EXPERT SYSTEMS FOR CONFIGURATION AT DIGITAL: XCON AND BEYOND

Members of Digital Equipment Corporation's team of expert system experts reflect and recount a decade's worth of lessons learned in designing and building a core of configuration systems

VIRGINIA E. BARKER and DENNIS E. O'CONNOR

The XCON configuration system at Digital Equipment Corporation was the first expert system in daily production use in industry [2, 10, 11]. It is the cornerstone of Digital's knowledge network vision: a number of expert systems embedded in both the company's order process cycle and its new product introduction cycle [4]. Today, Digital is continuing to extend the knowledge network, as well as using expert systems technology in many additional aspects of the company's business. In fact, XCON is only one of several expert systems dealing with hardware and software configuration which are currently in use or under development at Digital. The configuration systems "family" includes four expert systems in production use. Several additional configuration expert systems are in the research, advanced development, or prototype stage of development.

The development of these expert systems has spanned nearly 10 years; Digital's Configuration Systems Development Group (CSDG) has thereby gained considerable understanding of all phases of the life-cycle of production quality expert systems: design, development, production, and ongoing support. In this article we highlight some of the key lessons we have learned:

- Building a successful expert system involves much more than simply putting rules into a knowledge base. Rather, to successfully develop and provide ongoing support for expert systems and to integrate them into the fabric of one's business, as has been done at Digital, one must attend to the needs of the business and to human resource and organizational issues as well as to technical issues.
- Although many software engineering practices carry over to the engineering of expert systems, we have nonetheless needed to be creative in developing practices specifically tuned to this relatively new, and still evolving, technology.

While the accompanying article by Bachant and Soloway focuses specifically on the technical issues underlying XCON and the other configuration systems, we take a more holistic, integrative approach in this article and attempt to put the aforementioned concerns into a coherent perspective.

CONFIGURATION SYSTEMS AT DIGITAL TODAY

The Digital Configuration Systems Timeline (p. 302) shows, for each of the configuration systems, key stage-of-life milestones, including initial production use where appropriate.

XCON is used to validate the technical correctness (configurability) of customer orders and to guide the actual assembly of these orders. It provides the following functionality:

- Configures CPUS, memory, boxes, backplanes, cabinets, and power supplies, disks, tapes, HSC/CI, printers, etc.
- Diagrams complete system configuration (Figure 1 overlays selected pages of a sample XCON diagram)
- Checks marketing restrictions, system building block menus, and prerequisites
- Assigns addresses/vectors and determines box power status
- Partitions multiple-cpu orders and cluster systems
- Determines and lists cabling information
- Lists components ordered with configuration-related comments
- Generates warning messages on issues affecting technical validity.

XSEL [7, 9] is used interactively to assist in the selection of saleable parts which make up a customer order.

It provides the following functionality:

- Allows interactive selection by generic component name, partial or full model number
- Performs completeness checking, adding and suggesting required parts
- Checks software compatibility, prerequisites, license and media completeness
- Checks standard and system building block menus, marketing and engineering restrictions
- Provides computer room environmental data and requirements
- Links to Automated Quotation System.

XFL is used to diagram a computer room floor layout for the configuration(s) under consideration. It provides the following functionality:

- Provides "minimum footprint" floor layout of components
- Allows custom rooms (user-specified dimensions and placement)
- Can include several configurations or a cluster in one site layout.

XCLUSTER is used to assist in configuring clusters. It provides the following functionality:

- Clusters multiple-node configurations for validation
- Specifies device quantity for dual porting
- Upgrades appropriate clusterable tapes.

In addition to these four systems currently in production use, two other configuration systems are under development:

XNET is an expert system which will be used to design local area networks, to select appropriate components for such networks, and to validate the technical correctness of the resultant network configurations.

SIZER is a research effort addressing the need for tools to assist in the sizing of computing resources required for any of a wide variety of uses in various types of organizations.

To support the ongoing development of the configuration systems in production use and to enable us to more effectively build new configuration systems, we have developed a software engineering methodology, called RIME [1, 12, 13], expressly for expert systems. Because RIME provides substantial insight into the engineering issues that require particular attention in building expert systems, it is discussed at some length in "The Engineering of XCON."

SCOPE AND USAGE

The configuration systems provide full product coverage for Digital's current product set. This product set today consists of 42 different families of central processor types and their supporting peripherals and software. In order to be useful business tools, released versions of these systems must include configuration knowledge of Digital's newest products by the time of product announcement. In practice, this means that CSDG provides major releases of these systems once each quarter, with at least one interim upgrade to insure adherence to the time-of-announcement requirement.

Hardware and software configuration is at the core of Digital's business. The configuration systems are used worldwide, throughout the corporation, by a broad set of users across the company's major functions: sales, manufacturing, field service, and engineering. The users of these systems perform functions which span Digital's complete order flow and manufacturing cycle, and, thus, are involved with many different business processes. This is a large and varied constituency to support—each has different needs and takes a different perspective on the configuration information provided:

- Sales uses the configuration systems as an integral part of the automated process to generate quotations for customers, and to insure that every order is technically valid.
- Manufacturing uses the information to verify buildability of all incoming orders, to understand physical partitioning of an order into various subassemblies to determine which plants should build which segments of an order, to guide the assembly of all orders, to determine the optimal set of diagnostics to run on each order.
- Field service has the perspective of consolidation and assembly of the order in the customer's unique environment and possibly with existing equipment already installed.
- Manufacturing and engineering benefit from the configuration systems' focus on system integration, as analysis of product knowledge for inclusion in the configuration systems identifies potential problems in system-level design and manufacturability.

This user profile has expanded dramatically over the years (see timeline). For example, the initial purpose of XCON was to assist manufacturing plant personnel in validating the technical correctness of system orders about to be built. Since then, this technical validation function has changed in response to business needs and is now performed in the field as well. In addition, the technicians in the plants who build the computers now use XCON diagrams to see how to put the systems together, and they include those diagrams in the shipment to the customer for use by field service installers at the customer site. XSEL was originally designed for use by sales representatives, and is now used by our OEM customers as well. Implementation of XNET will add specialized field support personnel to the user list in the near future. There are additional "indirect" users of these systems through automated linkages to other software systems (both traditional and expert systems) which depend on the configuration information supplied.

The configuration systems user base not only represents over 10 distinct business functions, but is also spread across the world, and has varied geography-specific needs. Overall, CSDG supports over 50 production installation sites, and the number is growing. In sum, then, the configuration systems are firmly em-

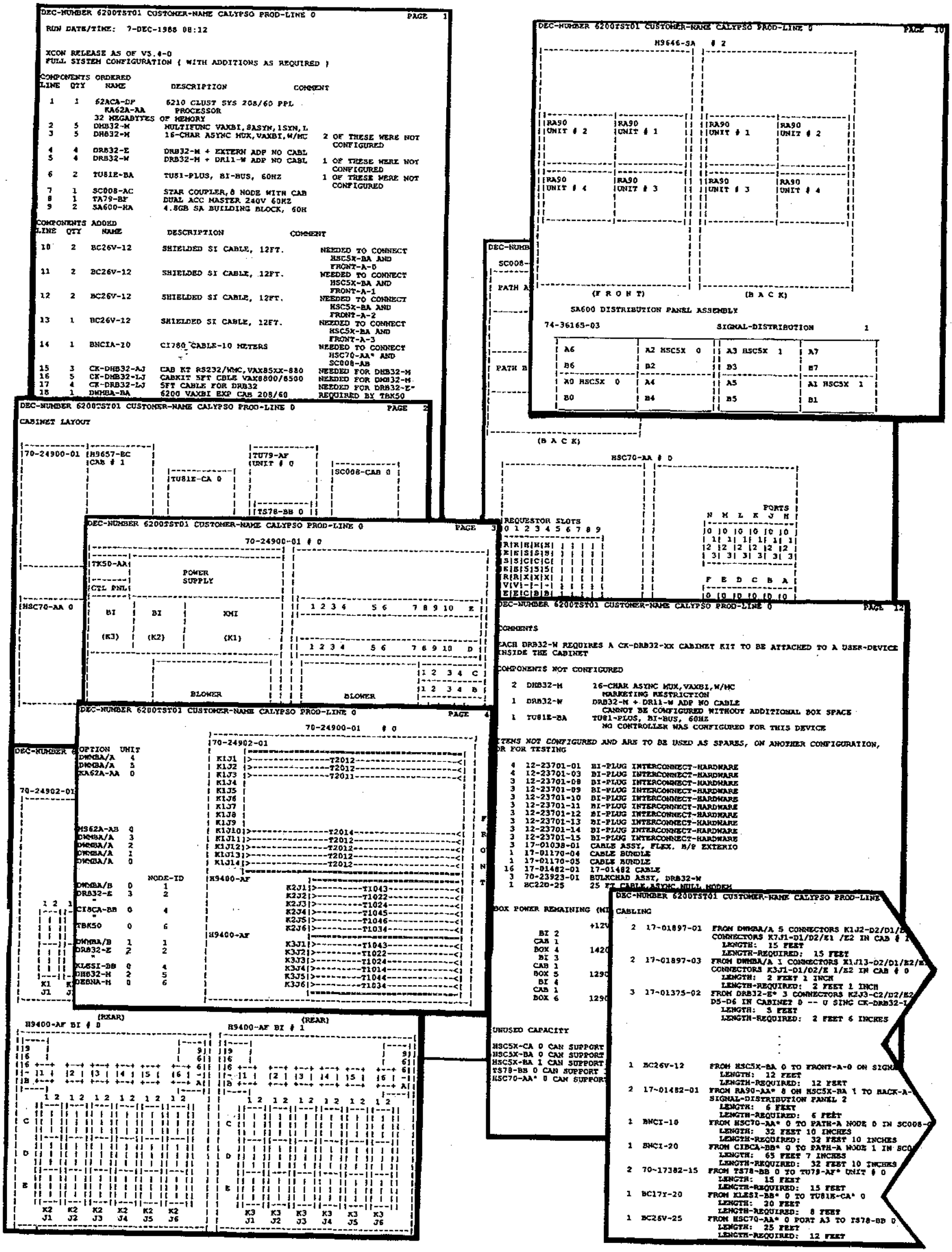


FIGURE 1. Selected Pages from a Sample XCON Diagram

bedded in Digital's most critical business processes. The existence of these expert systems has significantly improved these processes and continues to provide insight into opportunities for the future.

BENEFITS

Only a modest amount of attention has been given to documenting the benefits of applying expert systems technology [3, 8]. The configuration systems are a success and there is a major dependency on them within the corporation, worldwide. They benefit Digital in a number of ways, contributing to customer satisfaction, lower costs, and higher productivity. These systems are recognized as a critical factor in Digital's ability to maintain its highly successful *a la carte*, build-to-order marketing strategy (customized configurations to fit each customer's specific needs); this is one of the company's key competitive advantages. Some of the benefits are difficult to quantify, but overall the net return to Digital is estimated to be in excess of $40 million per year.

The use of the configuration systems insures that complete, consistently configured systems are shipped to the customer. Incomplete orders do not get through the process. In addition, XCON generates configurations which optimize system performance, so customers consistently get the best view of our products. Before the configuration systems, we would often ship the same parts configured differently. (There are multiple ways to configure the same set of parts to create a working system.) This was a major source of customer complaints and confusion, especially for OEM's who order large numbers of the same system and in turn reconfigure them all in the same way as part of their market-specific value added process.

The process of new product introduction is enhanced greatly by the focus on configuration information provided by the configuration systems. The existence of a single source of configuration information by the time of product announcement for use on initial customer orders simplifies field and manufacturing training needs and avoids confusion about new products which can delay time-to-market significantly. This is of critical importance given the volume of new products Digital continues to introduce each year. This single source of configuration information also increases manufacturing's flexibility by enabling product manufacture (and the knowledge of how it is done) to be moved from one plant to another without costly training or disruptive re-assignment of people. The use of XCON has facilitated this re-alignment of manufacturing capacity several times.

The use of the configuration systems has significantly increased the technical accuracy of orders entering manufacturing. Straightening out problem orders is a costly and time-consuming activity which disrupts the normal processes and increases order cycle time (i.e., delays order shipment). Overall, the additional discipline of both field and manufacturing use of these systems has shortened cycle times, contributed to more flexible, smoother-running processes, and lowered the number of people who would otherwise be needed for a given volume of orders.

XCON is seen as a critical component of Digital's current process of shipping segments of an order from various component plants, to consolidation points, and from there to the customer site. The use of XCON throughout the manufacturing process assures that when the components of the order come together for the first time at the customer site the system will work. Before the use of XCON, there were special manufacturing plants where every customer order was completely assembled, tested, disassembled, and repacked prior to shipping in order to insure the system would work when installed at the customer site. The elimination of this step in the manufacturing process (called Final Assembly and Test) has had a major positive impact on cycle times, inventory levels, and manufacturing costs, especially in light of Digital's four-fold increase in systems product volume since 1980.

Thus, the configuration systems are key in Digital's ability to handle the product complexity that technology advances are enabling and that our customers are demanding.

THE CONFIGURATION SYSTEMS EXPERIENCE

At Digital, we believe that our success in the application of expert systems technology is attributable to a conscious recognition of and careful balance among three important perspectives: strategic/business, technical, and human resource/organizational.

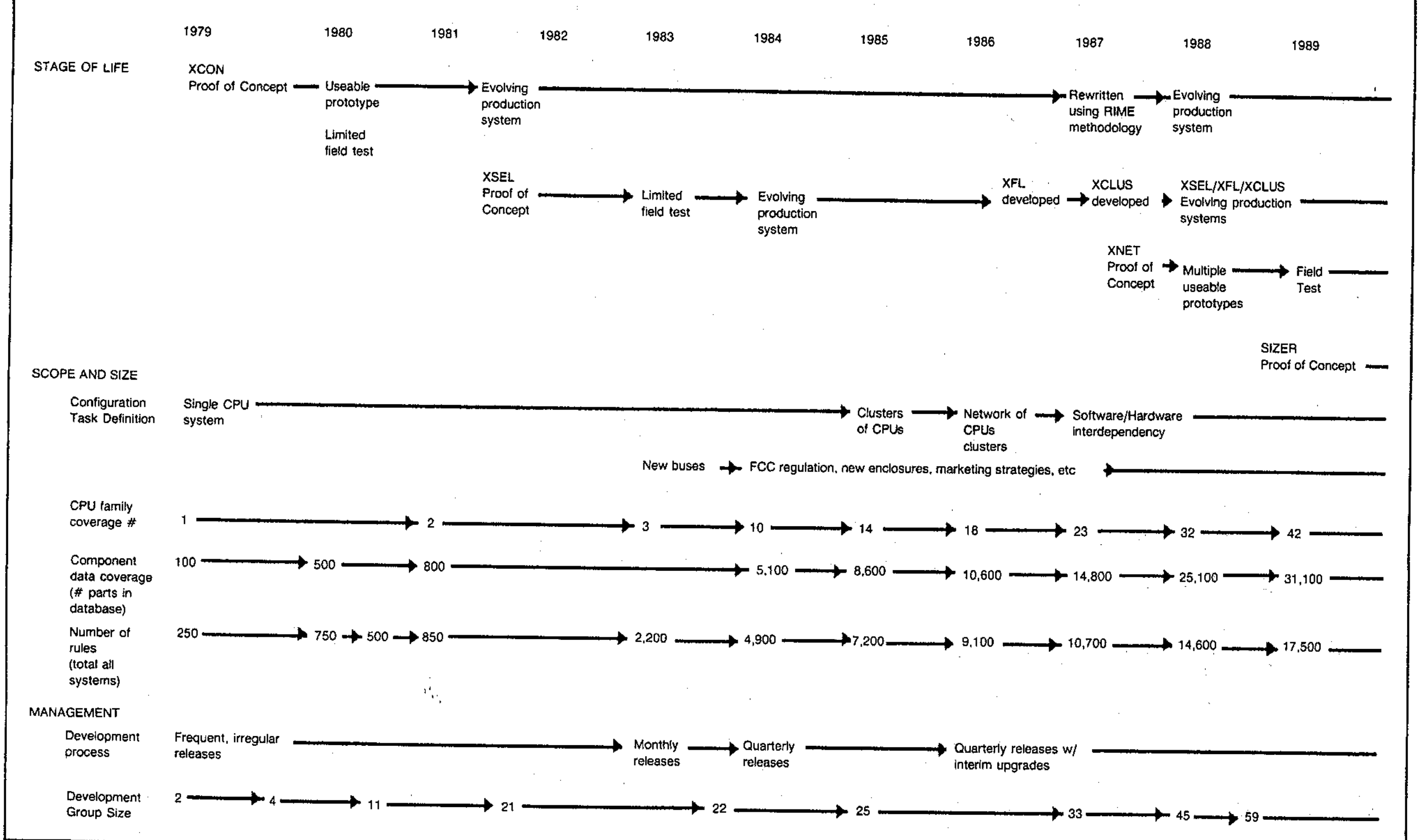
STRATEGIC/BUSINESS ISSUES

Digital has reaped and continues to reap enormous benefits from the configuration systems. It was, of course, an experimental effort at first but it was nurtured and allowed to progress and turned out to be a resounding success. Now Digital considers its use of expert systems technology a strategic investment.

What are the characteristics of an appropriate business problem for this type of strategic investment in a new technology? The problem must be real and systemic to the enterprise to justify the right kind of nurturing and supportive business environment. The solution will probably impact multiple organizations or functions and, hopefully, will bring about significant and positive changes in the way the enterprise operates. The configuration problem at Digital certainly qualifies. Digital's strategy of selling customized solutions is one of its critical competitive advantages, and gives rise to an indefinite number of valid configurations of its product set. In the 1970s, problems with handling this were starting to surface, and it was understood that automated support would be critically important as sales volumes increased and products became more varied and complex.

One of the main challenges we have encountered in developing systems which support strategic, cross-

DIGITAL CONFIGURATION SYSTEMS TIMELINE



functional business needs, is sorting out how the technologists connect with the business. As with any software development effort, it is necessary to form a solid partnership with the business being supported. Usually, that means with the end users and their immediate, operational management. In the case of strategic, cross-functional systems, we have discovered that the development effort must also be tightly connected at a more strategic level of management. This is simply a reflection of the fact that the problem being addressed has strategic impact and the solution may include and/or provide opportunities for far-reaching changes in the business [6] of which the tactically focused end user groups may be unaware.

CSDG is tapped into business planning at several levels to insure adequate awareness and connectivity and support. This has been and continues to be of critical importance to the ongoing success of the configuration effort at Digital. Much of the strategic connectivity over the years has been informal for the configuration systems effort: based on personal relationships with key individuals. Recently, we have initiated some efforts to formalize some of these processes. For example, we have set up a Configuration Systems Steering Committee made up of strategically focused managers representing our various business constituencies. This forum provides guidance on planning and priority-setting for the configuration systems in the context of a cross-functional vision looking out several years. In addition, a more formal corporate-level focus on system integration issues is evolving, which should provide formal long-term direction.

One of the most important factors in our success has been the existence of an enabling business/development environment for the introduction of new technologies like expert systems. During the early 1980s, Digital's manufacturing operations were open to embracing new ideas and technologies that would help them change the way work was done, affording higher productivity and shorter cycle times in response to customer requirements. Further, it was believed that this expert system technology could help shape the way Digital did business, allow the company functions to manage complexity, preserve the knowledge of its experts, and provide an approach to exception management more consistent with its growth demands. An environment has been provided which has nurtured the emerging technology and recognized that the people involved are managing change, evolution, and significant learning. It is understood that there will be new and changing operating norms and success criteria over time. Thus, at Digital there has been real management commitment to helping the technology succeed in helping the business progress.

TECHNICAL ISSUES

Technologies develop iteratively through a number of phases:

- Investigation and experimentation,
- Stabilization with limitations,
- Identification of opportunities for optimization, and
- Maturity.

No aspects of expert systems technology have really reached maturity yet. This need not be an impediment to its successful use, but it must be understood and proper expectations must be set. In a new and emerging technology, there are many opportunities for technical creativity as new discoveries and capabilities surface. There is also the likelihood that a serious application development effort will push at the limits of what the technology has to offer at a particular point in time, and in so doing, can contribute greatly to the further development of that technology.

Technology Challenges

The main challenges stemming from the emerging nature of this technology are:

1. Dealing creatively with the current limitations of the specialized languages and tools, and
2. Formulating an appropriate software engineering methodology (this is the thrust of the RIME development effort)

to produce efficient, maintainable, extensible expert systems.

Today's expert systems languages (and shells) offer varied but somewhat limited functionality. Most do not provide true extensibility to a production environment. There are very few knowledge engineering productivity tools available as yet on the market. The OPS5 language was originally chosen for the development of R1, the forebear of XCON. As XCON and the other configuration systems have evolved using VAX OPS5,® its strong pattern-matching capability and its speed have maintained it as our language of choice over all others on the market today. CSDG has directly influenced the development of VAX OPS5, based on our experience, and continues to do so.

Bachant and Soloway introduce a software engineering methodology appropriate for the development of expert systems. They make two primary points: (1) A task such as XCON's, where thousands of highly context-dependent decisions must be made in order to correctly configure a system, can take advantage of a language, like OPS5, that presupposes "situation recognition" rather than "algorithmic" control. (2) But, as the number of alternatives that an expert system must decide among becomes large, there is strong motivation to augment situation recognition with other kinds of control mechanisms. RIME, a software engineering methodology developed within CSDG, identifies three mechanisms that an OPS5 program can use to determine what to do and when to do it. This methodology, which was first used to reimplement XCON, has proven beneficial in the management of complexity and facilitation of change.

® VAX OPS5 is a registered trademark of Digital Equipment Corporation.

Testing, by traditional standards, is a major problem for expert systems. Can you test an expert system the way you evaluate a human expert? Since the understanding of the business problem is often enhanced over time by the use of the expert system, how can you define a set of tests that are sure to be relevant from one release to the next? Since there are, typically, an indeterminate number of paths through an expert system, it is not possible to exhaustively test all scenarios. What is correctness? Consider the relationship between the optimal solution and an accurate solution. An error can be difficult to distinguish from a wish-list item. The expert system may come up with the right answer, but for the wrong reasons. What about the cases where the experts disagree? There are other related testing issues as well, all of which cry out for new views of design-for-testability and new testing concepts, methods, and tools. This is assuredly an area which needs more attention, and we hope to devote more energy to this in the future.

Application Challenges

There are three application characteristics which are major contributors to the challenges faced in expert systems development:

1. Volatile subject domain,
2. Expanding functional scope, and
3. Large system size and complexity.

These are primary characteristics of the configuration systems, and have presented us with our toughest challenges over the years: (1) computer configuration is a dynamic subject domain; (2) once the systems were viewed as successful business support tools, users have wanted more and more from them; and (3) the configuration systems have become very large and are dealing with increasingly complex information relationships. Success has come only from our ability to respond to these challenges without disrupting or degrading the accuracy of the systems or our predictability in providing regular production releases.

Volatility

Each year, about 40 percent of the rules in the configuration systems knowledge base change (this includes rule additions and deletions, as well as rule modifications). This rate of change in the engineering, manufacturing, and marketing rules is due to a number of factors. Digital offers several hundred major new products and many more "minor" ones each year. Configuration information about them all must be included in the configuration systems knowledge base. In addition, sometimes knowledge about how to configure existing products must be changed to incorporate their configurability with new ones. Also, existing products are often re-packaged, as dictated by engineering, marketing, or government regulation. And, finally, CSDG periodically re-writes focused sets of rules as more is learned about the configuration of a particular product set and how to represent the knowledge more effectively.

There is another dimension of volatility in the configuration domain: the scope of the configuration task itself has changed significantly. The complexity and flexibility of Digital's product strategies has increased steadily since XCON was first designed. A single computer system can no longer be configured in isolation, but must be considered in relation to its role in a cluster or network. Clusters and networks offer multiple ways of configuring computers and connecting devices. Cluster and network connections are coming to be thought of as the main bus, with systems and devices providing functionality for one large "solution system." Moreover, distinctions between systems and devices are becoming less clear. The configuration-related interdependencies between hardware and software are much more numerous and complex today. In summary, the basic concepts and components of the configuration domain are being redefined continuously.

Expansion

The configuration systems we have noted serve increasingly varied user groups from different business functions and geographies. New types of users with new perspectives generate new requirements which frequently enlarge the functional scope of the systems. For example, when the plant technicians began to use the XCON diagram as the official document from which to construct the computer, a whole new set of requirements was generated. CSDG made an architectural change to the system in order to support these needs (altering some of the knowledge representation, transitioning functionality between the OPS5 rules and non-OPS5 code to create the diagrams, and developing a specialized database).

As their businesses change, existing users have discovered new and different ways to use the configuration systems. For example, as order volumes have increased and the technical validation function has changed, those users want to interact with the tool in different ways and need different kinds of information from the systems. They have asked for more streamlined ways to access the XCON results, as their processes have changed to preclude the need for steps previously performed and reflected in the XCON menu access scheme.

When XCON started feeding information to other systems, there have been instances where the rules have had to be changed to accommodate a new perspective on the configuration information. For example, there is another expert system used in manufacturing which helps judge how to split the construction of the order across several plants which specialize in certain sub-assemblies [4]. This system needs information from XCON on the physical partitioning of the order and what parts are used in what sub-assembly. XCON rules had to be changed to be able to explicitly provide this information at an intermediate point in XCON's overall process.

We learned that once we were successful, our users expected more. Users have higher expectations of a successful, operational system. Scrutiny has shifted from the accuracy of the knowledge base to things like ease of use, seamless integration with other software systems, etc. CSDG is in the process of re-designing the user interface across the full family of configuration systems, to make it more flexible and less cumbersome for users. Part of this effort includes a new interface architecture which will also respond to the increasing demand for more flexible, straightforward interfaces to other software systems. Also, various system functions previously considered luxuries, now become requirements. This can be anything from cosmetic features to customized output for different manufacturing plants.

Size and Complexity
The configuration systems are large and technically complex. We will discuss this from the perspective of the rule bases, the databases, and the traditional software programs and routines which make up the systems. It is hoped that this will give some perspective on the dimensions along which an expert system must be managed technically. Figure 2 gives a simplified view of the current configuration systems architecture (i.e., the systems' functional components and their interrelationships), focusing only on the systems in production use.

RULE BASES: The configuration systems rule bases are written in VAX OPS5. The specific rule counts (as of September, 1988) for each expert system are as follows:

XCON	10129
XSEL	3629
XFL	1808
XCLUSTER	243
XNET (prototype)	1700

Rule count alone is an inadequate characterization of a rule-based expert system. Representational complex-

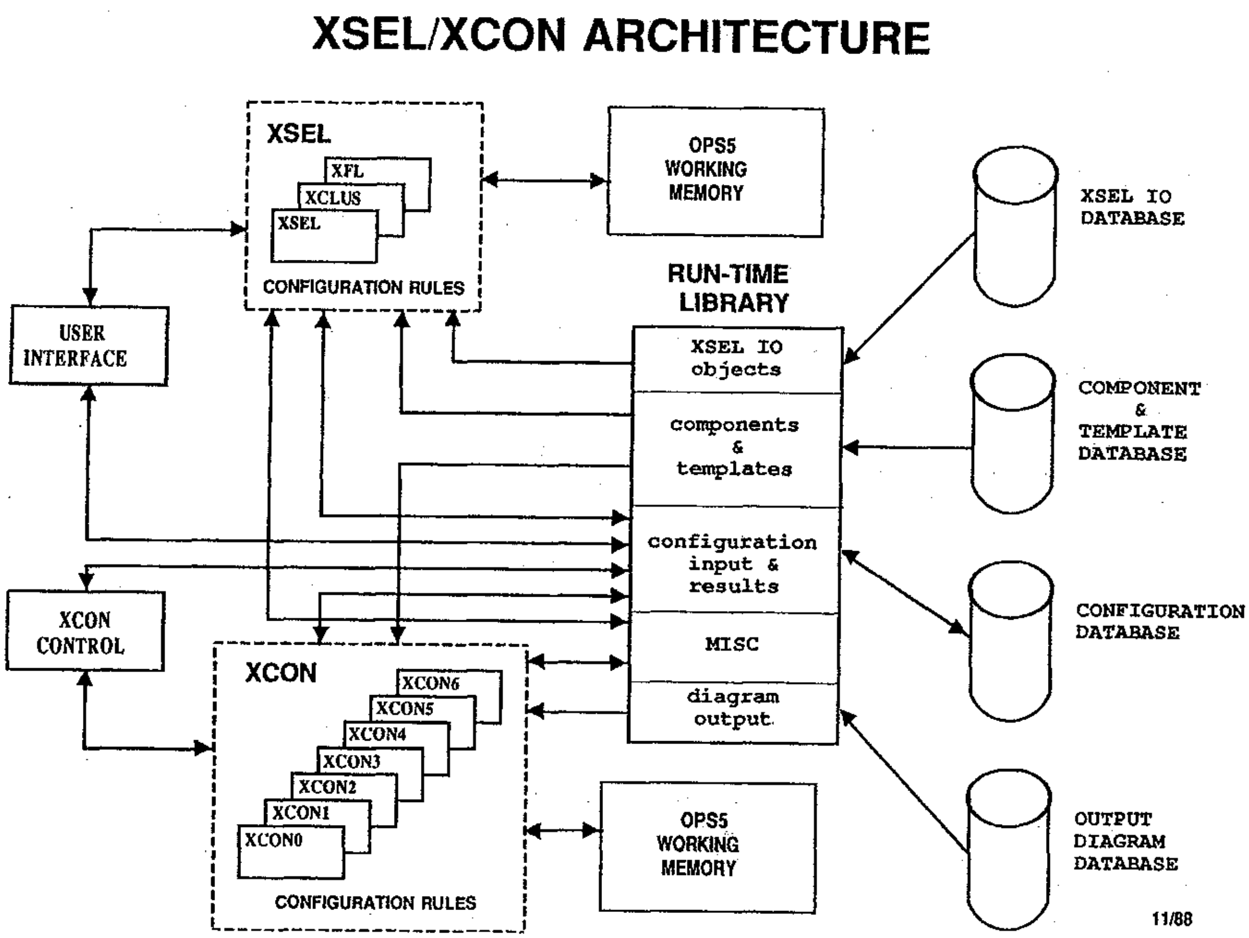


FIGURE 2. XSEL/XCON Architecture

Rule Name: *R1a-unmounted-ubx-options*

	Rule	Comments
	IF	
C1	The current step in the configuration process involves mounting options in containers;	*This condition is used to distinguish the group of rules that can potentially activate.*
C2	and the system being configured is not a vax11/780, vax11/782, vax11/785, vax8650, or vax8600;	*This rule is not applicable to those types of hardware systems.*
C3	and there is no unconfigured disk which sits on the idc bus;	*This condition insures that the rule will not activate before all disks on an idc bus have been configured. ("Disk" actually means "disk drive".)*
C4	and there is an unconfigured rl02-type disk which needs to mount inside a cabinet and whose pre-assigned controller sits on a unibus and it is the first disk assigned;	*This identifies the properties required of an appropriate disk to be configured by this rule.*
C5	and there is no unconfigured rl02-type disk assigned to a controller that is placed closer to the cpu than the controller assigned to the aforementioned disk;	*Disks assigned to controllers that are closest to the cpu need to be placed first, in case there is insufficient capacity for all of them.*
C6	and there is a requirement to cable the disk to be configured to a controller;	*Part of the activity of this rule is to determine that cabling. There may be several possibilities.*
C7	and there has been no connection made between the disk to be configured and anything else;	*This indicates that some of the activities that this rule will perform have not yet occurred.*
C8	and there is a controller to which the disk to be configured has been pre-assigned and which sits on a unibus;	*This identifies the appropriate controller.*
C9	and there is a requirement to cable the controller to a disk whose type and quantity of cable match one of the possibilities specified for the disk;	*This identifies the type and quantity of cable needed for this particular disk/controller combination.*
C10	and there has been no connection created yet to this controller from any disk;	*Another indication that the activities to be performed by this rule have not occurred.*
C11	and there are no unused disk spaces in any unibus cabinet;	*This insures that any spaces appropriate for disks in this type of cabinet will be filled before the rule can activate.*
C12	and there is a description for the capacity of a disk cabinet, whose name is not "h9643";	*This identifies a special type of cabinet that can only contain disks. An "h9643" is one variation of a cabinet to which the rule does not apply.*
C13	and there is an unconfigured disk cabinet;	*This identifies an appropriate cabinet in which the disk to be configured will be placed by the rule.*
C14	and the top space available for disk placement is unused;	*This identifies a location in the aforementioned cabinet where the disk to be configured can be placed. It needs to be on the top because of the removable medium.*
	THEN	
A1	mark the disk configured;	
A2	and update the top space in the cabinet to be used;	*The location needs to be marked so that nothing else will be placed there.*
A3	and create a connection relationship between the disk and its controller, fully specifying the identifying information for the disk, controller, cabinet, and the type and quantity of the cable to be used for the connection;	*This establishes the connection between the disk and its controller. Other rules will determine the length and choose the exact cable(s).*
A4	and create a containing relationship between the disk and the cabinet, specifying the identifying information for the disk and cabinet as well as the location of the placement;	*This establishes the placement of the disk in the cabinet.*
A5	and create labels for the output diagram showing the disk within the cabinet for both the skyline view of the cabinet layout and the detailed view of the particular cabinet.	*This insures that the output diagram will display this information correctly.*

FIGURE 1. Example of Pre-RIME XCON Rule

XNET (an expert system that configures computer networks—see Barker and O'Connor). We will now focus on two of RIME's major prescriptions: RIME's guidance in specifying control in a production rule context, and RIME's guidance in organizing a rule base.

NEW CONTROL TECHNIQUES

Problem-Solving Methods: Specifying Algorithmic Control

OPS5 supports "situation recognition" as its only form of control. Following the principles of *pure* production rule programming style, one should not force the firing of rules in a fixed sequence. However, the reality is that there are times when one wants to implement an algorithm, and thus execute a set of rules in some specific sequence. Rule developers did, in fact, use situation recognition to implement algorithm-like rule sequencing. What RIME has done is recognize that algorithms can and should be specifiable in a production rule context, and it provided explicit techniques for their definition. In RIME, pre-defined algorithms are referred to as problem-solving methods.

Problem-solving methods, then, are defined at development time—not at run time—and are composed of steps; each step of a method may contain one or more rules. Upon entry to a step, the rules fire (using "situation recognition") until all satisfied rules have been activated, and then processing proceeds to the next step. For example, in Figure 2 we present two rules from the "change-characteristic" problem-solving method used in XNET, an interactive system for specifying a computer network. This method allows a user to interact with XNET and change the characteristics of a part of the network (e.g., server, host, segment, etc.). The rule in Figure 2a is from the "ask-value" step of the method; in this step, the user is asked for a value of a network characteristic that the user wants to change. The rule in Figure 2b is from the "reset" step of this same method; as the last step it performs finalization activities, in this instance marking the characteristic as no longer identified as needing to be changed. There are also a number of other rules in each of these two steps.

Using this type of control reduces the amount of situation specification in the conditional part of the rule, since locating a rule within a pre-determined process eliminates the need for the individual rule to specify where it fits in that context. The individual rule's situation specification need not include reference to when it needs to activate in relation to rules in other steps of the process. In this case, without "algorithmic" control, one or more conditions would need to be added to the situation specification of the rule in Figure 2b to prevent it from activating before its time in the process (e.g., after the rule in Figure 2a fires).

Rule Name: Change-characteristics:ask-value:300b:default-units

	IF	*Comments*
C1	The current step of the process of changing the characteristics involves asking for a value;	*The current activity of the system is to ask the user for a value of a characteristic.*
C2	and the characteristic to be changed involves length;	
C3	and a default has been specified;	
C4	and the default has not yet been expressed in terms of units;	
	THEN	
A1	change the default to be expressed in units.	*For example, if an appropriate cable has a length of 40 inches, the default to the user will read "3 feet, 4 inches".*

FIGURE 2a. Example of RIME XNET Rule

Rule Name: Change-characteristics:reset:100a:reset-characteristic

	IF	*Comments*
C1	The current step of the process of changing the characteristics involves resetting after a change was made;	*The current activity of the system is to reset after obtaining a new value.*
C2	and the characteristic to be changed is still marked as such;	
	THEN	
A1	reset the characteristic so that it is no longer identified for change.	*After this rule activates, the characteristic that was specified by the user will no longer be identified.*

FIGURE 2b. Example of RIME XNET Rule

Deliberate Decision Making:
Explicitly Choosing Among Alternatives

As the rule in Figure 1 illustrates, XCON rules tend to be complex. This rule actually makes a number of decisions implicitly in the course of recognizing its applicability (e.g., selecting an appropriate disk, selecting a cabinet in which to place it, selecting a particular place within the cabinet), all in order to configure a disk. In effect, each rule that dealt with configuring a component needed to make sure that a constellation of components was available; such a need resulted in complex conditional parts of rules. Given the large number of rules that had such complex conditional parts, it became quite difficult to update the rules in light of new product announcements, etc.; it was hard to know if one had found all the rules that needed changing.

In contrast, RIME's philosophy is that complex rules need to be broken down; in particular, multiple tasks need to be factored out and each task needs to be made into an explicit, separate process. For example, in the new implementation of XCON, there is a set of rules that accomplishes the task of selecting a device (tape, disk, etc.), a set of rules that accomplishes the task of selecting a container for another component (cabinet, box, etc.), and another set of rules that accomplishes the task of selecting the particular location within the container to place the other component. Notice that all of these tasks are embedded in the conditional part of the rule in Figure 1.

The effect of accomplishing each of these tasks is to create intermediate results; components are selected, and made available to other tasks upon the completion of the task. In effect, these tasks work together to build up the set of components needed in order to perform the various activities involved in configuration. Thus, each rule can assume a certain amount of the specification as being accomplished; thus, rules, such as those in Figure 3, need only test those features of the state that are relevant to their specific task (e.g., selecting a particular disk).

Now that there are explicit tasks that need to be carried out, the question arises as to which tasks to execute and in what order. What RIME prescribes, in situations in which a choice must be made among competing alternatives, is that tasks to be accomplished are first proposed, and that there be an explicit arbitration process that examines the alternatives and selects one to carry out. We term this type of control technique "deliberate decision making" since we want to emphasize the need to have rules that deliberately examine the set of proposed tasks and decide, for specific, identifiable reasons why one task should be preferred over some others.

Putting It All Together:
An Example Illustrating Three Types of Control

The activity in the "propose-apply" problem-solving method provides examples of all three types of control techniques. Propose-apply was derived from the "universal weak method" developed by the researchers involved with the SOAR architecture [5]. The method is utilized fairly extensively in XCON because of the preponderance of complex decisions. The steps of the method include: proposing operators, eliminating the less desirable ones, selecting one, and performing the operation specified.

Figures 3, 4, and 5 present rules in three steps of the propose-apply method. The method, in this instance, is used for the task of configuring a device. The rules in Figures 3a and 3b are both from the propose step, and thus both put forth alternative suggestions. In this case, the alternatives are tasks themselves: 3a proposes to accomplish the task of selecting a device (tape, disk, etc.) to configure, while 3b proposes to accomplish the task of selecting a container (box, cabinet, etc.) in which to place a device. The rules in Figures 4a and 4b belong to a subsequent step (the eliminate step) in the method, and arbitrate between the two alternative tasks suggested in the previous step by the two rules in Figure 3. Deliberate decision making is this arbitration process: there are explicit rules that propose tasks, examine the proposed tasks, and choose which one to carry out.

Note in particular, that the rule in Figure 4a prefers one of the tasks proposed, while the rule in Figure 4b selects the

Rule Name: Configure-device:propose:200a:select-device

IF

	Rule	*Comments*
C1	The current step of the process of configuring devices involves proposing alternatives;	*The current activity of the system is to propose tasks that will ultimately result in the configuration of a device.*
C2	and no device has yet been chosen;	
C3	and there is at least one unconfigured device;	*Not all of the devices on the customer order have been configured.*
C4	and the process of selecting a device has not yet been proposed;	
C5	and no problem has been identified concerning selecting a device;	*If there is already an identified problem with selecting a device, the task should not be proposed again until the problem has been resolved.*
	THEN	
A1	Propose to go through the process of selecting a device.	

FIGURE 3a. Example of RIME XCON Rule

	Rule Name: *Configure-device:propose:500b:select-container*	*Comments*
	IF	
C1	The current step of the process of configuring devices involves proposing alternatives;	*The current activity of the system is to propose tasks that will ultimately result in the configuration of a device.*
C2	and there is at least one unconfigured device that needs to be placed inside a container;	
C3	and no container has yet been chosen;	
C4	and no device in which other devices can be placed has been chosen as a container;	*There are some large tape drives that have a compartment in the bottom suitable for disk placement.*
C5	and the process of selecting a container has not yet been proposed;	
C6	and there has been no problem identified concerning selecting a container;	*If there is already an identified problem with selecting a container, the task should not be proposed again until the problem has been resolved.*
	THEN	
A1	Propose to go through the process of selecting a container.	

FIGURE 3b. Example of RIME XCON Rule

other activity. The choice of whether 4a or 4b fires depends on the specific situation, (i.e., using situation recognition, these rules will fire depending on the details of the particular system configuration). Once the alternatives are narrowed to one, the apply step is entered. In this example, the rule in Figure 5 will activate since either alternative involves performing a new task. The new task will commence alternative involves performing a new task. The new task will commence (either selecting a device or selecting a container) and when it is finished, control will return to the previous task, configuring a device.

After, for instance, the task of selecting a device completes and control returns to the task of configuring a device, other tasks may be proposed, and then eventually performed, until all of the required intermediate results are available. At this point, the task of actually performing the functionality of configuring a device will be proposed and selected. Once selected, the actions (corresponding to those performed by the "then" part of the rule in Figure 1) will be performed: actually marking the device configured, placing it in the cabinet, connecting it to its controller and specifying the appropriate cabling, etc. This method is used for the configuration of other types of components, and at its highest level for the process of configuring a system.

In summary, there are three forms of control that developers using RIME can employ. If the task they are implementing can be accomplished through a sequence of steps with little variation, using algorithmic control is encouraged. The developer makes decisions at development time about what should happen when during execution. Developers are encouraged to make these algorithms as general as possible to allow for their reuse in other instances. If there is significant variation in what activities need to occur when depending on different inputs, situation recognition or deliberate decision should be examined. Each of these provides a means for the program to participate in the decision process. Situation recognition should be used if the conditions and actions are straightforward and limited. Deliberate decision is most useful for managing complex situations with many variations and interactions. RIME has identified guidelines for aiding the developer in deciding between the last two forms of control.

RIME's Guidelines for Organizing the Rule Base

In addition to making explicit three types of control, RIME also prescribes explicit criteria for organizing the rule base. For many years, XCON developers have been in the habit of attempting to physically co-locate rules that had some commonality. However, as the complexity of the rules increased and the number of them grew, it became increasingly difficult to actually co-locate similar rules. It was hard to determine the most critical similarity since there were so many dimensions to choose from. There were no defined (and agreed upon) strategies for the organization so there was little consistency among different developers. Also, the reason for co-locating specific rules was not explicitly recorded. Another developer needed to infer the commonality from reading the rules. Since rules developed under RIME were less complex with the reduction in the number of conditions needing to be specified and with the limiting of functionality per rule, it was easier to see how they could be organized appropriately. The use of the control mechanisms already provided some organization based on that dimension. However, within a step in a method there may still be hundreds of rules for whom the sequence of their activation is irrelevant. The rules in the same step need to be grouped along some other dimension. Moreover, the dimensions may vary among the steps. RIME requires that the criteria for grouping be explicitly identified and recorded in "subgroup schemas." These schemas, much like database schemas, provide abstractions describing sets of rules that allow developers to efficiently index into the rule base. The schema in Figure 6 presents the organization for the "apply" rules for configuring a device. These rules actually perform the functionality of marking a device configured, connecting it to a controller, placing it in a container, etc. These actions are similar to the actions in the rule in Figure 1, but they are distributed among many rules. These rules are grouped according to their functionality. The action of a "200" rule would update the status of a component, a