dynamically and interactively manage the process of finding a valid computer configuration (refer to [5] for specific details of the syntax and operation of Saturn's constraints). These constraints comprise two groups. First, there are the constraints that manage the search space. Second, there are the constraints that monitor the development of the hardware and software configuration together with price. The second group of constraints are generic in nature and are not frequently updated as they do not hold much specific product data but rather define weak interconnections among clusters of product data. Together, the two group of constraints provide the following functionality:

- Ensure that selected hardware requirements can support the selected software requirements.
- Check each component's pre-and co-requisites to identify their interdependencies and conflicts.
- Identify and infer missing information.
- Identify recommended and approved options.
- Based on customer preferences of hardware functionality and/or software functionality, retrieve a basic submodel from the product database.

- Retrieve standard features of a basic submodel from the product database.
- Make all non-available options inaccessible.
- Limit all the user options to avoid error input.
- Build up a customized system configuration based on user options.
- Infer required components based on user options.
- Generate and update pricing information as a configuration evolves.
- Check the compatibility between components of the system.
- Check the compatibility between software applications and the configured system.
- Provide advice to resolve conflicts when an incompatible configuration exists.
- Ensure the customized configuration is a valid configuration.

The customer preference attributes, shown on the constraint sheet in Fig. 4, illustrate one possible approach to computer product configuration. A typical approach taken by a sales representative would be to start with any combination of one or more

customer preferences in order to determine a standard computer that can then be loaded with options. This approach shows how a user can take advantage of navigational techniques so as to give structure to the product configuration process. This type of functionality is usually incorporated into a front-end, window interface that incorporates scroll bars, drop-down selection menus, spin buttons, and contextual help. The following section presents an example session with PC/CON that capitalizes on the above approach to finding a computer configuration that meets a customer's preferences.

7. An example PC / CON session

The Saturn approach to computer product configuration views the configuration task as an interactive constraint satisfaction problem where the decision variables and their interrelationships are modeled as constraints. Since the system is user-interactive, variables receive their values through constraint value

PCCON.SWK			
File Edit Cell Utilities Connect Close Help			
		B	
		PERSONAL COMPUTER CONFIGURATOR	
CUSTOMER PREFERENCES		TOTAL PRICE (\$): 0.00	
Target Price:	4000.00	xxxxxx	
Software to use:	"designer"	xxxxxx	
Choice of Processor:		xxxxxx	
Choice of Memory (KB):	4000	xxxxxx	
Choice of Hard Disk (MB):	80	xxxxxx	
Choice of LAN Capability:		xxxxxx	
Choice of Graphics Adapter:	"vga"	xxxxxx	
Start to retrieve?		xxxxxx	
MODEL STANDARDS		Basic Price(\$):	
Model		xxxxxx	
Submodel		xxxxxx	
Processor		xxxxxx	
Std Memory (KB)		xxxxxx	
Math Coprocessor		xxxxxx	
Internal Cache (KB)		xxxxxx	
Std Hard Disk (MB)		xxxxxx	
Std Floppy Drive		xxxxxx	
Keyboard		xxxxxx	
LAN Adapter		xxxxxx	
Graphics Adapter		xxxxxx	
No. of Serial Port		xxxxxx	
No. of Parallel Port		xxxxxx	
No. of Display Port		xxxxxx	
No. of PT Devic Port		xxxxxx	
Max Memory (KB)		xxxxxx	
Max Hard Disk (MB)		xxxxxx	
		HARDWARE OPTIONS	
		Sub Total (\$): 0.00	
		Memory Kit 1	
		Memory Kit 2	
		Memory Kit 3	
		Memory Kit 4	
		Memory Kit 5	
		Memory Kit 6	
		Optional Math Coprocessor	
		Processor Upgrade	
		Optional Level 2 Cache	
		3.5"HH Bay (1)	
		3.5"HH Bay (2)	
		5.25"HH Bay	
		5.25"HH/3.5"HH Bay (1)	
		5.25"HH/3.5"HH Bay (2)	
		5.25"FH/3.5"HH Bay	
		5.25"SL or 3.5"SL Bay	
		3.5"HH Bay for Model 25	
		3.5"HH Bay for 286 Machines	
		Hardfile Kit	
		Drive Kit	
		Tape Backup Kit	
		CD Rom Kit	
B5: Value: bf:software dom: target software			
AutoP			

Fig. 4. Configuration's attribute region in the PC/CON constraint sheet.

propagation and by the user directly assigning values to them. Saturn also allows users the flexibility to start anywhere in the problem space. This means that a user can start by assigning values to any of the decision variables at any time. Value propagation ensures that the configuration will be automatically adjusted to support the starting point chosen by a user. A user can assign certain values to variables solely to view the propagation and the effect that the variable's value will have on a configuration. Multiple constraint sheets can be opened and worked on concurrently, permitting a user to simultaneously compare several configurations. This example demonstrates only a subset of Saturn's and the PC/CON application's flexibility and concerns itself with the approach that would typically be taken by a sales representative interested in the following:

- *Order validation.* Determine that the system ordered by a customer can indeed be manufactured.
- *System configuration.* Begin with either a standard PC model or offering, and then add compo-

nents to create the computer system desired by the customer.

- *System pricing.* Determine the total price of the computer system taking into account such things as sales tax, customer discounts and mailing costs.
- The sales representative starts by asking for the customer's preferences which are then entered into the cells at the top left region of PC/CON's constraint sheet shown in Fig. 4. This figure categorizes the constraint variables for Customer Preferences, Total Price, Model Standards, and Hardware Options. Not shown in the figure is the lower part of the constraint sheet with the variables for the Software Options, Display Options, and Miscellaneous Options. In a typical case, the sales representative may input a value for the variable "Target Price for Basic Model" and choose any combination of other options from the Customer Preferences section to retrieve a submodel. If a user does not wish to specify any other preference, any one of these options can be omitted. As an illustration, consider a user who

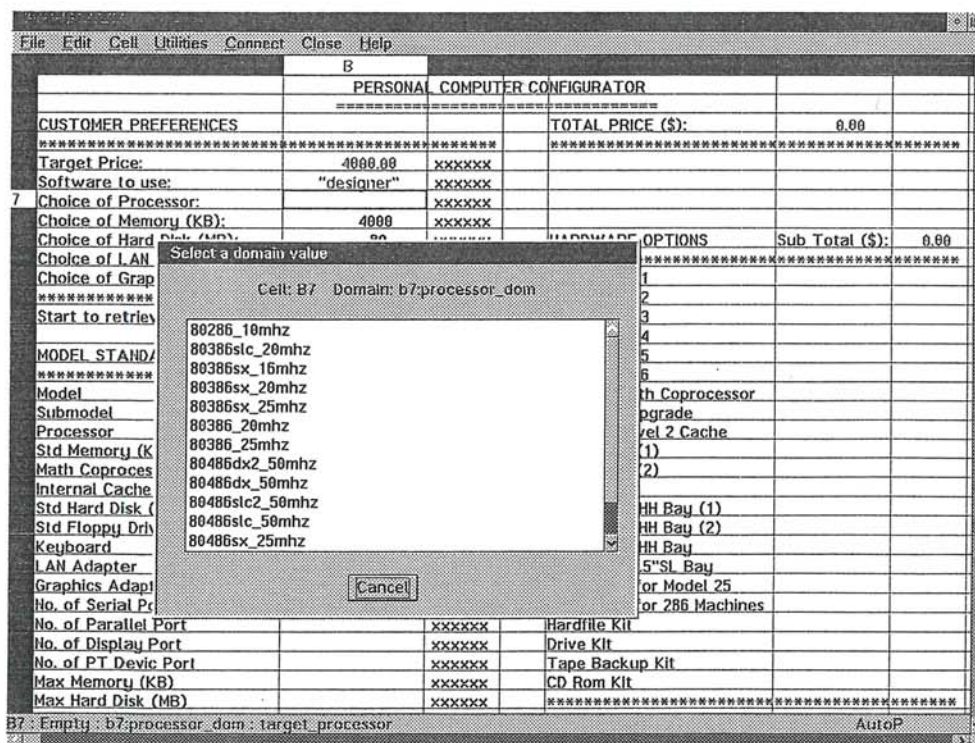


Fig. 5. Selection of a customer's preferred microprocessor.

inputs \$4000 for the variable "Target Price for Basic Model" and chooses *Designer*, a drawing package from Micrografix Inc., for the variable "Software to use". The system will respond by accessing the database to retrieve the minimum functional requirements for *Designer*, and by asserting those values to the variables "Choice of Memory", "Choice of Hard Disk", and "Choice of Graphics Adapter" as shown in Fig. 4.

The sales representative can now add to the customer's preferences by choosing the variable "Choice of Processor". The value for this variable is restricted to exist over a domain specified in a column of a database table in the external database. The domain is defined in Cell B7 as "PC.SUBMODEL.PROCESSOR" and reflects the fact that column "PROCESSOR" is selected from database table "SUBMODEL" in the external database "PC". The database table "SUBMODEL" is also seen as an entity in the information model of Fig. 3. A sales representative assigns a value to the variable "Choice of

Processor" by using a pull down menu located at the top of the constraint sheet. The selection of the pulldown menu and the "Give Value" option from it initiates an access to the "PC" database to locate the table "SUBMODEL" and retrieve all the values from the "PROCESSOR" column. The values are presented in a window, as shown in Fig. 5, from which the sales representative makes a selection. These values are the existing processor options in the database. By using a database domain definition for all the variables in the system, the sales representative is never allowed to assign a value to a variable that would produce an invalid configuration.

Continuing with the example, after choosing 80386sx_20mhz, the sales representative initiates value propagation by selecting yes for the variable "Start to retrieve". Once value propagation is initiated, the system accesses the database to find the submodels which meet the customer's preferences. The submodels are displayed in a window for the sales representative to select in a manner similar to

PCCON.SWK			
File Edit Cell Utilities Connect Close Help			
D			
PERSONAL COMPUTER CONFIGURATOR			
CUSTOMER PREFERENCES		TOTAL PRICE (\$):	1755.00
Target Price:	4000.00	xxxxxx	
Software to use:	"designer"	xxxxxx	
Choice of Processor:	"80386sx 20mhz"	xxxxxx	
Choice of Memory (KB):	4000	xxxxxx	
Choice of Hard Disk (MB):	80	xxxxxx	
Choice of LAN Capability:	"none"	xxxxxx	
Choice of Graphics Adapter:	"vga"	xxxxxx	
Start to retrieve?	yes	xxxxxx	
MODEL STANDARDS		Basic Price(\$)	1755.00
17 Model	"57"	xxxxxx	
Submodel	"57-049"	xxxxxx	
Processor	"80386sx 20mhz"	xxxxxx	
Std Memory (KB)	4000	xxxxxx	
Math Coprocessor	"none"	xxxxxx	
Internal Cache (KB)	0	xxxxxx	
Std Hard Disk (MB)	160	xxxxxx	
Std Floppy Drive	"2.88mb 3.5i"	xxxxxx	
Keyboard		xxxxxx	
LAN Adapter	"none"	xxxxxx	
Graphics Adapter	"vga"	xxxxxx	
No. of Serial Port	1	xxxxxx	
No. of Parallel Port	1	xxxxxx	
No. of Display Port	1	xxxxxx	
No. of PT Devic Port	1	xxxxxx	
Max Memory (KB)	16000	xxxxxx	
Max Hard Disk (MB)	960	xxxxxx	
		HARDWARE OPTIONS	Sub Total (\$): 0.00
		Memory Kit 1	
		Memory Kit 2	
		Memory Kit 3	"na"
		Memory Kit 4	"na"
		Memory Kit 5	"na"
		Memory Kit 6	"na"
		Optional Math Coprocessor	
		Processor Upgrade	
		Optional Level 2 Cache	na
		3.5"HH Bay (1)	"na"
		3.5"HH Bay (2)	"na"
		5.25"HH Bay	"na"
		5.25"HH/3.5"HH Bay (1)	
		5.25"HH/3.5"HH Bay (2)	
		5.25"FH/3.5"HH Bay	"na"
		5.25"SL or 3.5"SL Bay	"na"
		3.5"HH Bay for Model 25	"na"
		3.5"HH Bay for 286 Machines	"na"
		Hardfile Kit	
		Drive Kit	
		Tape Backup Kit	
		CD Rom Kit	
D17: Empty: Null: Null			
AutoP			

Fig. 6. Resulting propagation after entering the customer's preferences.

that just discussed. After the sales representative makes a selection, the system performs further data retrieval as well as value propagation. Propagation is automatic and results in a fully configured standard offering as shown in Fig. 6.

If at this stage, a model developer were to scroll one screen to the right they would encounter the Macro and Constraint region of the constraint sheet shown in Fig. 7. By clicking on any macro or constraint cell with the mouse, a short description of that macro or constraint appears at the bottom part of the screen. By pressing on the space bar, an editor window will pop up, showing the contents of that macro or constraint. The developer can utilize this facility to examine or edit the macro or constraint. The “#MAC” on the screen represents a macro which can be used to write Prolog-like subprograms to perform database retrieval/updates and other operations. The “#T” on the screen represents a constraint which has been satisfied. The “#U” represents a constraint in which there is insufficient

information to determine whether it is satisfied or violated. Because there is no “#F” (violated constraint) symbol on the constraint sheet at this stage, the configuration is still valid. This information is usually presented to sales representatives in a more succinct format.

After the initial retrieval of the standard model, the sales representative can add components to the customer's configuration by choosing available options. If we continue with the example and focus on the Hardware Options section of Fig. 6, all of the Bay Options and some of the other options are filled with the value *na*. *na* means that an option is “not applicable” and is used to indicate those options that have been removed from further consideration because they are not compatible with options already selected. In our example, “Memory Kit 1” and “Memory Kit 2” options are available.

Let us suppose the sales representative adds 8 MB of 80ns memory for both “Memory Kit 1” and “Memory Kit 2”; this specifies a total of 20 MB of

PCCON.SWK													
File Edit Cell Utilities Connect Close Help													
	J												
		Memory (KB)				Kit 1	Kit 2	Kit 3	Kit 4	Kit 5	Kit 6		
	Std	4000											
	Extra	0											
	Total	4000											
Constraints in IBM PC Configuration													

#T	#T		#MAC	#U	#U			#MAC	#MAC				
								#U	#U				
Implications in IBM PC Configuration													

#MAC	#T	#T	#MAC		#U	#MAC	#U	#U	#MAC	#U	#U	#MAC	#U
#MAC	#T	#T	#MAC	#U	#U	#MAC	#T	#T	#MAC	#U	#U	#U	#U
#MAC	#T	#T	#MAC	#T		#MAC	#U	#U	#MAC	#U	#U	#U	#U
#MAC	#U	#U	#MAC	#T		#MAC		#U	#MAC	#T		#MAC	#T
#MAC	#U	#U	#MAC	#T	#T	#MAC	#U		#MAC	#T	#T	#T	#T
#MAC	#T		#MAC	#T	#T	#MAC	#U	#U	#MAC	#U	#U	#U	
#MAC	#T	#T				#MAC	#U		#MAC	#U	#U	#U	
#MAC	#T		#MAC	#T	#T	#MAC	#U		#MAC	#U	#U	#U	
Macros and Constraints to retrieve Data from Database													

#MAC	#MAC	#MAC	#MAC	#T	#MAC	#MAC	#MAC	#MAC	#MAC	#MAC	#MAC	#MAC	#MAC
#T	#T	#T	#T	#T	#T	#T	#U	#U	#T	#T	#T	#T	
Macros and Constraints to retrieve Product Price from Database													

#MAC	#T	#MAC	#T	#MAC	#MAC	#U	#MAC		#MAC		#MAC	#U	
#MAC	#T	#MAC	#T	#U	#MAC	#U	#MAC	#T	#MAC	#T	#MAC	#T	
#MAC	#T	#MAC	#T	#U	#MAC	#T	#MAC	#T	#MAC	#T	#MAC	#U	
#MAC	#U	#MAC	#T	#U	#MAC	#T	#MAC	#U	#MAC	#T	#MAC	#U	
Constraint for Target Price													

Constraint ==total price <= target price : 0 AutoP													

Fig. 7. PC/CON's constraints and macros.

memory for the machine since 4 MB of memory comes standard with the model. The memory kit selection is done in a manner similar to that of selecting a value for "Choice of Processor". Once the selection is made, the system retrieves the price for the components from the database and recalculates the total system price using a constraint sheet formula. However, the maximum allowable memory for this submodel is 16 MB. Consequently, two constraints located in cells J8 and K8 are now violated and their respective cell boxes would change from "#T" to "#F".

Fig. 8 shows the *violation window* that identifies the violated constraints. Violated constraints are shown in the *Constraints* child-window. The values input by the sales representative which caused the violation appear in the *Assumptions* child-window. The contents of the violated constraint and a description of how to resolve the violation appear in the *Constraint* child-window. The *violation window* helps the sales representative identify what caused the violation and to resolve it. As shown in the

Constraint child-window, the violations were caused by the total memory being greater than the maximum possible memory for the submodel. If the constraint in cell J8 were also examined, we would find that the total system price also exceeds the target price.

By deleting the value for "Memory Kit 2", the system will re-evaluate the configuration and conclude that the configuration is valid. Continuing in this way, more options can be added. As long as there are no constraint violations, the sales representative is assured that the configuration is valid and can be manufactured.

8. Supporting PC configuration with PC/CON and Saturn

The example session with PC/CON demonstrated above is one approach to PC product configuration. This is only one of many scenarios available for configuring personal computers using PC/CON. PC/CON's architecture and Saturn's supporting fea-

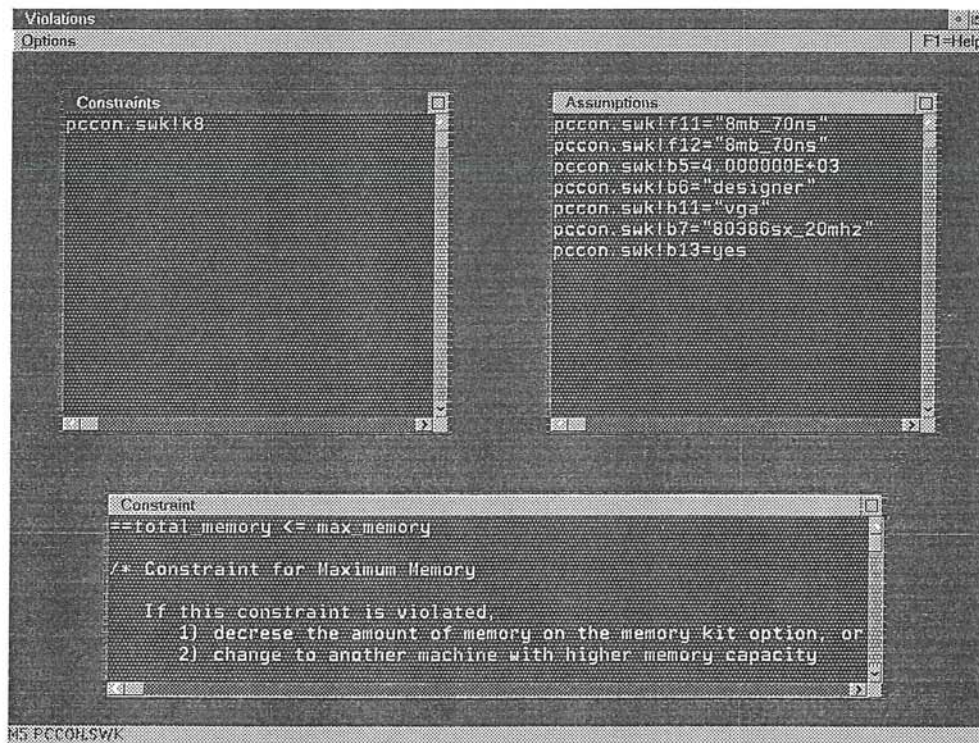


Fig. 8. Window displaying assistance to help resolve constraint violations.

tures permit a user to start from anywhere in the application. For example, a user can skip the Customer Preferences section and start by selecting a particular model or submodel. From there, the user can select different hardware, software, display, and other miscellaneous options to customize their configuration. This not only caters to the many diverse users but also accommodates those with varying expertise in terms of the products and the PC/CON application.

During the configuration process, every selection by the customer imposes various constraints in PC/CON to further reduce the solution space. Fig. 9 shows the reduction of feasible configuration combinations while progressing through the selection steps to form a complete product configuration (e.g., a configuration which includes a base model/submodel and consideration of all options). The numbers in Fig. 9 are combinatorial calculations using product data from the IBM Personal System/2 Sales advisor [7]. At the beginning of all three approaches, there are 36,744,208,624 possible PC configurations (see Table 1). Fig. 9 shows three approaches, all of which conclude with the same configuration for a 56-043 submodel.

As the user progresses through option selection, each option choice precludes the selection of options that conflict with it. The selection of an option constrains the possibilities that are available for the next choice. The application takes advantage of Saturn's unique features: domain restriction and set propagation. Domain restriction uses selected options as conditions in database queries to retrieve a restricted set of values. Domain restriction allows PC/CON to display only those values from which a user makes a selection that will not violate any previously satisfied constraint. Domain restriction prevents a user from selecting an option that conflicts with a previously selected option. Set propagation maintains consistency of the future variables with the current and past variable assignments. After the current variable receives a value, the remaining unbound variables or future variables which are related in the database are tested to see if there are any remaining values in their domains that are consistent with the past and current variable assignments as well as with each other. If the test reveals that the domain for a future variable has become empty, then this variable is no longer compatible with the solving design and is "not applicable". It is then assigned

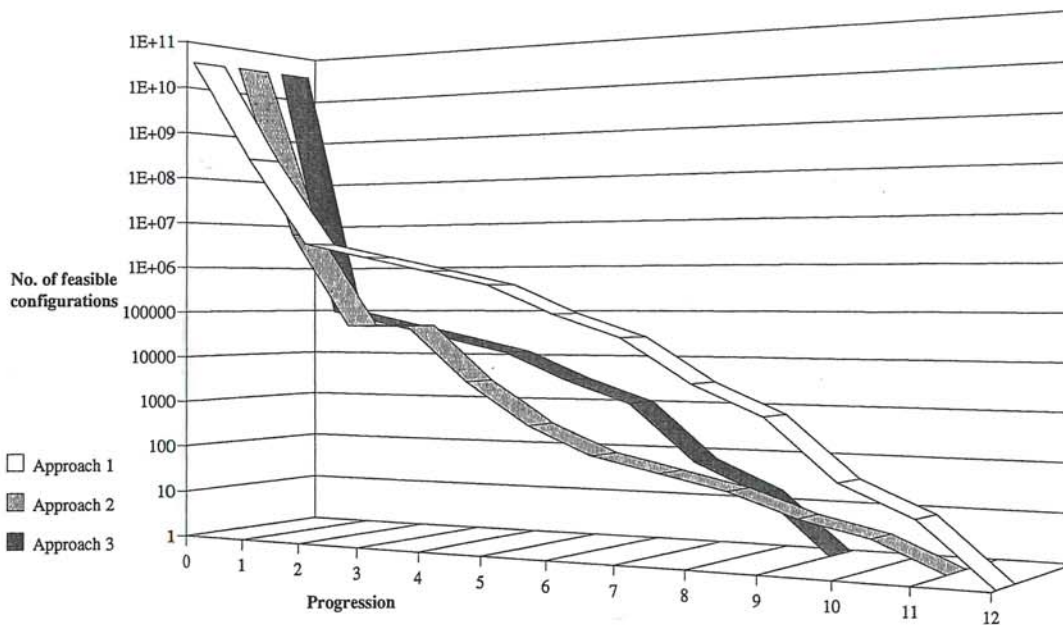


Fig. 9. Approaches to configuring the 56-043 submodel.

the value *na* and is removed from further considerations.

With domain restriction and set propagation, as a user makes configuration choices, the number of feasible configurations available from which to make additional choices is reduced. This means that the user is prevented from making a product configuration choice that is in conflict with previous or future choices. In this example, as shown in Fig. 9, the reduction is quite dramatic. However, if the user wants to explore other possibilities, domain restriction and set propagation can be turned off. In this case the user depends upon the violation detection and correction support of the ATMS to find a feasible solution. Domain restriction and set propagation are discussed in more detail in [19].

In Fig. 9, Approach 1 represents a sales representative who goes through a normal product configuration process to select a model, submodel, processor upgrade, math coprocessor, etc. The number of possible product configuration combinations is reduced significantly according to the user selection made in each selection step. Approach 2 is similar to the example in the previous section where a sales representative follows the navigational assistance provided through the customer preferences section on the constraint sheet to select maximum price, software to use, processor, memory, etc. Approach 3 shows how an experienced sales representative can further reduce the number of steps involved in the product configuration process by selecting the submodel first, followed by other selections. Notice that the more drastic reduction in the number of possible combinations corresponds to a decrease in the number of steps taken for a complete product configuration. This is consistent with Approaches 2 and 3. Since each solution approach is a different starting point for a user, the example demonstrates that the Saturn-based PC/CON system allows a user to begin configuring a computer by selecting any option, model or submodel, and can proceed without following a prescribed sequence. Thus, a user can proceed directly to the attributes addressing a customer's questions while avoiding the sequential questioning session typical of expert systems.

Saturn provides an immense amount of support during the development of constraint-based models. The constraint sheet is highly interactive and will be

familiar to a vast spectrum of users that have already been exposed to spreadsheets. This enables non-specialists to build models of computer product configurations that can then be exercised in a build-test cycle. This method of building models has significant advantages over systems that require a specialized program to be written, compiled and only then tested. In addition, connecting specific user interfaces to Saturn is a simple matter of using Dynamic Data Exchange and linking cells in the constraint sheet to entry fields in the user interface. Experiences with Visual Basic has shown that the development of a user interface and a computer product configuration application can be simultaneously performed. This can separate solution navigation development from configuration system development and simplify overall development of the product configuration system.

8.1. Custom user interface development

A prototype custom user interface was developed for direct response marketing users that would configure and sell PCs directly to customers ordering direct over the telephone. The prototype was developed using Visual Basic version 2.0 [31] running as an MS Windows program under OS/2, version 2.1. Development of the interface involved learning Visual Basic, learning IBM's common user access guidelines [32], and designing and implementing the prototype in Visual Basic. The custom interface interacts with the PC/CON constraint sheet as it executes within the Saturn Constraint-Modeling Environment. Communication between the interface and the constraint sheet is facilitated through the Dynamic Data Exchange protocol. The custom interface consists of 25 full screen forms that implement all the features described in the previous sections but in a more accessible format. The excellent development features in Visual Basic allowed the custom interface to be built by a single engineer in one month's time. Access to the other forms is obtained by selecting the appropriate tab on the right edge of the book-like form.

8.2. System maintenance

Our claim of development and maintenance ease is supported by our experiences in building PC/CON

with data from the 1991/1992 edition of the IBM Sales Advisor catalogue [6] and then updating this information to reflect the changes that appeared in the 1993 edition of the IBM Sales Advisor catalogue [7]. To construct a PC configuration system, we transform the product information that resides in the IBM Sales Advisor catalogue into constraints and a relational database in PC/CON. Since 99% of all product information is stored in PC/CON's relational database, most system maintenance is performed using a standard query manager. This precludes the necessity of implementing changes by altering the applications architecture. It also allows the product information to be stored centrally and either accessed remotely or shadowed to remote locations. These features enabled the complete PC/CON application to be rapidly constructed in three weeks by two engineers who were initially unfamiliar with the PC product configuration problem.

Updating PC/CON with the latest release of product/price information, the 1993 edition of the IBM Sales Advisor, demonstrates its ease of maintenance. The update included the deletion and addition of models and submodels for all PC models and significant changes in component options for all submodels. It was a major overhaul of the product information initially loaded from the 1991/92 edition [7]. To demonstrate maintenance ease, it was important that the maintenance be performed by a typical manufacturing engineer. At a company such as IBM, a typical manufacturing engineer will have been an above average student that is computer literate but is not a programmer/analyst, nor has any database experience. In addition, they would have little detailed knowledge of the product structure associated with the computers that are manufactured.

To test this hypothesis, a senior Industrial Engineering student fitting the above description was employed to update the configuration application with the 1993 product/price information. The student had never used PC/CON nor Saturn; nor had the student ever used a relational database nor OS/2. To prepare for the task, the student worked through a tutorial on relational databases which required approximately two days to complete. At this point the student had sufficient familiarity with OS/2's query manager command line interface to begin the update task. The update task itself required two days (16

hours) with an additional evening spent by the PC/CON implementers to correct some constraints in the application. The total update time was three days (24 hours total).

9. Conclusions

Saturn and PC/CON have proven very successful in supporting the key characteristics that exist in the expansive nature of computer product configuration in the computer industry. Saturn supports product configuration that is performed at the point of sale, thereby eliminating costly manufacturing errors that currently appear in a large number of customer orders. Its domain restriction, set propagation, and constraint satisfaction characteristics prevent an infeasible configuration from being created. Consequently, Saturn solves the computer product configuration problem and addresses all the issues raised in Section 1. In addition to performing computer product configuration, Saturn's facilities support both rapid constraint-based model development and computer product configuration from alternative perspectives. These facilities derive from integrating a relational database and a constraint sheet into a formal logic theory that implements non-monotonic, interactive constraint satisfaction. Unlike the expert system-based computer configuration systems, PC/CON's architecture and Saturn's supporting features permit a user to start from any desired perspective to configure a computer system. It provides the flexibility to accommodate both novice and experience users. The structure of the PC/CON constraint-based model was also designed to support rapid implementation of custom interfaces as well as rapid maintenance and new product inclusion tasks so that they can be performed by non-specialized personnel. Computer product configuration is now attainable through Saturn's interactive constraint-satisfaction capabilities, which enable PC/CON to be exercised from a number of perspectives under a number of scenarios. The result of this implementation is an easily maintainable configuration support system that ensures valid product configurations in terms of hardware, software, and price that can be manufactured to meet the customers requirements.

Acknowledgements

This work was partially supported by the IBM PC Company. The authors would like to acknowledge Dr. Mike Spano, Dave Hattori and Don Willenborg of the IBM PC Company for their support and extensive discussions on the configuration problem. The authors would also like to acknowledge Ramdus Murali for his work in implementing the Assumption-Based Truth Maintenance used in Saturn, and also Steve Minaar and Kare Christensen for extensively Beta testing Saturn.

References

- [1] Trilogy, *SalesBUILDER Handbook*, 1992.
- [2] V.E. Barker and D.E. O'Connor, "Expert systems for configuration at Digital: xCON and beyond", *Commun. ACM* 32(3) (1989) 298-310.
- [3] J. McDermott, "R1: A rule-based configurator of computer systems", *Artif. Intell.* 19 (1982) 32-88.
- [4] H. Lowe, "Extending the Proof Plan methodology to computer configuration problems", *Appl. Artif. Intell.* 5 (1991) 227-252.
- [5] S. Fohn, R. Young, A. Greef, R. Murali, and P. O'Grady, "Saturn: A computer environment for modeling large concurrent engineering systems" Technical Report, Group for Intelligent Systems in Design and Manufacturing, Department of Industrial Engineering, North Carolina State University, Raleigh, NC, 1992.
- [6] *IBM Personal System/2 and Printers Sales Advisor*, 1991/92 Winter edition, IBM Corporation, Purchase, NY, 1992.
- [7] *IBM Personal System/2 and Printers Sales Advisor*, 1992/93 Fall edition, IBM Corporation, Purchase, NY, 1992.
- [8] J. Bachant and E. Soloway, "The engineering of xCON", *Commun. ACM* 32(3) (1989) 311-317.
- [9] S.M. Mittal and C.L. Dym, "PRIDE: An expert system for the design of paper-handling systems", *Computer* 19(7) (July 1986) 102-114.
- [10] F. Frayman and S. Mittal, "COSSACK: A constraint-based expert system for configuration tasks", *Knowledge-Based Expert Systems in Engineering: Planning and Design*, September 1987, pp. 143-166.
- [11] R. Young and A. Greef, "Representing rule-based expert systems as constraint networks", Technical Report, Group for Intelligent Systems in Design and Manufacturing, Department of Industrial Engineering, North Carolina State University, Raleigh, SC, 1991.
- [12] W.M. Bulkeley, "New software is helping reps fill custom orders without glitches", *Wall Street J.* (August 11, 1992).
- [13] S. Fohn, A. Greef, R. Young and P. O'Grady, "A constraint system shell to support concurrent engineering approaches to design", *Artif. Intell. Eng.* (1994) to appear.
- [14] A. Oberschelp, "Order sorted predicate logic", in: K.H. Bläsius and U. Hedtstück (eds.), *Sorts and Types in Artificial Intelligence*, Lecture Notes in Artificial Intelligence, Springer-Verlag, Berlin, 1989, pp. 8-17.
- [15] A. Greef, S. Fohn, R. Young and P. O'Grady, "Implementation of a logic-based support system for concurrent engineering", *J. Data Knowledge Eng.* (1994) to appear.
- [16] R. Reiter, "Towards a logical reconstruction of relational database theory", in: M. Brodie, J. Mylopoulos and J. Schmidt (eds.), *On Conceptual Modeling*, Springer-Verlag, New York, NY, 1984.
- [17] J. deKleer, "An assumption-based TMS", *Artif. Intell.* 28 (1986) 127-162.
- [18] R. Murali, "A nonmonotonic support system for an interactive constraint satisfaction tool used in concurrent engineering", Masters Thesis, Department of Industrial Engineering, North Carolina State University, Raleigh, NC, 1992.
- [19] S. Fohn, "Dynamic constraint satisfaction with the relational model for concurrent engineering", Ph.D. Thesis, Department of Industrial Engineering, North Carolina State University, Raleigh, NC, 1994.
- [20] R.E. Young, A. Greef and P. O'Grady, "An artificial intelligence constraint network system for concurrent engineering", *Int. J. Prod. Res.* 30(7) (1992) 1715-1735.
- [21] J.-S. Oh, P. O'Grady and R.E. Young, "An artificial intelligence constraint network approach to design for assembly", *IIE Trans.* (1994) to appear.
- [22] P. O'Grady, J.-Y. Kim and R.E. Young, "A concurrent engineering system for rotational parts", *Int. J. Syst. Autom.* 2(3) (1992) 245-258.
- [23] J.-Y. Kim, R. Mittal, P. O'Grady and R.E. Young, "Process selection for concurrent engineering in the domain of rotational parts", *J. Des. Manuf.* 2 (1992) 199-209.
- [24] C. Kim, P. O'Grady and R.E. Young, "TEST: A design for testability system for printed wiring boards", *J. Electron. Manuf.* 2 (1992) 61-70.
- [25] J.S. Liau and R.E. Young, "PACE: A process planning and concurrent engineering system for PCBs", *Manuf. Rev.* 6(1) (1993) 25-39.
- [26] S. Fohn, A. Greef, R. Young and P. O'Grady, "A constraint based approach to AS/RS design", *Proc. IIE Research Conf.*, 1994.
- [27] Y. Kim and P. O'Grady, "A methodology for analyzing large scale concurrent engineering systems", *Int. J. Prod. Res.* (1994) to appear.
- [28] K. Kim, D. Cormier, P. O'Grady and R. Young, "A system for design and concurrent engineering under imprecision", *J. Intell. Manuf.* (1994) to appear.
- [29] D. Cormier and P. O'Grady, "Using genetic algorithms and constraint modeling in concurrent engineering", *Proc. IIE Research Conf.*, 1994.
- [30] P. O'Grady, K. Kim and R.E. Young, "A hierarchical approach to concurrent engineering systems", *Int. J. Comput. Integr. Manuf.* 7(3) (1994) 152-162.
- [31] *Microsoft Visual Basic Programming System for Windows*,

Version 2.0, Standard Edition, Microsoft Corporation, Redmond, WA, 1992.

- [32] *Object-Oriented Interface Design - IBM Common User Access (CUA) Guidelines*, IBM OS/2 Technical Library, 1st edn., QUE Publishers, IBM Corporation, Purchase, NY, 1992.



Steffen M. Fohn is a Knowledge Engineer in the Network Services Division at IBM. He is currently developing the knowledge base and modeling language for a functional reasoning system to be used in the configuration of complex products. His research interests include the application and theory of constraint satisfaction systems, deductive and semantic databases, and logic programming to solve information and manufacturing problems. He received his Ph.D.

degree in Industrial Engineering and his Masters degree in Integrated Manufacturing System Engineering from North Carolina State University and his BS degree in Mechanical Engineering from the University of North Carolina at Charlotte.



Jong-Shin Liao is a post doctoral research associate in the Department of Industrial Engineering at North Carolina State University. He received his M.S. and Ph.D. degrees in Industrial Engineering from North Carolina State University, and his bachelor's degree in Mechanical Engineering from Tamkang University in Taiwan. He is currently involved in the research and development of interactive constraint modeling systems to solve various design problems.

His present research interests include design and implementation of concurrent engineering systems, computer integrated manufacturing systems, and the application of artificial intelligence to manufacturing systems.



Arthur R. Greef is a Software Engineer at IBM Corporation where he is developing a logic-based, functional reasoning system for configuring complex products. Previously, he worked as a Post Doctoral student at North Carolina State University. His interests are in applying artificial intelligence to engineering and manufacturing problems, particularly applications requiring constraint satisfaction and logic programming. He received his Ph.D. in Industrial Engineering

at the University of Stellenbosch in South Africa, and his M.Sc. and B.Sc. in Mechanical Engineering at the University of Natal, also in South Africa.



Robert E. Young is an Associate Professor in the Department of Industrial Engineering at North Carolina State University. His principal interests are concurrent engineering and CIM system design. He has worked extensively with automotive, electronic and aerospace industries as a consultant and with them fulltime on leaves of absence from academia. Based upon his experience, he has been called to give expert testimony before Congress on manufacturing

and automation. For the National Science Foundation he has participated on many panels and reviewed over 200 proposals. He has been working in the area of computer technology for 30 years and in manufacturing systems since 1975. He has had more than \$5,000,000 in research funding during his career and published over 60 articles on various aspects of manufacturing, many book chapters, and two books.

In 1982 he was named as *Young Manufacturing Engineer of the Year* by the Society of manufacturing Engineers. In 1978 he was a *University Fellow* with the U.S. Air Force's Integrated Computer-Aided Manufacturing (ICAM) Program, and subsequently worked on many of its projects. He is a past Director of the Computer and Information Systems Division of the Institute of Industrial Engineers. He received his Ph.D. and Masters Degrees in Industrial Engineering from Purdue University, and his Bachelor Science in Engineering from UCLA. He is a Registered Engineer in the State of Texas.



Peter O'Grady is a Full Professor of Industrial Engineering at North Carolina State University. He received his bachelors degree from University of Cambridge and his Ph.D. from University of Nottingham. His main areas of expertise are in manufacturing engineering and include manufacturing analysis, manufacturing systems, computer integrated manufacturing, manufacturing control, and concurrent engineering. Dr. O'Grady is the author of over one hundred papers, and several books. He has substantial management and administrative experience.

He is Editor, Design, *Transactions of Institute of Industrial Engineers*; USA Editor of *Journal of Design and Manufacturing*; Associate Editor of *Journal of Manufacturing Systems*; a member of the Editorial Board of *International Journal of Computer Integrated Manufacturing*; a member of the Editorial Board of *International Journal of Production Planning and Control*; a member of the Editorial Board of *Journal of Computers and Industrial Engineering*; and a member of the Editorial Board of *Journal of Engineering Design and Automation*.



ELSEVIER

Computers in Industry 27 (1995) 1

**COMPUTERS IN
INDUSTRY**

Editorial

The application of information technology in single operations is becoming normal practice. This fact is not only true for manufacturing and transportation operations, but also for operations of professional knowledge workers such as engineers. Therefore, the mere statement that a particular type of work is supported by computers raises hardly any interest today.

The observation that IT becomes part of almost everything has consequences for the editorial policy of *Computers in Industry*. If we want to publish papers about computer applications in single operations, then these applications have to be new, unexpected, and challenging. Moreover, they have to be new *with respect to the usage of IT*. We believe that the papers presented in this issue are good examples. These papers are concerned with classical problems (configuring customer products, object measurement, computer-aided process planning, relay ladder logic) but they have in common that the IT solutions which are presented are new.

However, the contribution of computers in industry is less and less based on IT applications in individual operations – rather, the strength of information technology is based on integration

and communication. Therefore, architectures and infrastructures are themes which play an important role in the publishing policy of *Computers in Industry*. In fact, the paper written by Raja et al. on fieldbus applications in the issue before you is a good example.

We have published a special issue on CIM architectures (Vol. 24, Nr. 2/3), and we will proceed with this theme. The next issue (Vol. 27, Nr. 2) is entirely devoted to the application of CIMOSA in practice. We would be delighted to get more papers on application of standards, architectures or infrastructures in practice. These papers do not necessarily have to be concerned with “classical” CIM issues such as shop floor control and CAD/CAM. For example, we will publish soon another special issue on architectures for cooperative environments, including multimedia applications. There is so much challenge in industrial application of the new technologies based on powerful PCs and today’s network technology that it is a pleasure to be editor-in-chief of *Computers in Industry*!

Professor Hans Wortmann
Editor-in-Chief

COMPUTERS IN INDUSTRY

Editor-in-Chief

Prof. Hans Wortmann
Eindhoven University of Technology
The Netherlands

Editorial Office

Eindhoven University of Technology
Graduate School of Industrial Engineering and
Management Science
P.O. Box 513, Paviljoen U42a
5600 MB Eindhoven, The Netherlands
Fax: +31-40-451275

For regular papers: **Vincent Wiers**
Tel.: +31-40-472291/472493
e-mail: v.c.s.wiers@bdk.tue.nl

For Special Issues: **Monique Jansen**
Tel.: +31-40-472952/472290
e-mail: m.h.jansen@bdk.tue.nl

Editorial Board

S. Bologna, Maria di Galeria, Italy
J.P. Crestin, Guer, France
M.J. Cronin, Billerica, MA, USA
J.D. Goldhar, Chicago, IL, USA
G. Guariso, Milan, Italy
T.J. Harrison, Tallahassee, FL, USA
D. Kochan, Dresden, Germany
M. Kubicek, Prague, Czech Republic
C. Kuo, Glasgow, UK
M.E. Merchant, Cincinnati, OH, USA
T. Miura, Tokyo, Japan

Co-editor North America

Prof. Colin L. Moodie
School of Industrial Engineering
Purdue University
Grissom Hall
West Lafayette, IN 47907, USA
Tel.: (317) 494-5442
Fax: (317) 494-5448
e-mail: moodie@ecn.purdue.edu

Co-editor Industrial Acquisitions

Ms. Gloria Karlmark
Philips Lighting B.V.
Building EH-2
P.O. Box 80020
5600 JM Eindhoven
The Netherlands

L. Nemes, Preston, Vic., Australia
D.H. Rawal, Bombay, India
A. Rolstadås, Trondheim, Norway
C.W. Rosenthal, Portland, OR, USA
O.I. Semenov, Minsk, Belarus
B.G. Tamm, Tallinn, Estonia
M. Tomljanovich, Milan, Italy
M. Villabo, Trondheim, Norway
J. Vlietstra, Willits, CA, USA
E.A. Warman, Peterborough, UK
T.J. Williams, West-Lafayette, IN, USA

© 1995, Elsevier Science B.V. All rights reserved.

No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Elsevier Science B.V., P.O. Box 1991, 1000 BZ Amsterdam, The Netherlands.

Upon acceptance of an article by the journal, the author(s) will be asked to transfer copyright of the article to the publisher. This transfer will ensure the widest possible dissemination of information.

Special regulations for readers in the U.S.A. – This journal has been registered with the Copyright Clearance Center, Inc. Consent is given for copying articles for personal or internal use, or for the personal use of specific clients. This consent is given on the condition that the copier pays through the Center the per-copy fee stated in the code on the first page of each article for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. The appropriate fee should be forwarded with a copy of the first page of the article to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, U.S.A. If no code appears in an article, the author has not given broad consent to copy and permission to copy must be obtained directly from the author. The fee indicated on the first page of an article in this issue will apply retroactively to all articles published in the journal, regardless of the year of publication. This consent does not extend to other kinds of copying, such as for general distribution, resale, advertising and promotion purposes, or for creating new collective works. Special written permission must be obtained from the publisher for such copying.

No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation of and methods, products, instructions or ideas contained in the material herein.

Although all advertising material is expected to conform to ethical (medical) standards, inclusion in this publication does not constitute a guarantee or endorsement of the quality or value of such product or of the claims made of it by its manufacturer.

Editorial Board

- S. Bologna**, ENEA CRE-CASACCIA, Via Anguillarese 301, I-00060 S. Maria di Galeria (RM), Italy
J.P. Crestin, Directeur Général de l'Enseignement et de la Recherche, Ecoles de Coëtquidan, F-56381 Guer, France
M.J. Cronin, Cognition, 755 Middlesex Turnpike, Billerica, MA 01821, USA
J.D. Goldhar, Illinois Institute of Technology, IIT Center, Chicago, IL 60616, USA
G. Guariso, Department of Electronics and Information, Politecnico di Milano, Via Ponzio 34/5, I-20133 Milan, Italy
T.J. Harrison, Florida A&M University/The Florida State University, College of Engineering, Department of Electrical Engineering, P.O. Box 2175, Tallahassee, FL 32316-2175, USA
D. Kochan, GFAI Sachsen, Bergstraße 69, 01069 Dresden, Germany
M. Kubicek, Prague Institute of Chemical Technology, Department of Mathematics, Technicka 5, 16628 Prague 6, Czech Republic
C. Kuo, University of Strathclyde, Department of Ship and Marine Technology, 100 Montrose Street, Glasgow G4 6LZ, UK
M.E. Merchant, Institute of Advanced Manufacturing Sciences Inc., 1111 Edison Drive, Cincinnati, OH 45216-2265, USA
T. Miura, Hitachi Ltd., Hitachi Omori Building #2, 27-18 Minami-Oi 6-Chome, Shinagawa-ku, Tokyo 140, Japan
L. Nemes, CSIRO, Division of Manufacturing Technology, Locked Bag No. 9, Preston, Vic. 3072, Australia
D.H. Rawal, Systech Consultancy Services, c/o Chhaganlal Vishram & Co., 107-109 Sherif Devji Street, Bombay 400 003, India
A. Rolstadås, University of Trondheim, Production Engineering Laboratory, NTH-SINTEF, Richard Birke-landsvei 2B, 7034 Trondheim, Norway
C.W. Rosenthal, Engineering Consultants, 3080 S.W. Fairmont Boulevard, Portland, OR 97201-1439, USA
O.I. Semenov, RNUIC Research Centre, Academy of Management, 17, Moskovskaya Str., Minsk 220001, Belarus
B.G. Tamm, Institute of Cybernetics, Akadeemia Tee 21, EE0026 Tallinn, Estonia
M. Tomljanovich, Italcad, Business Development, Via Inverigo 6, I-20151 Milan, Italy
M. Villabo, MARINTEK, Otto Nielsens veg 10, P.O. Box 4125 Valentinlyst, N-7002 Trondheim, Norway
J. Vlietstra, 1942 Buckeye Road, Willits, CA 95490, USA
E.A. Warman, K. Four Ltd., Nene House, Town Bridge, London Road, Peterborough PE2 8AH, UK
T.J. Williams, Purdue University, Purdue Laboratory for Applied Industrial Control, 1293 Potter Engineering Center, West-Lafayette, IN 47907-1293, USA

Abstracted/Indexed in:

ACM Computing Reviews
Computer Literature Index
Current Contents
Data Processing Digest
Engineering Information
Inspec
Management Contents
Science Citation Index

© 1995, Elsevier Science B.V. All rights reserved.

No part of this publication may be reproduced stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photocopying, recording or otherwise, without the prior permission of the publisher, Elsevier Science B.V., P.O. Box 1991, 1000 BZ Amsterdam, The Netherlands.

Upon acceptance of an article by the journal the author(s) will be asked to transfer copyright of the article to the publisher. This transfer will ensure the widest possible dissemination of information.

Special regulations for readers in the U.S.A. – This journal has been registered with the Copyright Clearance Center, Inc. Consent is given for copying articles for personal or internal use, or for the personal use of specific clients. This consent is given on the condition that the copier pays through the Center the per-copy fee stated in the code on the first page of each article for copying beyond that permitted by Sections 107 or 108 of the U.S. Copyright Law. The appropriate fee should be forwarded with a copy of the first page of the article to the Copyright Clearance Center, Inc., 222 Rosewood Drive, Danvers, MA 01923, U.S.A. If no code appears in an article, the author has not given broad consent to copy and permission to copy must be obtained directly from the author. The fee indicated on the first page of an article in this issue will apply retroactively to all articles published in the journal, regardless of the year of publication. This consent does not extend to other kinds of copying, such as for general distribution, resale, advertising and promotion purposes, or for creating new collective works. Special written permission must be obtained from the publisher for such copying. No responsibility is assumed by the Publisher for any injury and/or damage to persons or property as a matter of products liability, negligence or otherwise, or from any use or operation or any methods, products, instructions or ideas contained in the material herein. Although all advertising material is expected to conform to ethical (medical) standards, inclusion in this publication does not constitute a guarantee or endorsement of the quality or value of such product or of the claims made of it by its manufacturer.

∞ The paper used in this publication meets the requirements of ANSI/NISO Z39.48-1992 (Permanence of Paper).

Published 9 times per year

0166-3615/95/\$09.50

Printed in The Netherlands

Aims and scope

The aim of *Computers in Industry* is to publish original, high-quality, application-oriented research papers that:

- Show new trends in and options for the use of Information Technology in industry;
- Link or integrate different technology fields in the broad area of computer applications for industry;
- Further the state-of-the-art within each area of information technology for industry.

A continuous quality policy, based on strict peer reviewing, shall ensure articles are:

- Technologically outstanding and front-end;
- Application-oriented with a generalized message;
- Representative for research at an international level.

General topics covered by the journal include, while not exclusively, the following areas:

- The unique application of information technology in business processes such as design, engineering, manufacturing, purchasing, physical distribution, and production management. This is the main thrust of the journal.
- The industrial use of information technology in knowledge intensive fields such as quality control, logistics, engineering data management, product documentation will certainly be considered.
- Demonstration of the enabling capabilities of new or existing technologies such as hard real time systems, knowledge engineering, applied fuzzy logic, collaborative work systems are also welcomed.
- Papers solely focusing on IT or manufacturing processes may be considered out of scope.

Types of Contributions

Computers in Industry features

- *Survey papers.* Overviews of the State-of-the-Art of a chosen topic within the scope of the journal. These papers should be around 10,000 words (20 to 25 typescript pages).
- *Regular papers.* These papers should report original findings on IT applications. These applications should be new and raise the interest of engineers and other professionals and should contain a generalised message. Papers should not exceed 5,000 words in length. Papers describing individual case studies can be included here, provided the relevance of the study is clearly mentioned in a broader context.
- *Short Notes.* Comments or position papers of no more than 2,000 words in length. These notes may be a commentary on a previously published paper or an invitation for discussion on a particular point.

How to submit

Submission of a paper implies that it has not been previously published elsewhere and is not under consideration for publication elsewhere and that the author has obtained all necessary permissions to publish the text and illustrations in the papers.

Four copies, or one copy and a PostScript-version on 3.5" diskette, of each manuscript (typewritten, double spaced, in English) should be sent to our Editorial Office:

Prof. Hans Wortmann, c/o Vincent Wiers, Eindhoven University of Technology, P.O. Box 513, Paviljoen U42a, 5600 MB Eindhoven, The Netherlands. Tel +31 40 472291/472493, Fax +31 40 451275, e-mail v.c.s.wiers@bdk.tue.nl.

The paper must be accompanied by the full address of the corresponding author, telephone, fax and e-mail numbers.

Preparation of the Manuscript

The title page should state the title, names of authors and their academic or professional affiliations. The second page should state the title page and a brief biography of the authors. The third page should include the title, an abstract of 150 words or less and up to 5 keywords. The main text begins on the following page. Tables and original illustrations should be submitted on separate pages with a clear indication to which part of the text they belong. References should also be submitted on a separate page, should be numbered and listed at the end of the paper and referred to in the text by the number e.g. Journals: P.R. Sherrer, "Determining distance to a surface represented in piecewise fashion with surface paths", *Computers in Industry*, Vol. 6, 1985, pp. 173-183. Books: S.A. Coons, *Surface Patches and B-splines*, Academic Press, New York, 1974.

Reviewing Procedure

All contributions are submitted to referees. Reviews are double blind, authors should therefore avoid revealing their identity, not citing their own publications with undue frequency. Manuscripts which do not comply will not be reviewed

Accepted Manuscripts

All accepted manuscripts should be accompanied by

- A copy of the final accepted text on diskette. This may be on either 3.5" or 5.25", either high or double density and either MS-DOS/PC-DOS or Macintosh OS. This disk will be used by the publishers for rapid typesetting of the manuscript and therefore it is essential that, apart from minor corrections and accompanying figures, the disk version exactly matches the accepted hardcopy. Full instructions for Electronic text preparation are given in the back pages of this issue.
- All original artwork.
- Keywords.

Proofs and reprints

Proofs of survey papers and regular papers will be sent to authors to check for typographical accuracy. No part of an article may be rewritten at this stage. No proofs will be provided for Short Notes. The corresponding author will receive 50 reprints free of charge. Additional reprints may be ordered at the publisher's regular prices on a form which accompanies the proofs, or in the case of Short Notes is sent directly to the corresponding author. There are no page charges.

Subscription Information

Computers in Industry (ISSN 0166-3615). For 1995 Volumes 27-29 are scheduled for publication.

Subscription prices are available on request from the publisher. Individual subscription prices are available under certain conditions. Please apply to the Publisher. Subscriptions are on a pre-paid basis and are entered on a calendar year basis. Issues are sent by surface mail except to the following countries where delivery by SAL mail is ensured: Argentina, Australia, Brazil, Canada, China, Hong Kong, India, Israel, Japan, Malaysia, Mexico, New Zealand, Pakistan, Singapore, South Africa, South Korea, Taiwan, Thailand, USA. For all other countries airmail rates are available on request. Claims for missing issues must be made within 6 months of our publication (mailing) date. Please address all queries for subscriptions to: Elsevier Science B.V., Journals Department, P.O. Box 211, 1000 AE Amsterdam, The Netherlands. Tel +31 20 4853642, Fax +31 20 4853598.

CONTENTS

1 Editorial

Regular Papers

3 Configuring computer systems through constraint-based modeling and interactive constraint satisfaction

S.M. Fohn, J.S. Liao, A.R. Greef, R.E. Young and P.J. O'Grady

23 OMNE-Vision —Object measurement in a noisy environment using vision

S. Vaidyanathan and S. Raman

33 Planning for automation for shipyards. An illustrative study

C. Kao, D.-C. Li, C. Wu, J.J. Lyu and H.-J. Shaw

43 Simulating fieldbus applications with DRUGH simulator

P. Raja, J. Hernandez, L. Ruiz, F. Guidec and J.D. Decotignie

53 A neural network approach for datum selection in computer-aided process planning

J. Mei, H.-C. Zhang and W.J.B. Oldham

65 An event chaining relay ladder logic solver

J.T. Welch

Short Note

75 State-of-the-art automatic test system for multi-access rural radio system

A.K. Goyal, U.J. Mathews, S.N. Baranwal, A.M. Verma and P. Ray

Events

83 Calls for Papers

87 Calendar





3 9343 01677 7889

SCIENCE & ENGINEERING LIBRARY
WAYNE STATE UNIVERSITY LIBRARIES
PERIODICALS

COMPUTERS IN INDUSTRY

26-27

1995