MANAGING EXPERT SYSTEMS:
A Framework and Case Study

Rob R. Weitz

Department of Information, Operations, and Management Sciences
Leonard N. Stern School of Business, New York University
44 West 4th Street, New York, NY 10012

IS-90-06

Abstract

This paper addresses the problem of managing the development and implementation of a large expert system in an organization. A traditional systems analysis and design methodology is used as a framework to highlight similarities and differences in managing large scale traditional computer based projects and large expert systems. As a non-technical, prescriptive guide, this article focusses on defining at each stage in the project, the tasks to be accomplished, resources required, impact on the organization, likely benefits and potential problems. The case of a large expert system implemented by a multi-national corporation across several European sites is used to clarify and expand upon the management guidelines provided.

1. Introduction

Research in the field of Artificial Intelligence (AI) signals great promise for the next generation of hardware and software. At the present time, expert systems are arguably the most commercially successful product of Artificial Intelligence research; they have crossed the threshold of the laboratory and are beginning to make their presence felt in real world applications. While others have described applications, suggested opportunities for the technology, or emphasized organizational considerations, (Leonard-Barton and Sviokla 1988, Luconi et al. 1986, Leonard-Barton 1987), to date the management of their introduction into the workplace and their impact once there still remain largely unexplored. This paper presents a prescriptive guide for managing large scale expert systems from inception through implementation and maintenance. It is motivated by experience with the development of a sizable expert system by a large, multinational company, the growing literature of cases describing expert systems in practice, and the belief that management and organizational considerations (as opposed to technical wizardry) must remain paramount for the success of such systems to be achieved. The thrust of this paper is to focus awareness on 1) the processes and resources required for an expert system project, 2) the costs and benefits of such an undertaking, and 3) organizational and task changes likely to result from the introduction of the system. This paper is not a technical guide for building expert systems; technical concerns are expressed here only insofar as they are inextricably linked to the management of such systems.

A great deal of experience has been gathered regarding the design, implementation and maintenance of "traditional" computer based systems, from both technical and management viewpoints. The pitfalls, players and positive practices have been identified in a large body of existing research, and are well described in the information systems literature. (See Senn 1984, and Burch and Grudnitski 1986, for example.) The management of expert systems does not lie completely independent from this previous computer based project management experience. Indeed, one trend is towards "invisible" expert systems - that is, expert systems which are integrated into conventional data processing hardware and software (Kozlov 1988). This paper builds on the existing groundwork by taking as a framework a traditional systems analysis and design (SA&D) methodology and adapting it for application to expert systems.

2. Expert Systems

Expert systems (ES) are computer programs for solving difficult, "fuzzy" problems in domains where human expertise is normally associated with a great deal of training and experience. Application areas to date include such areas as fault diagnosis, tax planning, credit evaluation, geological prospecting, chemical analysis and medical diagnosis. (See Kupfer 1987 and Kneale 1986 for popular-press descriptions of expert systems operating in business environments; Waterman 1986 provides an extensive overview of ES applications.) Expert systems are typically characterized by:

The utilization of large amounts of domain specific knowledge.

The ability to use incomplete or uncertain information.

The capacity to explain their behavior (a kind of self-knowledge).

Symbol manipulation, that is "reasoning" about objects, as opposed to numerical manipulation (which typifies traditional computer programs).

Performance levels at, or exceeding, those of experts in the problem domain.

Typically, sizeable expert systems are built in an iterative, incremental fashion via repeated interviews between one or more domain experts and a "knowledge engineer". Briefly, the task of the knowledge engineer is to elicit the expert knowledge, map the knowledge into a suitable structure and actually code the knowledge using appropriate software and hardware. The process tends to be tedious and time consuming, and has in fact been referred to as the "knowledge engineering bottleneck". A good overview of experts systems in general, and of the building process is provided by Hayes-Roth et al. (1983) and Harmon and King (1985).

For ES the importance of early development of working prototypes is stressed. The prototype is incrementally improved and its capabilities expanded by repeated trials with the domain expert and actual use in a test environment. In fact, several versions of the prototype are typically successively developed, until a sufficiently evolved version is realized for possible release.

Building and implementing a large expert system is both time consuming and resource intensive. While improved software environments have helped speed development, and recent experience has suggested some guidelines for easing the overall process somewhat, existing verifiable ES applications suggest that, for large systems, the time required to go from prototype to implementation is typically on the order of person-years, with costs measured in at least the tens of thousands of dollars. For a case in point, see Linden (1982). Clearly, larger systems (as measured by the amount of knowledge acquired and by the number of users) require more effort than smaller ones. In any case, it seems that under the appropriate circumstances the value of automated expertise supports the magnitude of these efforts.

3. A Traditional Systems Analysis and Design Framework

Creating a large computer based system for more than personal use is a complex task requiring technical skills, creativity and good management of resources. A guide for this process has been established through experience and while details vary from source to source, the overall thrust is fairly standardized in the information systems literature. One such systems analysis and design framework is provided in Figure 1, and is due to Lucas (1982). As indicated previously, this paper adapts the traditional SA&D procedures for use with expert systems.

Figure 1. about here.

The next section traces this outline as it applies to expert systems; fundamental variations of the traditional SA&D process are noted, and described or referenced. The resultant SA&D process for expert systems is summarized in Figure 2.

Figure 2. about here.

After detailing the SA&D framework for expert systems, a "real world" example is presented. It should be stressed that the case description serves to illustrate the framework and not define it. The Comet expert system was developed without benefit of the normative model outlined here - in fact the project was in large part a motivating influence in development of the framework. It should be clear that the case description not only clarifies the model, but also serves to demonstrate the benefits of following the proposed SA&D framework, and the disadvantages of deviating from it.

4. Framework

4.1 Inception

Inception refers to the realization that a computer based technology or application can be of value to an organization. The idea may refer to investigating a technology in an exploratory sense, or more specifically applying a technology to an existing or proposed procedure within the organization. At this stage the envisioned system is naturally somewhat murky in its details, though the overall goals - to reduce costs, ease a production bottleneck, maintain a competitive position, or better manage a process, for example, are more clear. Several variations of possible systems are likely entertained at this point. The question proposed here is, why should the envisioned system (or one of the envisioned systems) be an expert system?

It should be kept in mind that expert systems are not cheaper or easier to build than conventional systems; it is more realistic to assume the contrary. Therefore, there should be strong, positive reasons for proposing an expert system solution. Selecting the right tool (or technology) is always a question of matching the characteristics of the problem with the capabilities of the tool. Criteria for tasks well suited for expert systems have been enumerated elsewhere (Bobrow et al. 1986, Dym 1987 for example). These criteria are briefly summarized below.

Expert systems are used to replace or assist experts. (A rough definition of an expert is someone who can solve difficult problems more quickly and with less effort than a novice, or "average" person. Typically, expertise is acquired through lengthy training and experience, and is limited for an individual to a particular field of knowledge.) Insofar as technical criteria are concerned, for applying an expert system to a problem, the task should clearly

> Have semi-structured solution processes. Expert systems were created in an attempt to model problems that do not have algorithmic or "step-by-step" solutions. (Traditional, mathematical models or decision tree programs are appropriate for these type of problems.) Expert systems are well suited for tasks where a body of loosely structured knowledge exists; this technology was designed for capturing the heuristics or "rules of thumb" for arriving at good solutions.
>
> Involve a well circumscribed, limited field of knowledge. Expert systems are not general problem solvers. (As a case in point, each ES application in medicine focusses on one specialty; for example, MYCIN's domain is bacterial infections, while INTERNIST covers the domain of internal medicine.) Moreover, problems requiring "common sense" are not well suited for an ES solution.
>
> Be difficult, but not too difficult. As a general guideline the task should take a human expert somewhere between minutes and hours.

Given the above necessary conditions, an expert system is more strongly recommended for tasks that

Require explanation to the user of the system's reasoning in solving the problem.

Involve reasoning with uncertain or incomplete information.

Entail reasoning about symbols, or "objects", as opposed to mathematical manipulations.

The above criteria are purely technical requirements, and define the minimum considerations for further pursuing the process of thinking about applying expert system technology. Related to the above technical concerns are more organizational requirements for an expert system to be envisioned as a solution. These include that

Human experts exist and are willing and able to participate in the project.

Human expertise is scarce and/or expensive.

Management is likely to commit the required resources to the project. (This implies that the task is viewed as an important one.)

At the inception stage, some indication that these non-technical concerns are satisfied must be sensed, or at least their negation must not be evident.

For both traditional and expert systems, the inception stage produces a proposal outlining the reasons for the project, its goals, expected benefits, possible risks, alternative approaches, time frame, and required resources. The entire inception process serves to better define the project, and to garner organizational support. The inception phase ends with a decision to either stop consideration of the project, or to continue further investigation and evaluation.

In traditional SA&D, the next step is a full-fledged feasibility study. (The prototype proposal may in fact be viewed as a "first draft" feasiblity study.) For expert systems, successful completion of the inception stage means approving the development of a demonstration prototype, and concurrently undertaking the feasibility study.

Typically, the inception report includes an estimate of the resources which will be required, and an approximate schedule for the entire project. However, since approval at the end of the inception phase means commitment to the project solely through the feasibility study (for traditional SA&D) or through prototyping and feasibility study (for expert SA&D), the inception report should specify in detail a timetable and the resources required for these next steps. A decision to go ahead with the full system will be made when the feasibility study, or in the case of expert SA&D, the demonstration prototype and feasiblity study are ready.

Finally, the inception report for an ES may include evaluation criteria for the prototype. While criteria may be difficult to specify, what constitutes "success" for the prototype should be defined in advance. (Determining evaluation criteria is discussed in the next section.)

For many organizations, a process for gaining a basic understanding of what ES technology has to offer, in general or with respect to the application domain, is required in order to compile the inception proposal. This may entail formal training, attendance at seminars, transfer within the organization of individuals experienced with ES technology, contact with ES software firms, or the hiring of consultants. The objective is to acquire the capability of fully assessing the potential project, and to begin to coalesce the resources for its undertaking.

4.2. Feasibility Study

As with traditional SA&D, the feasibility study is comprised of a comparative evaluation of all conceivable alternative systems. (At minimum a single new system is proposed and evaluated with respect to the existing procedures.) The ultimate purpose of the study is to select the best of all possible alternatives, but this part of the analysis serves many other purposes. Among these, it helps solidify ideas, provides a common source of reference, and serves as a focus for gaining commitment from users and management. The feasibility study is a fuller, more detailed version of the inception report. It should be ready for release when the demonstration prototype is ready, or shortly thereafter.

The contents of a traditional SA&D study report are outlined below. Following each directive is a description of how that directive applies to an expert system feasibility study. The report should:

Describe the current system including a rationale for changing it.

> As the saying goes, "If it ain't broke, don't fix it!" Shortcomings in the current system and improvements provided by the suggested system should be clearly specified. Expert system proposals based on rationales of selecting a "high tech, state-of-the-art solution" should be rejected.

Explain why the particular solution is proposed, as opposed to some other alternative; already existing, similar systems both inside and outside the organization should be referenced if possible.

> Why is an expert system appropriate? The previous section describes problems and environments which suggest an expert system solution.

Lay out the goals, scope and objectives of the envisioned system.

> It is important to set realistic expectations for the proposed system. Terms such as "artificial intelligence" and "expert system" tend to spur the imagination. Bear in mind that this technology typically cannot do what a human expert cannot do and that, in any case, the breadth of knowledge encompassed by the system will be limited. Specify the bounds of the proposed system; i.e., what are <u>not</u> the goals, objectives and expected capabilities of the system.

Describe what the proposed system will do. Include in this description how the system will work, who will use it, how the task itself and related tasks will change.

> The following questions should be addressed. Will "experts" themselves be using the system (i.e., doctors using a medical diagnosis system) or will "technicians" use the system? What skills will be required of the users? Will the users be new employees or current workers already doing the task under the present system? If current workers, how will their input into system design, and their goodwill in general be solicited? Will the system "de-skill" their task, and if so how do they feel about it? How will job responsibilities and lines of communication change? The overall technical specifications should be outlined; i.e., how the system will interface with existing information systems, data needs, input/output mechanisms, performance requirements, etc.

Specify a timetable for the project, including periodic reviews and expected performance of the system at each review.

As mentioned above, expert systems are developed incrementally. The first review should occur at the unveiling of the demonstration prototype system. (This review should take place two to eight months from the start of the development work.) Feedback from this review should direct further technical work on the system, while any organizational difficulties which have surfaced should be aired and attended to. Attendees at the meeting should include those involved with development, representatives of the ultimate users of the system, and those responsible for the direction of the project. At minimum, one more review will be required (on the order of 6 months after the first meeting) to decide to either test release, delay or cancel the project. Typically another review is held after test release, and just prior to full release.

Detail the resources required, when they will be required and from whom they will be required.

For the proposed project, allocations should be included for knowledge engineers, domain experts, programmers, hardware, software, training, and travel expenses. Note that the participation of domain experts is crucial, while their time is typically at a premium. (Hence the value of automating their expertise.) Provisions for re-specifying their job description to include work on the new system should be detailed. Commonly, programmers are required to write user, and system to system interfaces. A fuller description of all the players required for such a project and how they differ from those involved in developing a traditional system is given at the end of this section. As with traditional systems, the choice of hardware and software is one of making trade-offs. Indeed, much of the criteria (price, speed, compatibility, reputation of supplier, for example) are the same. Hardware selections fall into three categories: microcomputers, minis/mainframes, and workstations. Special purpose expert system development software (typically called "shells" or "environments") exist for each category of hardware and ease the programming task considerably. It is typically the knowledge engineer's responsibility to choose appropriate hardware and software for the project. (Gilmore and Howard 1986 and Mettrey 1987 discuss ES software selection. For details on ES hardware and software, and more on the specifics of the knowledge engineering process see Harmon and King 1985, Holsapple and Whinston 1987, Waterman 1986 and Hayes-Roth et al., 1983.) The resources required to achieve the demonstration prototype should be made explicit, as typically after the initial approval, the next "go - no go" decision point is at the prototype review.

Provide a cost-benefit analysis. This analysis should include both tangibles and intangibles; typically some estimates are required, but the process should be supported with a sensitivity analysis.

Benefits commonly ascribed to expert systems include: preservation and dissemination of scarce expertise, relieving experts of tedious tasks and thereby allowing them to concentrate on more difficult/more interesting problems, speedier solutions and more consistent problem solving. Huber (1984) identifies knowledge as a key, strategic resource in the "post-industrial" organization. Important costs include: personnel (i.e., knowledge engineer and expert), software and hardware (perhaps specifically for expert system development and use), and those expenses associated with training, operations and updating.

Define, as best as possible, evaluation criteria and agree on how the success of the system will be measured. The evaluation criteria should be utilized during the periodic reviews.

Expert systems will typically be evaluated on a host of criteria. These include the quality of the solution, the speed of solution, the manner in which the solution is reached (transparency), breadth of knowledge, explanation and help facilities, user satisfaction, and the ease with which the system can be transported, modified and updated. The relative importance placed upon each of these criteria is determined by the project's goals and objectives. The criteria should be established by the ultimate users of the system and the managers of the task in question, in conjunction with the developers of the system.

Assign ownership of the system over the course of its lifetime.

In many applications, updating of the knowledge encompassed by the system is a large, ongoing job due to the nature of the task in question. Consider, for example, a diagnosis expert system for some large piece of machinery. Frequent updates will be required if new models of the machine are produced, parts (or part numbers) change, and new faults (and procedures for finding them) appear as the machine ages. Responsibility after release of the system may or may not rest with the developers, but in any case this responsibility and the mechanisms for undertaking the maintenance task should be made explicit.

Many of these points will be elaborated upon in the subsequent sections.

4.2.1 Personnel

Some comments about the players involved in expert systems development and how they compare with those for traditional systems are in order. In traditional systems, three major players are identified: users, the information systems (IS) department, and management. Much has been written of the responsibilities of each of these agents over the course of the life of the information system project (see Lucas 1982, for example). Briefly stated, ideally the user participates in the design process and may in fact originate the project idea. The user is best equiped to understand the workings of the present system and therefore to provide input for system specifications, and good test examples. Clearly user satisfaction with the system is an important criterion for success. The IS department reviews the feasibility study, designs the system, specifies possible alternatives and the trade-offs they imply (languages, "off-the-shelf" or special purpose programs, use of service bureaus, batch versus time sharing mode, hardware, etc.), handles the required programming, documentation, training, conversion and maintenance. Management responsibilities are overall approval and direction of the project, and providing commitment in terms of resources and recognition.

Expert systems projects include additional players. First, the domain expert is not necessarily a typical user of the system. If the expert is in fact just an "average user", then his/her involvement in the ES development will, in any case, be much more intimate than that required of a user in traditional systems. Second, the position of knowledge engineer requires skills that are not necessarily found in IS departments. This role may be filled by grooming in-house personnel, or through external services. These services may be obtained via independent consultants, and expert system software or hardware companies.

In traditional system development, the IS department has the responsibility for choosing outside vendors and services. The situation is analagous here, with many of the trade-offs (such as reputation of the company, cost, type of service provided) being similar. However, the IS department must be knowledgeable about ES technology and the market in order to evalulate these trade-offs. Further, the IS department will have to work with the external organization to promote a smooth interface between existing systems and procedures and the new system. If an outside organization is contracted for developing the system, arrangements for transfering ownership to the IS department (and what that ownership entails) should be specified.

4.3. Prototype

The use of prototypes has been referred to several times. Prototype in this sense means a scaled-down (in scope, power or both) version of the fully envisioned system. While a working prototype phase may exist in the development of traditional information systems (Alavi 1984, Janson 1986), it is not an inescapable part of the standard system building methodology; further, for traditional systems prototyping may refer simply to creating screen "mock ups" in order to improve the user interface.

The formal prototype stage in ES development is suggested because 1) it may be hard to know what is feasible when it comes to automating expertise without attempting a working, trial version, 2) given the newness of the technology (and the hype that surrounds it) a prototype can help set realistic expectations, 3) a demonstration can serve to garner and solidify enthusiasm for the project, and 4) a phased approach involves less risk. Finally, due to the incremental nature of building expert systems and the software development tools created to support this development, prototypes may be built rather quickly.

4.4. Analysis, Design, Specifications, and Programming

In developing traditional information systems, each of the steps of analysis, design/specification and programming is ideally separate and distinct; one does not proceed to a subsequent step until finished with the previous one. In the analysis phase, the current system is studied; transactions, data volumes, decisions made, etc., are detailed. Design/specification of the new system involves enumerating hardware and software, input and output files, media, procedures for use, security and error control. Writing and testing the program follows the design phase.

For expert systems, some analysis, design and detailing of specifications is usually required for the inception phase, and a large part of these tasks plus some programming is completed as part of the prototype system and feasibility report. These processes however are far less differentiated in ES development than for traditional systems. In fact, these processes more closely resembles a paradigm for decision support system development. (See Sprague and Carlson, 1982, and Keen and Scott Morton, 1978 for example.)

The incremental, iterative procedure for building ES has already been described. This is a function of the fact that experts simply cannot sit down and fully and completely specify their own problem solving behavior. Expert systems are constructed as a collaborative and iterative effort between one or more knowledge engineers and one or more domain experts. The knowledge engineer is experienced in eliciting knowledge from an expert, structuring knowledge such that it can be programmed, and then coding the knowledge. Working from the first prototype, the knowledge engineer can improve, refine and expand the capabilities of the system through further interaction with the expert. This loop: eliciting, structuring and coding knowledge is traversed many times before a suitable version for release is produced. Here, programming may be considered to be a function of the analysis, design and specification of the system and visa versa.

At the beginning of the development of the system, infrequent contacts between the knowledge engineer and the expert will suffice; as development proceeds, more frequent and more lengthy meetings will be required. In any case, it is crucial that the expert has sufficient time and incentive to work with the knowledge engineer on the system.

4.5. Testing

For both traditional and expert SA&D, the feasibility study should specify a timetable for release of the system, first as one (or more) trial or test versions, and later as the "product" version. The test version receives limited distribution and is used to fine tune the system for general release. For each release, the system should pass the minimum requirements set up in advance in the feasibility report.

Disagreement and uncertainty concerning the evaluation procedure is likely to exist. Developers may be most concerned with technical criteria, and users with quality interfaces, while managers worry about how the system will help solve "business" problems. (For ES even a purely technical evaluation based on the quality of solutions provided by the system may be difficult to conduct. For example, experts themselves may have differing opinions as to what constitutes a "right" answer.) While disagreement may exist as to how to evaluate the system, the time to resolve such differences is prior to development. (See Hayes-Roth et al. 1983, for a good discussion of evaluating expert systems.) In any event, expert systems must be evaluated under multiple criteria. Possible evaluation criteria include whether use of the system:

Results in better solutions.

Results in faster solutions.

Results in more consistent solutions.

Provides greater worker satisfaction.

Is easy to use.

Reduces training time (for the human users).

Reduces dependence on scarce individuals.

Has resulted in greater insight into the problem.

Reduces the number of extremely bad solutions.

These criteria, should be easy to derive from the goals of the system as stated early in the project. The more difficult part is measuring how the system fulfills these criteria. While some data collection procedures may already exist (time or quality standards for diagnosis tasks, as an example), data collection may have to be initiated as part of the project to allow for a "before and after" study. While the level at which the system satisfies each criterion may be flexible, other requirements may be both easy to determine and rather fixed, for example:

Time to repair must be reduced by half.

Relatively novice technicians must be able to do the task (with the aid of the system) as opposed to the experts who do it now.

Calls to the resident expert for help on difficult problems must be all but eliminated.

The number of people doing the task should be reduced by 10%.

During testing, the system should be pushed to discover its limits, in terms of the range of its knowledge, but also on very traditional dimensions: security, back-up procedures, and error handling, for example.

User and/or management satisfaction with the system can be assessed via formal questionaires or interviews. Acceptance will be predicated on not just how well the system does what may be narrowly defined as its task, but also on ease of use, appropriate help facilities, speed, and on how well it supports the user in doing the task as (s)he sees it. A more fundamental question regards not the system per se, but rather the individuals for whom the system is designed. Knowledge workers, may or may not want a machine looking over their shoulders while they do their jobs, particularly if they view the system as skill or prestige reducing. (Doctors are the classic case in point.) This possibility should be addressed early in system development, so that disaffection due to resentment does not show up among users at the evaluation stage.

As with traditional systems, both tangible and intangible outcomes will likely need to be measured; release of the system is, of course, contingent on the benefits outweighing the costs.

4.6. Training, Conversion/Installation

Training, conversion and installation are similar to that for traditional systems. Appropriate documentation should be developed by the knowledge engineer in conjunction with the IS department if necessary. Training procedures are set up in cooperation with the users and management. Hopefully, conversion may be phased in, with the date for conversion agreed upon by all those involved. As with traditional systems, the question of compatibility of hardware and software is important. With expert systems, compatibility may focus on the use of AI languages or specialized workstations which need to interface with more conventional hardware and software. Methods for accessing company data, and generally interfacing with existing systems are frequently a necessity. These issues should have been planned for in the design phase.

Problems can occur involving the conversion and installation phase of ES if development of the system was performed by personel outside the organization's MIS function. This outside development may happen, given that the skills and training required to develop an ES are rather specialized. Cooperation of the MIS group is essential for installing the ES as part of the organization's overall network; their "buy in" should be managed from the start of the project.

4.7. Operations

The operations phase for traditional SA&D involves fixing errors, and when necessary making changes to the system.

For both traditional information systems and expert systems, assuming an outside agent has been involved in the development phase, the arrangement should include provisions for ongoing maintenance of the system, in the traditional sense (i.e., information "hot-line", software revisions, etc.). Maintenance duties which are the responsibility of the IS department should be clarified.

For ES particular attention must be paid to the maintenance task as these systems work in knowledge intensive areas, and problem solving knowledge in practical applications frequently changes. Updating may be required for knowledge proper, for data the system uses, or both. Some assessment of the extent of updating should be made early in the project, based on the nature of the task. For the XCON system, Digital Equipment Corporation's ES for configuring their Vax family of computers, some eight individuals are employed full time on the updating, that is, collecting and encoding task. This huge effort is due, at least in part, to the fact that the system must constantly be modified to work on new products (McDermott

1984). An estimate of the frequency of knowledge updating and the assignment of updating responsibilities should be part of the project report.

New knowledge (and alterations to the system independent of the knowledge base) will likely be suggested by users of the system, particularly if they are experts. Some formal mechanism should be established to capture these suggestions, and system performance in general. Unfortunately the knowledge encapsulated in expert systems cannot be updated simply by adding knowledge. (Automated updating, that is, updating performed entirely by users suggesting new knowledge to the system, is an important area of research in AI. Of course, systems which learn from their own mistakes, another research area, would solve this problem.) Aside from having to put the new knowledge into a form the system can understand, the new knowledge must be tested for its effects on the existing knowledge. This is to say that individuals familiar with the details of structure and operation of the system must be involved in the maintenance task, in addition to domain experts. If the system was developed by knowledge engineers outside the organization, updating will have to be performed with the assistance of these knowledge engineers. Alternatively, as part of the project sufficient expertise must be brought in-house (i.e., to the IS department). Again, depending on the nature of the task, its projected future requirements, and the data involved, maintenance may be a considerable job.

Finally, at some point after the system has been in regular operation, some thought should be given to how well the system performs. Are the projected cost savings, or productivity improvements being realized? Does the system satisfy current and projected demands? How well does the project contribute to the overall strategy of the organization?

5. Application

5.1 Inception

Digital Equipment Corporation, one of world's largest computer manufacturers, is perhaps best known for its Vax series of minicomputers.

An internal report describing expert system technology was circulated within Digital's manufacturing/repair engineering group in Nijmegen, Holland in early 1985. The report defined the technology and its capabilities, and suggested a range of possible application areas within the group's purview. An ES approach for these applications was supported by the following rationales:

> Knowledge, currently, in and of itself is a vital commodity for high technology companies. Expert systems are a means of managing knowledge.
>
> In particular, the knowledge required for the manufacture and repair of computer hardware will likely increase in scope and complexity as the products themselves become more complex and product life cycles shorten.
>
> A forecasted high demand and increasing cost for human experts.
>
> A long term need to increase productivity and to reduce costs.
>
> A competitive incentive to maintain and develop expertise in the area of AI.

The document recommends one application in particular, an ES for assisting in the task of hardware module ("board") diagnosis and repair. This domain was suggested firstly on economic grounds, and secondly on the rationale that relatively well defined, diagnosis type problems are typically well handled by ES technology. A pilot project is called for in order to both more fully explore the application area, and to provide support towards a long term strategy. The creation of a project team is stipulated, composed of a knowledge engineer, domain expert and local process manager, who would receive overall direction from a management steering committee. The report appointed individuals to the steering committee. Internal consulting support (from Digital's AI group) and external consulting support (from local university and government projects) were suggested. Finally, the required funding, and a timetable for forming the project team, evaluating resources, and developing the pilot (prototype) system(s) was delineated.

Upon approval of the report, the two individuals whose responsibilities were to include the knowledge engineering function were sent to internal training courses covering ES technology, and its application within Digital. This education phase was necessary in order for the individuals, who work in the manufacturing function and whose professional training is in engineering, to 1) become fully acquainted with the opportunities and methodologies associated with ES, 2) provide expertise in the selection of an appropriate application domain, 3) contribute to the feasibility study and 4) ultimately, act as knowledge engineers during development. Lastly, at this point, liason with Digital's internal AI consulting group was established for maintaining ongoing assistance.

The timetable provided by the project report specified approximately six months for the process of education of the knowledge engineers, further specification of the domain application, and development of a demonstration prototype. The full feasibility study would be required within a month of the prototype.

5.2 Feasibility Study

The feasibility study (or "Project Plan" as it is called within Digital), was prepared by the manager of the strategic engineering function at the Nijmegen facility, the two knowledge engineers assigned to the project, and a representative of Digital's internal AI consulting group. The authors did not include a domain expert.

As all computer vendors, Digital is concerned with assuring reliability of its products and, in the case of hardware failure, minimizing customer downtime. When a hardware fault occurs at a client site in Europe or the Middle East, the offending board (or boards) are isolated and removed from the computer installation by a field technician who then replaces them with functioning equipment. The faulty boards are then sent to one of 17 field service sites located throughout Europe, or to the one in Israel. At the field sites, the boards are either repaired, scrapped, or sent to Digital's central repair facility (CRF) in Nijmegen, Holland for more extensive testing. The CRF has more sophisticated test equipment and more experienced personnel than the field service sites.

Major costs are incurred in this process due to the inventory of boards required to support such a system, shipping fees, expensive test equipment and the training of repair personnel. Training costs are particularly important as diagnosing procedures may be different for different boards, and hardware is constantly changing. Additional costs arise when technicians replace sets of components on a board unnecessarily. This occurs when a problem area is isolated, but determining exactly the component responsible is especially difficult and time consuming.

There are currently over 500 different boards, each one containing some 150 - 250 electronic components. Repair times range from 30 minutes to several hours, depending on the board itself, the equipment available at the repair site, and the expertise of the technician. Test equipment varies from simple voltage and current meters, to purpose-built devices which automatically run a series of tests on the suspect module, to full-fledged, high-end computers. Many thousands of modules pass through the system per month, each module typically costing in the hundreds of dollars. Digital's repair function can be considered quite a sizeable business in its own right.

The system will be available at each repair site, in the form of terminals at the workbench of the repair technician. Through interactive dialogue with the technician, the system will direct the repair process by suggesting appropriate test procedures and interpreting the results of the tests.

An ES solution is proposed because there is a good match between the characteristics of the problem and the capabilities of expert systems. From a technical point of view, the problem is difficult but limited in breadth, experts exist, and the task involves symbolic, not mathematical manipulation. Moreover, this is a standard diagnosis problem and as such tends to be well suited for an ES application. From a practical/business point of view, experts are available, and the possible tangible benefits are sizable. Further, Digital, as a major computer company and one involved in AI research and products, has a practical interest in learning about this technology through its own experience.

According to the report, the Knowledge-Based Board Diagnosis System (KBBDS) will reduce costs by increasing local repair, thereby minimizing "in-pipeline inventory", speeding repair, distributing scarce expertise, lowering test equipment requirements, cutting training time, and decreasing component usage. Further, the knowledge and records developed through the system may provide fault information useful for the design of future hardware products. Diagnosing all conceivable faults or incorporating all possible repair information are <u>not</u> goals of the project.

Digital's motives in this project are, firstly, to implement this particular expert system. However, their goals extend beyond this immediate task towards two others: 1) the development of a mechanism for evaluating the feasibility of this technology in other applications, and 2) the creation of a set of generic software modules that can serve as a basis for future expert systems in the manufacturing/repair environment.

The feasibility study begins with a description of the current system and a discussion of the opportunities for improvement. A single alternative, the expert system is proposed; therefore the analysis compares current procedures to the proposed one.

5.2.1 Benefits

Cost reduction in inventory was estimated via a simple model which estimates the savings in boards "in the pipeline" expected via use of the KBBDS. The following input to the model is required; input estimates were varied in order to provide a sensitivity analysis.

The total volume of boards in the system.

The fraction of boards sent for repair to the central repair facility.

The carrying cost of inventory.

The "turn around time" of boards sent to the central facility.

The "turn around time" of boards in local repair.

The cost of a module.

Additional expected benefits were approximated by providing estimates of the impact of the system on each of the following categories.

Decrease in average time to diagnose.

Reduction in learning curve cycle.

Lowering of depreciation costs.

Reduction in component usage.

Actual percentages were proposed for each category, based upon judgments of the project team. No dollar value was provided for these savings, though such a transformation could be made.

5.2.2 Costs

Costs for the system are broken down into three categories: hardware, software and personnel.

The system will be developed on a micro-Vax (a stand-alone Vax workstation), and will run on any Vax under the VMS operating system. At the repair workbenches, access to the system is to be assured through hardware and software connections to Vax's already on site, or if a Vax is not already available on site, use of the system will require the purchase of a micro-Vax. Software required includes the OPS-5 shell, the knowledge base, the knowledge updating mechanism, and the interface code, all of which is considered part of the KBBDS. (See the next sections on prototype development, and systems analysis and design for more on hardware and software.)

Personnel requirements include the project manager, two knowledge engineers, internal consultants, domain experts, and users. While specific individuals were identified and assigned the task of manager, knowledge engineer and consultant, only the eventual necessity of recruiting articulate, qualified, and willing domain experts and users was recognized.

A quarterly budget extending over the length of the project (a period of almost two years) included costs for personnel, travel, training, hardware and software.

5.2.3 Project Planning and Specifications

The following time frame for the project was suggested.

I: Quarters 1 and 2.

Select a computer and several of its boards on which to focus. The application should be directed towards a stable, well known product. The Vax 11/750 (Comet) was suggested.

II: Quarters 3 and 4

Develop prototypes. The first demonstration prototype is to be tested at Nijmegen. A second prototype is to be field tested at a local repair facility. Evaluations of each prototype are to be conducted. Revision and improvement of the system should be expected at each evaluation. The system is to be distributed to those local facilities which request copies.

III. Quarters 3, 4, 5 and 6

Select and develop an ES for a new and lesser known project.

IV. Quarters 7 and 8

Create additional module repair systems for additional products. Develop, based on experience gained thus far, procedural guidelines and generic software tools for further use in the repair process development environment.

The project plan clarifies the limits of the project: the system should not be expected to diagnose all conceivable faults, or to incorporate all possible information concerning board repair.

System specifications include: 1) capability to aid in the diagnosis of 30% - 50% of the faults, 2) a response time varying from 10-40 seconds, 3) a context sensitive "help" function, and 4) a "history file" option for recording the diagnosis procedures used by the technicians for problems not solvable via use of the KBBDS. Detailed criteria by which the system would be evaluated were not cited in the report.

Technicians currently doing the repair task will still do the task under the new system; that is, no change in the individuals doing the task is expected. The task itself will change in the sense that a new piece of test equipment (the ES) will be added to the tools available to the technician. It is expected that fewer boards will have to be sent to central repair locations, less boards will be scrapped, fewer consultations with other technicians (on difficult problems) will be required, and the learning curve for repair will shorten. No special training is expected to be needed, other than minimal on the job training.

Some concern was expressed about the KBBDS acting to de-skill the task of board diagnosis, and a concommitant resentment of the system on the part of the technicians. Solutions to this possible eventuality were to a) design the system to avoid de-skilling, b) encourage technicians to move on to more complicated products, and c) encourage technicians to take on updating and expanding the KBBDS as part of their jobs.

5.3 Prototype

A number of expert system shells were evaluated with the assistance of Digital's internal AI group. The first prototype was developed with a commercially available PC-based product. This initial prototype contained knowledge about a subset of possible faults for a single Comet board. A decision was then made to move to the OPS-5 language on a Vax-based system. It was felt that this platform allowed for the greatest potential in terms of communication and networking within the Digital manufacturing and repair environment. Digital policy directs only the use of already "proven" in applications ES software. While ES within Digital have employed a variety of software tools, OPS-5 was ultimately chosen for its flexibility, and long history of use.

The prototype served to: 1) deepen the project team's understanding of the technical and organizational issues surrounding the board diagnosis problem, 2) gather technical support from the internal AI consulting group, 3) gather project support from the repair facilities, and 4) overall, demonstrate the technical feasibility of the application.

5.4 Analysis, Design, Specification, and Programming

Development of the system progressed in stages, from prototype to test-release version to full-release version. While the prototype contained limited knowledge about one board, the test release version could reason about faults on four. Knowledge was incrementally added to the system, broadening and deeping its capabilities through full release.

Aside from some commonly encountered problems (i.e., the experts had difficulty expressing their knowledge, the process was extremely time consuming), the project team acknowledged their frustration in accessing the experts. The team members emphasized the importance of gaining commitment from verifiable experts earlier in the project.

Each use of the system during test and full release automatically sent a "use report" electronically back to the project team at the Nijmegen facility. The report recorded the type of board under test, and a rating on a seven-point scale of how well the KBBDS was able to diagnose the fault.

(This scale, described in the next section was used for evaluation purposes.) The technician could also enter free form comments regarding performance. Thus, if the system could not find the fault, information was collected regarding the type of fault which occurred, and how the technician proceeded to diagnose the problem. This knowledge, promptly passed back to the project team in Nijmegen, could then be incorporated into the system by the knowledge engineers.

5.5 Testing

5.5.1 Test Release

Several months after the demonstration prototype was completed, an updated system with a more extensive knowledge base was ready for evaluation. A successful review at this stage would mean an official test release at one repair facility. (This facility was selected based on the willingness of the plant manager to participate in the test.) In order to allow for more complete evaluation, the system was installed at the proposed test site, in Evry, France, where local staff could somewhat informally examine it. This feedback could then be utilized in the review for test release.

Members of the test release review team included the Evry facility plant managers, a repair engineer from Evry who had had access to the system, a representative from Digital's internal AI group, and two representatives from the advanced manufacturing group who had experience with another ES project. The day long meeting's agenda began with an historical overview of the goals and objectives of the project, an outline of its current and planned status, and a demonstration, including "hands-on" usage of the system. Overall the project was on schedule, the designers having created a system containing knowledge regarding four Comet modules. In the process, they had gained significant understanding of the board diagnosis task, and solid experience in knowledge engineering. Problems regarding accessing domain experts and the task of gleaning expert knowledge were discussed.

After the presentation by the development team and an opportunity to try the system themselves, the review members broke into groups, spending an hour and a half discussing the project. The team then regrouped to discuss their conclusions with each other and the project team, and subsequently provide feedback to the project team concerning further development and implementation.

From both the response of the technicians who had tried the KBBDS on site, and from the demonstration at the review meeting, it was felt that the system was ready for field testing. While no technical redirection was suggested, several proposals (made by the Evry test site management) concerning organizational issues were entertained and ultimately incorporated into further development plans. The first of these concerned better orientation of the personnel at the test facilities concerning the nature, scope and requirements of the project. It was suggested that the technicians, engineers and managers at the repair sites needed to be better informed prior to release of the system as to how the system works, the hardware required, who will use it, and what if any, training would be required. It was agreed that a steering committee should be set up at the repair facilities, composed of project members, and facility managers and technicians; the objective of these steering groups would be to better manage the transition and maintenance of the KBBDS at each site. It was also suggested that several technicians be brought more closely into collaboration with the development team in their work towards enhancing the system. Further, the nature of a formal evaluation mechanism was debated.

The project team proposed that the system be evaluated during test release phase according to the following mechanism. Each use of the system could produce one of seven possible results, listed below. The project team had specified the maximum or minimum percentage of boards to fall in each category; these are listed in the right hand column.

0) No Response	< 30%
1) Totally Incorrect Response	< 5%
2) Unclear and Misleading Response	< 5%
3) Neutral Response	< 10%
4) Somewhat Helpful Response	> 15%
5) Very Helpful Response	> 20%
6) System Found Fault	> 15%

The review team, in particular the manager of the Evry test site, proposed alternative criteria by which the system should be measured for successful exit from the test release phase. As opposed to specifying minimum or maximum percentages for each category, it was suggested that at least 60% of the trials produce results greater than 3, with an average performance of at least 4.5. Moreover, it was pointed out that of major concern to management was that average time to repair (ATTR) be reduced; a goal for success would be a reduction in this measure by 33%. Finally, it was noted that no measure for operator satisfaction with the user interface had been established. It was proposed that feedback from the technicians regarding satisfaction with the user interface be obtained.

These proposals were approved by the review and project teams as suitable criteria for evaluating the system during test release, in anticipation of full release.

5.5.2 Full Release

The system was in test release for some three months. At the end of this period, an evaluation was held to determine if and how a full release should be undertaken. This evaluation concluded that:

Overall, the system performance generally met the technical standards required for full release. (Reliable ATTR data were not obtainable however; it was agreed that these data would be appropriately determined and made available in the near future.)

The system, both from the technicians' and manager's viewpoints, was extremely valuable as a learning tool; that is, as a means for novice technicians, or more experienced technicians unfamiliar with Comet boards, to "get up to speed" quickly and with minimum frustration.

Novice technicians working with the KBBDS could maintain output at about the same level as very experienced technicians.

Some concern was expressed by the manager of the repair facility concerning the time spent by his technicians in working with the knowledge engineers in updating the knowledge base. It was suggested that this be considered formally as an investment by Digital, as this time could not, under present corporate guidelines be counted "productive time" vis-a-vis the established repair metrics.

Finally, the experience during the test phase reinforced the belief that a careful preparation be made at each repair facility prior to introducing the KBBDS. Personnel at the facilities should be made aware of the capabilities, requirements, limitations, and benefits of the system before it comes on site.

5.5.3 Post-Release

Full, incremental release to the other test facilities followed. Six months from the start of this full diffusion of the system, a final evaluation was performed based both on the criteria established at the review for test release, and a cost savings analysis.

Performance data were collected for the start, mid-point, and end of the six month long, full release evaluation period. Continuous improvement vis-a-vis the seven point rating scale was observed, though the level and improvement of performance varied between the boards. With respect to the average time to repair metric, as compared to process standards defined corporate wide, the reduction in ATTR ranged from 20% to 6%, depending on the board. More impressive were the "real" ATTR improvements for novice technicians; using the system reduced the ATTR by approximately 50% for two of the boards.

Verbal feedback from both technicians and managers was very positive. Again, the big advantage of the system was perceived to be the sharp rise it induced in the learning curve, and the improved overall performance of novice technicians.

5.6 Training, Conversion/Installation

After successful evaluation of the test release system, the KBBDS was released to the other Comet repair facilities, one at a time. The project team in each case was responsible for presenting the system to the facility staff. Training required was minimal.

5.7 Operations

The automatic, electronic forwarding of diagnostic reports assured a mechanism for keeping track of needed updates. These could be incorporated into the system by the knowledge engineers in the Nijmegen facilities, and the updated knowledge bases tested. At regular intervals, the revised system can be transmitted to the local repair sites.

As the inception phase described, the Comet KBBDS was seen as an early stage of an overall knowledge engineering process at Digital. With the future development of additional ES in mind, a rough analysis was performed comparing the estimated cost savings of the Comet system, with the estimated costs of developing and maintaining a similar system.

Based on the Comet project experience, the estimated total costs for development and maintenance (i.e., knowledge acquisition, validation, hardware, etc.) of another similar system would amount to between $25,000 and $50,000. Estimated annual savings due to the Comet system, based solely on the reductions in ATTR (based on process standards) and in the learning curve, more than offset this cost. (These savings do not include those due to inventory or scrap reduction, nor those due to reductions of boards "in the pipeline".)

It should be noted however, that the managers of the repair process find that the greatest value of the system is in the increased flexibility it provides. In bringing novices "up to speed" quickly, and in general allowing less experienced technicians to perform more proficiently, the Comet ES has allowed for 1) peak work periods to be handled with relative ease, 2) decreased dependence on the constant availability of expert technicians and 3) greater freedom for the experts to work on the more difficult problems.

Overall, the system was seen to have achieved its original objectives. Since the implementation of the Comet ES, the system has been expanded to include diagnostic knowledge about modules on Micro-Vax 2000 workstations. Current discussions focus on future directions of expert system technology as part of the overall strategy in Digital's repair and manufacturing environment. Figure 3. presents an overview of milestones of the Comet expert system project.

Figure 3. about here

6. Summary

The task of managing the development and implementation of a large scale expert system is in many ways similar, but in substantive ways different than that for large, traditional computer based systems. This paper has served to highlight the similarities, describe the differences and provide a rationale for the contrasts. Figure 4 summarizes the differences in the SA&D process for the two systems.

Figure 4. about here

Among the differences with important managerial implications, are that an ES:

Captures and manipulates knowledge, as opposed to information.

Includes a working prototype phase.

Is developed in an incremental, iterative style.

Requires additional players; in particular, at least one knowledge engineer and one domain expert.

May involve non-traditional software and hardware.

Will likely involve significant revisions (updating) of the system once in operation.

Successful management of an ES project requires:

First, clearly, recognizing when an ES solution is appropriate to the problem at hand, and when it's not.

Understanding the likely financial resources required.

Identifying attainable goals for the system, and the associated benefits.

Specifying evaluation criteria at several phases of the project.

Setting an appropriate timetable.

Identifying an expert, and ensuring his participation.

Enlisting or training a knowledge engineer.

Caution concerning the use of ES hardware and software, vis a vis maintenance and interfacing with existing systems.

Understanding that conventional programming resources will no doubt be necessary.

Expecting knowledge engineering to be tedious, time consuming, iterative and incremental.

Managing expectations and skepticism.

Considering the organizational and/or task changes which are likely to result from implementation of the system.

Allotting resources and mechanisms for ongoing updating.

This paper has addressed in detail each of these issues, by describing the life cycle of an expert system. Each phase has been defined, and a prescriptive guide toward managing the resources and responsibilities required over the course of this life cycle was presented. An example of the development, implementation and maintenance of a large-scale, multi-site expert system served to illustrate the conceptual framework.

References

Alavi, M., "An Assessment of the Prototyping Approach to Information Systems Development", Communications of the ACM, vol. 27, no. 6, June 1984.

Bobrow, D.G., Mittal, S. and Stefik, M.J., "Expert Systems: Perils and Promise", Communications of the ACM, vol. 29, no. 9, 1986.

Burch, J. and Grudnitski, G., Information Systems: Theory and Practice (4th ed.), Wiley, 1986.

Dym, C.L., "Issues in the Design and Implementation of Expert Systems", Artificial Intelligence for Engineering Design, Analysis and Manufacturing (AI EDAM), vol. 1, no. 1, 1987.

Gilmore, J. and Howard, C., "Expert System Tool Evaluation", Proceeding of the Second Annual Conference on Expert Systems Tools and Applications, Avignon, France, 1986. (Authors' address: Artificial Intelligence Branch, Georgia Tech Research Institute, Atlanta, Georgia 30332, USA.)

Harmon, P. and King, D., Expert Systems: Artificial Intelligence in Business, John Wiley and Sons, 1985.

Hayes-Roth, F., Waterman, D. and Lenat, D., Building Expert Systems, Addison-Wesley, 1983.

Holsapple, C. and Whinston, A., Business Expert Systems, Irwin, 1987.

Huber, G., "The Nature and Design of Post-Industrial Organizations", Management Science, vol. 30, no. 8, August 1984.

Janson, M., "Applying a Pilot System and Prototyping Approach to Systems Development and Implementation, Information and Management, vol. 10, no. 4, 1986.

Keen, P. and Scott Morton, M., Decision Suport Systems: An Organizational Perspective, Addision-Wesley, 1978.

Kneale, D., "How Coopers and Lybrand Put Expertise Into Its Computers", Wall Street Journal, p. 33, November 14, 1986.

Kozlov, A., "Rethinking Artificial Intelligence", High Technology Business, May 1988.

Kupfer, A., "Now, Live Experts on a Floppy Disk", Fortune, October 12, 1987.

Leonard-Barton, D., "The Case for Integrative Innovation: An Expert System at Digital", Sloan Management Review, Fall 1987.

Leonard-Barton, D. and Sviokla, J., "Putting Expert Systems to Work", Harvard Business Review, March-April 1988.

Linden, E., "Intellicorp: The Selling of Artificial Intelligence", High Technology, March 1985.

Luconi, F., Malone, T., and Scott Morton, M.S., "Expert Systems: The Next Challenge for Managers", Sloan Management Review, Summer 1986.

Lucas, H., Information Systems Concepts For Management, McGraw-Hill, 1982.

McDermott, J., "R1 Revisited: Four Years in the Trenches", AI Magazine, Fall 1984.

Mettrey, W., "An Assessment of Tools for Building Large Knowledge-Based Systems", AI Magazine, vol. 8, no. 4, Winter 1987.

Senn, J., Analysis and Design of Information Systems, McGraw-Hill, 1984.

Sprague, R. and Carlson, E., Building Effective Decision Support Systems, Prentice-Hall, 1982.

Waterman, D., A Guide to Expert Systems, Addison-Wesley, 1986.

Inception

Feasibility Study

Systems Analysis

Design

Specifications

Programming

Testing

Training

Conversion/Installation

Operations

Figure 1. A Traditional Systems Analysis and Design Framework (from Lucas 1982)

Inception + Prototype Proposal

Approval of Prototype Development

Working Versions (Incremental Improvements)

Demonstration Prototype + Feasibility Report

Approval of Project

Development of System for Test Release

Approval of Test Release System (Evaluation Criteria Passed)

Test System Released

Evaluation of Test System Performance

Improvement of Test System For General Release

General Release (perhaps phased in)

Evaluation of General Release System Performance

Maintenance/Updating

Figure 2. Life Cycle of An Expert System

February 1986:

Knowledge Engineer brought on board; started coordination with external consultants; selection of Comet application.

April 1986:

Evaluated ES shells; coordination with internal AI group.

July 1986

Developed first prototype (for one Comet module) on a PC using Personal Consultant Plus from Texas Instruments; decision to go to OPS5 based system on Vax; transport prototype to Vax.

September - October 1986

Prototype reviewed; project approved; direction defined to include four Comet modules, second Knowledge Engineer brought on board.

November 1986 - May 1987

System capabilities expanded, review for test release; evaluation criteria better defined; problems/successes isolated.

June - September 1987

Trial implementation; approval for phased in full release.

October 1987 - March 1988

Phased release to Comet field repair sites; remote update acquisition; project evaluation.

Figure 3. Milestones of the Comet ES project

	Traditional System	Expert System
Inception	Approval for: feasibility study. Task Selected: information-based	Approval for: feasibility study and demonstration prototype. Task Selected: knowledge-based
Prototype	Not Applicable	Working, limited version. Incrementally developed.
Feasibility Study		Includes evaluation of prototype. ES specific costs and benefits.
Analysis Design Specifications Programming	Consecutive, well defined stages.	Iterative, incremental process. Additional players required: knowledge engineer and domain expert.
Testing	Well established procedures.	"Right" answer may not exist. Experts may disagree.
Training		Likelihood that users are skilled "knowledge workers".
Conversion/ Installation		Possibility of non-standard hard/software.
Operations	Fixing errors. Occasional updates.	Likelihood of frequent updates.

Figure 4. Systems Analysis and Design: Contrasting Traditional and Expert Systems