ETM

ENGINEERING & TECHNOLOGY MANAGEMENT

Title: Knowledge Sharing System for PC Compatibility - Solution Mechanism Model

Course:
Year: 1990
Author(s): C. Scott

Report No: P90029

ETM OFFICE USE ONLY	
Report No.:	See Above
Type:	Student Project
Note:	This project is in the filing cabinet in the ETM department office.

Abstract: This report explores the feasibility of an Expert System to solve an information sharing problem. We conclude that Expert Systems will accomplish necessary tasks, and development of a "Knowledge Sharing System" is the right design for the Expert Systems.

KNOWLEDGE SHARING SYSTEM FOR PC
COMPATIBILITY-SOLUTION MECHANISM MODEL

C. Scott

EMP - P9029

Knowledge Sharing System for PC Compatibility--Solution Mechanism Model

Project Report
EMGT 510E

By
Cathryn Scott

System Purpose

Compatibility as a computer term has been around since IBM introduced the first personal computer (PC) and published the technical details for the computer architecture. Their purpose was to set the standard that all future PC manufacturers and software developers would follow. "IBM compatible" was a term used to market every PC clone and new software package that came to the market for a good part of the 80's. The PC industry consists of software vendors, peripheral manufacturers, developers of add-in cards, and motherboard and system designers. The old AT bus standard has given way to ISA, EISA, and MCA standards.

There are several operating systems that the new purchaser of a PC can choose from--UNIX V.3, UNIX V.4, SCO UNIX, ISC UNIX, XENIX, OS/2, DOS 4.01, DOS 5.0, Extended DOS, Windows 3.0, and the list keeps growing. The purchaser of a new personal computer expects all existing software packages used on their old PC to work on their new system. They expect previously purchased add-in cards to continue working, along with any new device they just read about and purchased. We entered the 90's with one enormous, confusing mess, and anticipate no foreseeable end.

There are seven organizations within my company that are concerned with compatibility standards. Each organization is concerned with various parts of computers, from CPU chips to complete personal computers. Each groups has their own testing labs and engineering resources. Most groups have their own technical marketing support team and design engineering staff. They each have their own market base, but they all share many of the same compatibility problems. There is a need to share the knowledge of problems found among the teams in all the organizations.

Attempts have been made to coordinate the knowledge. The Managers meet once a month to share information, including compatibility issues. A weekly phone conference is held with representative engineers from each group. There is communication among the design, marketing, and testing groups within each organization, but nothing across the company. We need a way to communicate problems and possible solutions company wide.

My project is to propose the design of a Knowledge Sharing System that will enable us to not only share knowledge that we gather, but be able to use the knowledge to make intelligent judgements concerning the potential problem found.

The problem I want to solve is very complex. The experts, in most parts, are the users. The expert can also be a large software vendor who decided to do something a certain way and refuses to change. If they are powerful enough, the industry will conform and standards will change. We must capture that knowledge in our KSS.

System Functions

System Objectives

The objective of this KSS is two fold; first the user must be able to query and obtain suggestions to help identify compatibility problems; second, the user should be able to input a found problem and receive suggestions to probable cause. This means, for the objectives to work, the knowledge shall be represented so that the user can enter *If [problem symptoms] then (probable solution)* as well as enter *If [solution] then If(possible symptoms) then (possible cause).* Problems are irregularities found during testing or reported by a customer. The probable cause is what is inducing the irregularities. If the KSS is to be of benefit to the entire corporation, the solutions shall be based on information gathered from many sources, including readings.

Currently all problems found are stored in a simple database. This information contains the configuration of the unit under test, UUT, and is an important link between past problems and UUT. One can do a simple query and ask for all problems on Unit A and receive that information. Or, just as easily, ask for a list of all problem reports that used video card B and failed test X,Y or Z. From the information gathered, an experienced tester could develop assumptions to either of the two objectives stated above. One can't argue that this is enough and doesn't warrant the expense of the proposed system if they noticed the phrase *experienced tester.* This takes six to twelve months, and the experience is in only one area of expertise. Also lacking is information from other sights. The next argument might be to network a common database. This position has been analyzed, and the complexity of how to manage the information into a coherent form, understandable by different types of expertise, doesn't justify when looking at the quality of the information. Once again the information would only be from actual testing.

This information must be shared across departments within a facility, as well as other facilities. Since the experts will be many of the users, we will provide a link between development engineering, technical marketing, and testing. This link is currently missing and a database solution would not have the ability to give a gift of extra intelligence to the novice tester or experienced engineer.

Knowledge Representation

Scope of Knowledge

The knowledge obtaining function of this KSS will be no easy task, since the experts exist in many disciplines. The knowledge base must be continually updated, with new information, as technology grows and new software and hardware becomes available.

It is essential that information be obtained from research and readings. In the PC industry there are several good publications, such as *PC Weekly* and *PC Magazine*. These periodicals give the consumer a complete update on new PC technology and software packages, as well as, compatibility problems found in the industry. Information must be dissected into an understandable format, as if gathered from an expert, and input into the KSS. Knowledge must also be gathered from the customer. The expert becomes the technical marketing engineer who listens, verifies, and finds solutions to each customer call.

As previously stated, one of the important users is the tester; yet the tester seldom resolves a problem. The resolution comes from the responsible engineer assigned to the project. This person interfaces with experts from different disciplines within the organization. Through out this process to discover the reason for the test failure, the engineer becomes an expert and his or her knowledge must be obtained.

Another source of information comes from employees who have contacts within other companies. The wide scope of where the knowledge is, exhibits the importance, to the organization as a whole, to have one source to obtain expert assistance.

The complexity of gaining the knowledge will be a difficult maintenance task, since most input will come during the maintenance phase. Yet one of the most important parts of maintaining the KSS will be the cleansing of outdated knowledge and should not be ignored in analyzing the cost to maintain the proposed system.

Structure of Knowledge Base

Except for the output, the structure of this KSS based system shall exist as a risk based system[1]. While the output of the risk mechanism model are risks, the output of the compatibility model will be suggestions of solutions for potential problems found during testing and will be known as the Solution Mechanism Model.

---

[1] Niwa, Kiyohshi, *Knowledge-Based Risk Management in Engineering: A Case Study in Human-Computer Coopereative Systems,* John Wiley & Sons, Inc 1989.

The standard work package method developed by Niwa and Okuma[2] will be used to map solutions to testing elements. The work package method contains two matrices, each of which is a standard work package matrix. The horizontal axis represents activities which, in this case, will be the test performed. The vertical axis represents objects, which will consist of parts of the items under test. For clarity, one can think of the horizontal axis as software and the vertical axis as hardware. Since the "work package" is a subset of the product of objects and actions, it will become the link of how the software and hardware interact.

There is an added complexity using this type of solution. In some cases, but not all, the BIOS interacts between the hardware and software. The BIOS is software, and many compatibility problems found are related to errors in the BIOS. We need this information, and to fit into a workable scheme, the BIOS should be thought of as an object not an activity. This is because it is a part of what is being tested.

[2] Niwa, K., and M. Okuma, "Know-how transfer method and its application to risk management for large construction projects," *IEEE Transactions on Engineering Mangement,* EM-29(4), p. 146, 1982.

An example of an activity or software test:

$$A = \{a_1, a_2, a_3, \ldots\} = \{\text{First Choice, QAPlus, Windows 3.0}\}.$$

The modem portion of the First Choice software package is used as part of the modem test suite. QAPlus is diagnostic software for reporting status of memory, ports, adapter cards, and low-level hardware setups. Windows 3.0 is a multitasking windowing environment and the tests are designed to report irregularities under this environment.

An example for the object set for represented hardware or BIOS:

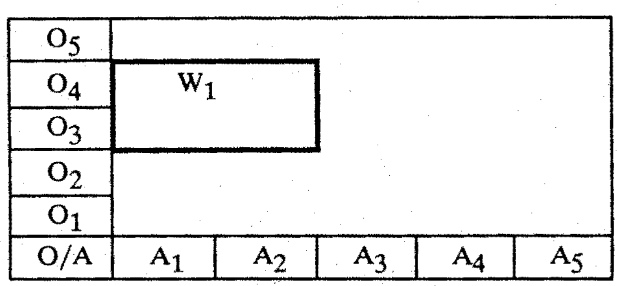
$$O = \{o_1, o_2, o_3, o_4, o_5 \ldots\}$$

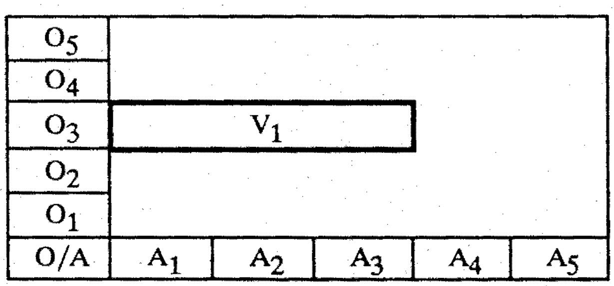
$$= \{\text{ DPT SCSI Controller, Onboard Video, COM ports, BIOS, Video BIOS }\}.$$

An example of the work package for a problem found during running the modem tests:

$$w_1 = \{(a_1, o_3), (a_2, o_3, o_4)\}$$

$$= \{(\text{First Choice, COM ports}), (\text{QAPlus, COM ports, BIOS})\}$$





If the Knowledge Set is represented as $K = \{k_1, k_2, k_3, \ldots\}$. and v_j is defined as the domain of work to which knowledge k_j corresponds, then v_j is a member of the set V where:

$$V = \{v_j \mid v_j \; A \times O\}$$

In the above example the solution k_1 occurs in the domain of v_1.

The relationship between these will become clearer in the next section by using the above sets as examples.

Example Usage

Sample Users and Tasks

For example purposes, the user will be an engineer from the sustaining department who has just received two problem reports from the compatibility testing lab that require investigation. The first report states that while running the First Choice Modem Software, the test failed trying to dial out on Com1 and Com2. The report also states that if Com1 was disabled, then dialing out on Com2 was successful. The second problem report states, while running the QAPlus Diagnostic software a failure with the serial ports was identified.

Sample Input/Output

Based on the example sets above, the input and output could be as follows:

IF (First Choice and QAPlus) then
1) Possible problem with COM ports.
2) Potential BIOS problem.

At this point there are many possible identification factors to the actual problem, but based on the two reports, the engineer would probably choose to look at the COM ports first. The first inquiry for knowledge was based on the logic **if A then B**. The Engineer can now obtain knowledge to the possible cause where the cause is C. The logic is **if B then A then** C.

This sample is based on a real situation. The First Choice Modem software makes a BIOS call that disables the interrupts and the BIOS fails to report the appropriate Com Port status which, in turn, prevents the communication package from dialing out. This was discovered from the possible solution **B**. There was also a problem with the Serial Chip Set. This was a very difficult problem to debug because the errors existed across several models of computers. This was the possible cause **C**. On two models it was determined to be the Serial Chip Set. On four models it was a BIOS problem.

Expected Benefit of usage

The benefit of the KSS will change as the knowledge increases. The first time this occurred, the engineer wouldn't have been able to find the possible cause. Once the information was known, the possible cause C would have been entered in the KSS. The significance of being able to obtain the solution at a later date can be demonstrated using the continuation of the above problem as an example. There is a large staff of testers, technical marketing engineers, design engineers, and sustaining engineers. The Serial Chip Set problem was identified by a design engineer. The BIOS problem was found to exist on production boards and under the control of the sustaining engineers. The next time the Serial Chip problem happened was on a board that had just gone into production and was under the sustaining group. They didn't know that there was a problem with a specific Serial Chip Set and several days of time was wasted finding the cause. The tester was a different person both times the problem was detected.

The work package relationship, showing the relationship between tests and hardware, can be obtained during the implementation phase. The mapping to solutions must come from the other sources and won't always be known until investigation of a problem happens. From the information gathered, the KSS user would have been advised to test past versions of the BIOS and found that the failure did not occur. The information on the serial chip problem wouldn't have been entered into the KSS system until the problem had been identified. The benefit would be felt during future test runs of other boards with the same chip set. This benefit would be hours of saved investigation time.

System Planning Phase

User-Oriented

The reason a Knowledge Sharing System was suggested, rather than a Knowledge Based System, is the user has the knowledge, and in most parts, is the expert. The user base is large, spread across multiple sights, and the users are engineers and technicians. Management will supply expert knowledge, but will depend upon the their subordinates to use and maintain the system.

One of the largest failures of such a system is getting users to support and utilize the system after its design and development. There are several reasons this won't happen. Many of the users will be on the design and development team. This is possible because many of the users have the expertise to design and develop such a system.

Because this project will be used in different locations around the country, the design team should consist of representatives from the locations. Modern communications make this possible and insures that each group accepts the project. While this might not work in all companies, the engineers are already working on joint projects and function well in this environment. By adding technical marketing representatives to the team, we gain their support. Structuring the decisions so that all potential users feel part of the final project is very important. Though it increases the chance of confusion and delay, a well experienced facilitator can keep the design team moving.

Top Management Support

Top management support should not be a problem. The need for an Expert System was part of a presentation to Corporate Staff. If the overall proposed project is approved, then there should be no resistance from the current management staff. Maintenance will be expensive and require two full-time people. This also should pose no problem, as it will be understood from the beginning. There will always be the risk that if top management changes, the project will no longer be supported. Top management doesn't change very rapidly, however their goals and priorities do. This will be a great risk, but not enough to warrant dropping the project.

One important aspect to consider when presenting this project to top management for financial support is that this project would be classified as research and development. Because of this, there will be tax breaks and financial paybacks to consider.

Knowledge Supplier Cooperation

The initial knowledge gathering phase will be tedious. Once the objects and actions are identified, the work packages must be decided. This should be considered part of the design phase, because that team has the knowledge to do this. This is also where other users who are not on the team can start participation. I feel that since the designers, users, and knowledge suppliers are pretty much the same people, the pride of ownership should eliminate any cooperation problems.

The incentive for the knowledge suppliers to join or cooperate with the project team is very high, because of the basis for the reward system that already exists within the company. Faster, better and cheaper are words our groups hear all the time. This means improving our processes so that we complete our work faster and better. It also means if you accomplish a task, produce better quality workmanship and you did it faster, the final product cost will be cheaper. There is an interpretation for each department from design to manufacturing. The goals of the compatibility group are to improve our tests and problem detection by writing better tests, organize and automate our efforts to shorten our overall test time, and fine no new compatibility problems from second fab to production release. This will save expensive reworks and design changes. This system will help every user improve the efficiency of their job and save precious hours of resource time. Those who show proven success at reaching the goals above are the ones who will be rewarded at review time.

System Development Phase

Incremental Development

Since the developers and users are teammates, the communication between them will already be open. Development progress can be shared at staff meetings with those users not currently involved in the project. Since each compatibility team will have at least one participant on the project team, this progress report will include the overall picture to reinforce that this is a team project. The managers from the groups meet once a month, at which time they can be updated and show their support for the project.

On a project this size, management will require a schedule and goals. They will expect that individual progress be reflected in the individuals monthly status reports. They will commit, before the project starts, to an understanding that a project of this type is hard to schedule and will not pressure the team to meet strict commitment dates. There is one exception: A pilot demonstration will be required. This will be for senior management who is financing the project. Once this is scheduled and a commitment date given, it shouldn't be allowed to slip by more than a few weeks.

The pilot demo can be treated as a prototype. A project of this type should start small; by developing a subset of the final project to test the design, and testing for functionality of the design with a few samples. The user interface can be fine-tuned later. This demo will probably be about 3/4 through the first phase of the project. The knowledge acquisition phase of this project will not be as intense as some projects studied, so this project is divided into two phases. The first is requirement analysis, conceptual design and demonstrable subset of detailed programming design, and user interface. The second phase is to gather knowledge and complete the programming and user interface.

System Maintenance Phase

Motivation

Maintenance will be very expensive. All software projects require bug fixes, updates, and enhancements. This one will be no any different. The recommendation will be for a minimum of three full-time people to maintain this system. There are several ways to attract personnel to the task. Two of the people can be entry level engineers who have joined the company out of college or recently been promoted from technician status. The third person will be considered the leader. This person will be from the original project team who already knows the system. This person could be a senior level engineer, who needs to acquire leadership skills. This job should be presented as a stepping stone to higher grades, which leaves an opening for others to get the experience. It is very difficult for software engineers, working in a hardware specific company, to find a chance to be involved in a project like this. Therefore, motivation to work on maintenance of the system should not be a problem.

Summary

The purpose of this project was to explore the feasibility of an Expert System to solve our information sharing problem. The approach taken was to follow the outline of this report and answer each subject. The conclusion was, yes an Expert System will accomplish our needs. Several options were available and development of a Knowledge Sharing System is the right design for our Expert System. Only the basics were covered concerning how the knowledge is represented. Work Packages seems the right method for data representation, but much work needs to be done on the actual design. The next step will be to fully understand and design how the knowledge will be represented.