CAN BUILD -- A State-of-the-Need Application

Robin M. Krumholz

Digital Equipment Corporation
5 Carlisle Road
Westford, Mass. 01886

Abstract

This paper describes the development of an inventory management tool for Digital Equipment Corporation. The tool, CAN BUILD, began as a *rule-based prototype,* and evolved into an *integrated software solution* that addresses a critical business problem: inventory reduction. The CAN BUILD system illustrates the power of integration on two levels: integration of data from multiple sources, and integration of multiple VMS [1] technologies and tools in a single business application. The development history of the project demonstrates the need to *manage the introduction of new technology into a business process by evolutionary steps, over time,* and to *concentrate on the business needs,* rather than on state-of-the-art technology.

Background

In any industry where the product line is constantly evolving, the accumulation of inactive (slow moving and obsolete) inventory can be a materials management nightmare. As a product declines toward end-of-life, decisions must be made about adding material and/or labor to build it up, disassembling it (salvaging usable materials from obsolete parts), and selling back unused parts to other vendors—or writing it off. Understanding these complex tradeoffs is extremely difficult.

In 1985, Digital created a task force, the Inventory Programs Team (IPT), made up of experts from Materials, Finance, and Marketing. IPT's three-year mission was to find out what the company could do to reduce millions of dollars of inactive inventory. Their goal was twofold: to reduce inactive inventory to a minimum, and to design a process to maintain inventory at these levels.

[1] VMS, VAX OPS5, and VAX BASIC are trademarks of Digital Equipment Corp.

After a year of investigation and experimentation, IPT realized that they needed more sophisticated tools than quarterly aging reports and spreadsheet applications to achieve their goal. The process they had developed was time-consuming, subjective, and inefficient. It limited their efforts to a handful of products each quarter.

At this time, Digital's Applied Expert Systems Group (AESG) offered to assist the Digital Materials community by working with IPT to capture some of the IPT decision processes in a rule-based model.

The joint AESG/IPT project began in August 1986. AESG assigned a knowledge engineer to understand IPT's business by interviewing members of IPT and studying IPT's goals and methods. Together, they drew up a comprehensive vision to model the tradeoff decision process that IPT had developed. Figure 1 shows the model of this original vision.

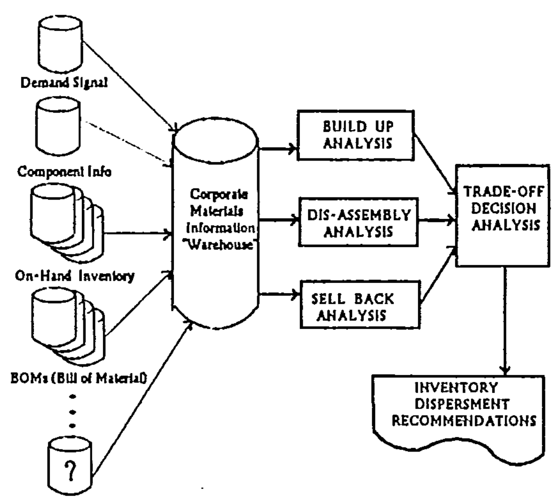

Figure 1—Model of Comprehensive Vision

The comprehensive vision included several different application modules that would share a common data environment. In the vision, each module was to analyze a different alternative for dispersal of inventory. Analysis results were to be used as input to the final tradeoff module. The tradeoff module would use *reasoning* to select the alternative that best satisfied various sets of business objectives.

Implementing this vision required, in addition to a significant development effort, the creation of new data elements and data standards that would have to bridge separate divisions of the corporation. AESG's experience and development strategy (Gill, 1987) suggested that the initial goal be narrowed in scope and focused to be attainable within a year. The first phase of the project was intended to test the feasibility of the full model. At this point, AESG identified the piece of the model that would have the greatest impact on IPT's productivity, and that would use only existing data. The original CAN BUILD system was the result of focusing on this narrowed model.

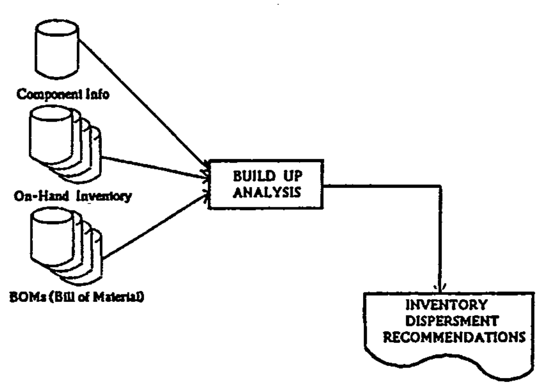

Figure 2—Scope of Phase I Project

Functionality

"Can-build analysis" is the process of simulating the financial impact of building various quantities of a product.

The CAN BUILD system is an interactive simulation tool that allows a materials analyst to play "what-if" games with alternative build plans. Detailed inventory data is supplied to the system in a monthly "snapshot" (extract) from each of the scattered stockrooms and inventory holders in the company. Product and corporate reference information is supplied by data extracts from a number of corporate data systems. Bringing all this information together into a single system creates a decision support environment for management that has *opened the door to a better, simpler way of managing inventory.*

For any given product, the system identifies the total inventory dollars on hand that could be used to build a given product. It separates those dollars into inventory that is unique to the product (that can be used only for that product) and common parts (inventory that could be used for other products as well). The analysis can be done with inventory across the whole corporation, or it can use some geographic or functional subset of inventory. The user can select a series of build quantities for simulation. For each build quantity, the system determines and tracks the inventory consumed, the additional labor and materials required, and the materials remaining.

CAN BUILD *precalculates* a set of recommended build quantities for the selected product—including the *quantities at which significant business milestones would be reached.* The system builds these numbers into its knowledgebase. By looking at these recommendations, users gain insight as to which build quantities they might want to simulate. [2] Through simple constraint optimization rules, the system can model the natural *manufacturing "food chain"* (all the plant sources that feed each other during the manufacturing process, from raw materials to finished goods) In addition, it draws inventory from the minimal of stockrooms to build any given quantity of the product. At any point, the user can request detailed information about the inventory used for the simulated build and the inventory remaining (the candidates for writeoff).

CAN BUILD differs from other materials systems in that its purpose is *strategic* and *tactical.* Before this tool was developed, materials applications that had access to detailed inventory data modeled only what was within the four walls of any single plant. Corporate evaluations and decisions were almost always based on "rolled up" (summarized) information, usually provided in hard-copy reports, and manipulated with spreadsheets. Management reporting was limited to what could be extracted from systems with an operational focus. [3]

[2] This methodology, and the recommendations that the system provides, are currently being evaluated for a software patent, and can therefore not be discussed in this paper.

[3] The Sloan School of Management at Massachusetts Institute of Technology refers to this approach as the "by-product technique."

Before this project, it was not considered necessary to use corporate-wide stock status data for online interactive query. The prime focus of CAN BUILD, from its conception, was to address management's strategic concerns - rather than the daily operational issues of any one plant or group of plants. (Daily operations are a different, and equally legitimate, business concern.) By demonstrating to management the benefits of pulling information from multiple sources into a single "data warehouse," we created the platform for this application and other similar decision support tools to be envisioned.

CAN BUILD in no way duplicates or replaces the systems that are used to manage materials within the individual plant. It provides its users with the capability of looking *beyond* their "four-wall" plant perspective, simulating various business alternatives, and seeing the impact of their decisions on all the plants above and below them in the manufacturing food chain.

System Design

This application exemplifies the power of integration on two levels: First, it demonstrates the integration of multiple systems sharing information over a network, without human intervention, as mentioned above. It also illustrates the integration of multiple software technologies, using the appropriate software techniques for each piece of the problem, to achieve a complete business solution. One can think of this application as using a multi-paradigm approach, where *traditional data-processing is one of the paradigms.* We prefer to describe CAN BUILD as an *integrated software solution,* rather than an AI or expert system, because the rule-based part of the system is only one of several elements contributing to its success. The VAX OPS5 language interface provides the flexibility to use other VMS languages and tools; the developers were thus able to choose the most appropriate methodology and tools for each function.

The system architecture is a layered modular design, which separates the *data capture* functions from *data access* and *data manipulation.* All data, including the results of any simulation analyses, is maintained in traditional data files that can also be shared by other applications. Figure 3 shows how the architecture is layered, and how the rule-based elements of the system are isolated from the data environment.

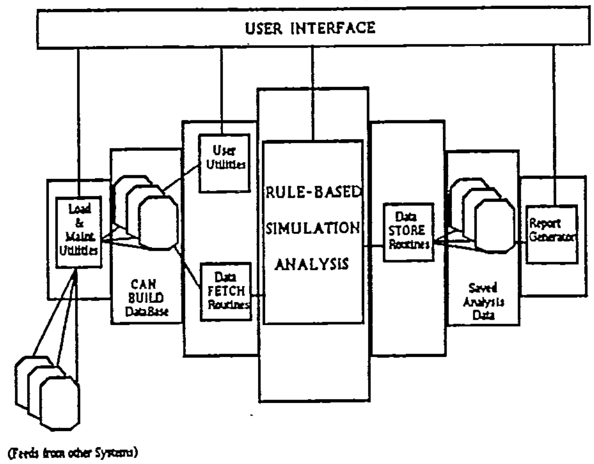

Figure 3—CAN BUILD System Architecture

The large central box in Figure 3 represents the core of rule-based modules in the system. On either side of the central box is the interface layer, which passes information in and out between the VAX OPS5 environment and the data layer. The use of data layers gave the developers the flexibility to change external data extracts (in the outermost layer) and to refine the rule-based modules, without adversely affecting the other parts. Data layers also allowed users and external applications to share the same data. The last element is the report generator. Although our solution has report generating capabilities, users can use any fourth-generation language reporting tool for customized output. The system is bound together by a menu-driven user interface designed by the users; it is extremely easy for any materials analyst to use.

AI and the Triangle of Change

When one looks at the quantity of rule-based code that was used in the baseline implementation of CAN BUILD, relative to the quantity of traditional software tools, one might indeed question whether this application should be labeled an *expert system.* While the initial prototype, consisting of some 200 rules, was written entirely in VAX OPS5, only 20% of the system (approximately 300 rules) was coded in a rule-based language in the baseline implementation. Less than half of those 300 rules (about 10% of the system) could be considered to do any form of reasoning.

Initially, these statistics were disturbing to those who wanted to call CAN BUILD an AI solution. However, the percentages fit well in the "Triangle of Change" model developed by Dr. Gregory Gill at the University of New Hampshire. (See Figure 4.) As knowledge engineers, AESG has been using this model to better influence and manage the organizational change associated with the development and use of expert systems within Digital.

The Triangle of Change model shows that to successfully effect change, the affected organizations must value making investments in *working better* and *working differently* (Gill, 1987).

According to this model, only 70 to 85 percent of the resources in a business entity should be dedicated to the daily work necessary to get out the product. The remaining 15 to 30 percent of resources is further broken down into two parts, with from 10 to 20 percent dedicated to improving the current processes, and *only 5 to 10 percent representing real change.*

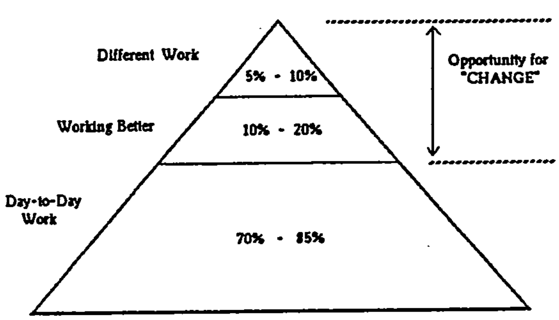

Figure 4—The Triangle of Change Model

We use this model to show our sponsors that if they spend too much time getting day-to-day work done, little room is left for change. They must therefore respect the time required for projects that will help bring about change, and commit themselves to invest people and time in the project.

However, the model is also intended to show that if the investment in changing the current work is too great, it may distract from getting the day-to-day work done and lower business productivity. More important, change is likely to be more effective if it is *evolutionary. Too much change too soon may be rejected.*

When we looked at the use of AI methodologies in CAN BUILD as a business solution, with the triangle of change in mind, we saw a direct correspondence to the model's implied recommendations for *evolutionary* change. We propose that *having limited change* (AI) helped build the credibility of the CAN BUILD system with the materials analysts and made it more acceptable to the Materials business.

Development Strategy

CAN BUILD was developed through a *participative design* approach, using frequent interviews to acquire the necessary knowledge and iterative prototyping to define the tool.

The project proposal, a simple, nine-page document, was accepted by the IPT sponsors in September 1986. The proposal included the original model, a description of the AESG development strategy, a brief description of the business needs that had to be met, and most important, a *definition of success,* agreed to by all those involved.

The description of the business needs included a list of ten materials management questions for effectively managing inventory. These questions were difficult or impossible to answer using existing tools and processes; at the time, the project team had only a very vague idea as to how to find the answers. Over the next six months, the developers would help the rest of the team experiment with different ways to arrive at the answers.

The definition of success was the result of a brainstorming session on the question: "If we are successful, how will we know it? What will be different two years from now?" IPT and AESG put together a list of business goals as criteria. Part of our strategy was to focus on the *critical success factors of the business* (Rockhart & Bullen 1986). The business goals that our team laid out in that nine-page project plan helped to keep us all grounded, and to keep our common goals in mind as we developed the functionality of the tool.

This document remained the only formal documentation on the system until after we implemented it the following year. Another part of our strategy was to bypass the the need for a formal functional specification. The prototype would serve that purpose.

The first prototype was developed entirely in VAX OPS5, using artificial data, in just seven weeks. This prototype captured the essence of the desired functionality. When it was demonstrated to the project sponsors, they gave their approval to get the tool working with real data. It served as a functional specification for subsequent efforts.

A second developer was brought onto the project. It took six further iterations on the design, over the next four months, before the team developed an appropriate knowledge representation of the problem, the baseline functionality was agreed on, and the developers began their final design for implementation.

At the start of the project, the team had had only a vague understanding of all the functionality required by the comprehensive vision. Experimenting with flexible data structures in a rule-based paradigm was invaluable. With each new design, the need for traditional data processing grew. As the functionality became well defined and stabilized, we wrote more and more of the solution in VAX BASIC, focusing the VAX OPS5 environment on those aspects of the system where rules provided an advantage.

The baseline system was implemented in September 1987—meeting the one-year goal. In the three remaining months of 1987, IPT used the tool to analyze some fifteen end-of-life products. By identifying opportunities for revenue from inventory that would otherwise have been written off, they recouped for the company more than 30 times the cost of developing the tool in those three months.

Business Impact

The original definition of success for this project was to increase the productivity of the IPT materials analysts (our "experts"). The analysts, when we first met them, were spending six to eight weeks analyzing a single product. As a team, they were able to make decisions about only two or three products each quarter. They needed easier access to the data, and they needed to be able to make timely, consistent decisions from a corporate-wide perspective.

CAN BUILD enables the analyst to do a product analysis in less than a day, with far greater accuracy and detail. Decisions can be made about a specific product at regular weekly meetings, or as the need arises.

The success of CAN BUILD is evident. However, the payback is only one measure of the system's impact on the corporation.

The tool also enables the analyst to look at the entire product pipeline, using data that was previously not readily available and information based on calculations that were difficult or impossible to make. The result has been a more thorough picture of the potential impact of each decision. However, not even these capabilities truly measure the success of the project.

The ultimate contribution of this system is the insight it has provided for the future of materials management. Once the analysts began to experiment with this simulation tool, they were able to better understand the causes of our end-of-life problem at Digital. If they could monitor and synchronize inventory throughout the manufacturing life cycle, then the end-of-life decision would be simplified. By using this same tool to periodically monitor all major products, they could reduce the amount of inventory accumulated, and they could thus simplify, if not eliminate, the need for the writeoff decisions that drove the original model.

In March 1988, six months after the initial implementation, the original vision was refocused to become a proactive "inventory goodness barometer," rather than a reactive decision support system for inventory tradeoffs. Use of the tool began shifting away from responding to symptoms, and toward curing the disease of excess inventory.

Until now, the analysts in Digital's manufacturing plants, with the support of IPT, had all been sharing the single corporate implementation. We gradually made CAN BUILD available in the plants. With eight sites running the system, we began accumulating much new expertise. We started to enhance the functionality and work toward a new, improved model. IPT had satisfied its mission and began to phase out its role.

The system has proven itself, and the analysts are ready to trust it. They no longer feel the need to use the tool primarily for interactive trial and error; they are beginning to relinquish their control and step back to let the system do the analysis. They have begun to see the advantages of having the tool provide better recommendation and explanation capabilities, and additional functions that depend on the AI knowledgebase. They are now looking for more automatic features.

Conclusions

What distinguishes this AI effort from some of its predecessors is that with this project, the developers continually compromised with the users on the functionality they wanted to provide, in favor of the functionality that the users required for their business. Because we saw ourselves, not only as technologists, but as *agents of change,* we studied and applied many change management concepts in our project decisions.

More important, we focused on the "state of the need," rather than on the "state of the art." We allowed the reasoning component of the system to become a minor part of the *full business solution,* and *took ownership of the total solution, rather than just the AI part.*

Although we always tried to open the users' minds to new opportunities by prototyping more creative solutions, we *listened to their requests* —and often yielded to a simpler, more traditional approach. The users had little awareness of where the AI was; they didn't care. We let go whenever we saw signs that the users weren't ready to take the next step. But we left a trail of seeds behind us, and those seeds did not long lie dormant. We focused on the business "hot buttons" that would generate the largest payback in the shortest possible time, and we used our newest technologies only where they were needed. We valued ourselves, not just as "knowledge engineers," but as creative software applications developers, with a full workbench of tools, conventional as well as AI, at our disposal.

Postponing the development of our system's expertise was a conscious—if very difficult—choice that has in retrospect paid off handsomely. We see the proof of our strategy when we look at how time and experience have watered the seeds we planted in our prototyped garden, and now, one year later, we are beginning to see the little sophisticated flowers blooming in our user's requests. (Better yet, many of the system's users believe the ideas are their own.)

The difference was *timing,* for now *the users are ready* (Conner, 1985). They feel true ownership of the system and the plans for its future. We gave them a system that could be compared to a new college hire, as opposed to a Sloan Fellow, and over time, they took ownership of the continued development of the new employee's expertise and awareness. Our system seems to have avoided the resistance and resentment often shown toward senior newcomers in an organization. It hasn't suffered from the classic perception of AI that it is just another "textbook genius" that doesn't fully understand the real world. Instead, it is welcomed as a naive but important resource with infinite potential, which the users, as the experts, are eager to train.

Each new release of the software promotes AESG's "state-of-the-need" system to a higher level of awareness and responsibility. Someday soon, as newer hires are trained as materials analysts, the new generation of users may think of our system as an expert.

References

1. Conner, D.R. 1985. "Managing Organizational Change: Dangers and Opportunities." Atlanta, Georgia: O.D. Resources, Inc.

2. Cooper, T.A. & Wogrin, N. 1988. Rule-based Programming with VAX OPS5. San Mateo, California: Morgan Kaufmann Publishers, Inc.

3. Gill, G. 1987. Orchestrating the Dual Perspective of Change from Theory to Practice. University of N.H. (Contract # 449671 with Digital Equipment Corp.)

4. Reitman, W. 1983. Artificial Intelligence Applications for Business. Norwood, New Jersey: Ablex Publishing Corporation.

5. Rockhart, J.F. & Bullen, C.V. 1986. The Rise of Managerial Computing: The Best of the Center for Information Systems Research at Sloan School of Management, MIT. Homewood, Illinois: Dow Jones-Irwin.