Chapter 5

INTEGRATION OF INTELLIGENT TECHNOLOGIES INTO CONVENTIONAL INFORMATION SYSTEMS: KEY ISSUES, OPPORTUNITIES AND POTENTIAL PITFALLS

Paul R. Watkins
Daniel E. O'Leary
University of South Carolina, Los Angeles, CA

Applied artificial intelligence technologies such as expert systems, natural language processing, intelligent databases and neural nets are receiving considerable attention in both the academic and professional communities. Much of this attention has focused on the exploration and development of expert systems to support decision making and problem solving. This chapter describes a views model for evaluating integration issues in intelligent technologies.

5.1 INTRODUCTION

A recent study by O'Leary and Watkins (1989a) showed a wide variety of application areas being addressed through the use of expert systems technology by Fortune 1000 firms. Much of the early expert systems development has focused on standalone systems which are self-contained and independent of any existing conventional management information systems (MIS)/decision support systems (DSS) in the organization. With the wide availability of expert systems shells, the trend toward standalone expert systems development is expected to continue, primarily through means of end-user development on microcomputers. Concurrent with the continued development of standalone systems is the desirability of integrating not only expert systems, but other derived artificial intelligence technologies into conventional information systems in organizations. Note that the conventional information system may be defined as any existing (non-intelligent) information systems technology, regardless of the hardware delivery mechanism such as PC's, Mainframes, Minis, Distributed Networks, and so on.

Expert Systems in Business and Finance: Issues and Applications. Edited by P.R. Watkins and L.B. Eliot

Turban and Watkins (1986) present a framework for the integration of expert systems and decision support systems. While this framework serves as a general basis for considering expert systems and decision support systems integration and presents arguments describing the synergistic benefits of ES/DSS integration, it does not address many of the specific issues that concern integration of intelligent technologies with conventional information systems. Also implicit in the Turban, Watkins (1986) article is a rather singular definition of integration which focused on hardware and software issues.

The focus of this chapter is to consider problem solving/decision making as the key motivation for the existence of conventional information systems and the augmentation/enhancement of these systems with intelligent technologies. In order to effectively support and enhance problem solving and decision making, the appropriate data, information and knowledge must be made available. Thus, this chapter develops a conceptual model for augmenting current conventional technologies in support of decision making/problem solving by considering the integration of knowledge provided by several types of intelligent systems such as knowledge based systems, expert systems and neural nets. The characteristics of each of these technologies (both conventional and intelligent) are described both independently and from a point of view of integrating these technologies into conventional information systems. The concept of integration is further developed and expanded from that of prior studies into a views model which provides a more comprehensive perspective of the integration issues.

The chapter is organized as follows: the subsequent section provides a conceptual framework of the environments of information systems and the attributes that might be considered most relevant in systems integration. Next the conceptual views model of integration is described and a discussion is presented of data, information and knowledge. The concluding section of the chapter provides a summary set of issues, opportunities and pitfalls which potentially derive from the integration effort.

5.2 CONCEPTUAL MODEL OF INFORMATION SYSTEMS INTEGRATION

Information systems in most organizations can be viewed from a variety of perspectives. Figure 5.1 presents a perspective that shows the various external and organizational boundaries of a typical information systems environment.

5.2.1 General Characteristics

Of primary importance within the boundaries are the task/problem domain and the decision making domain. The task and the decision making activity,

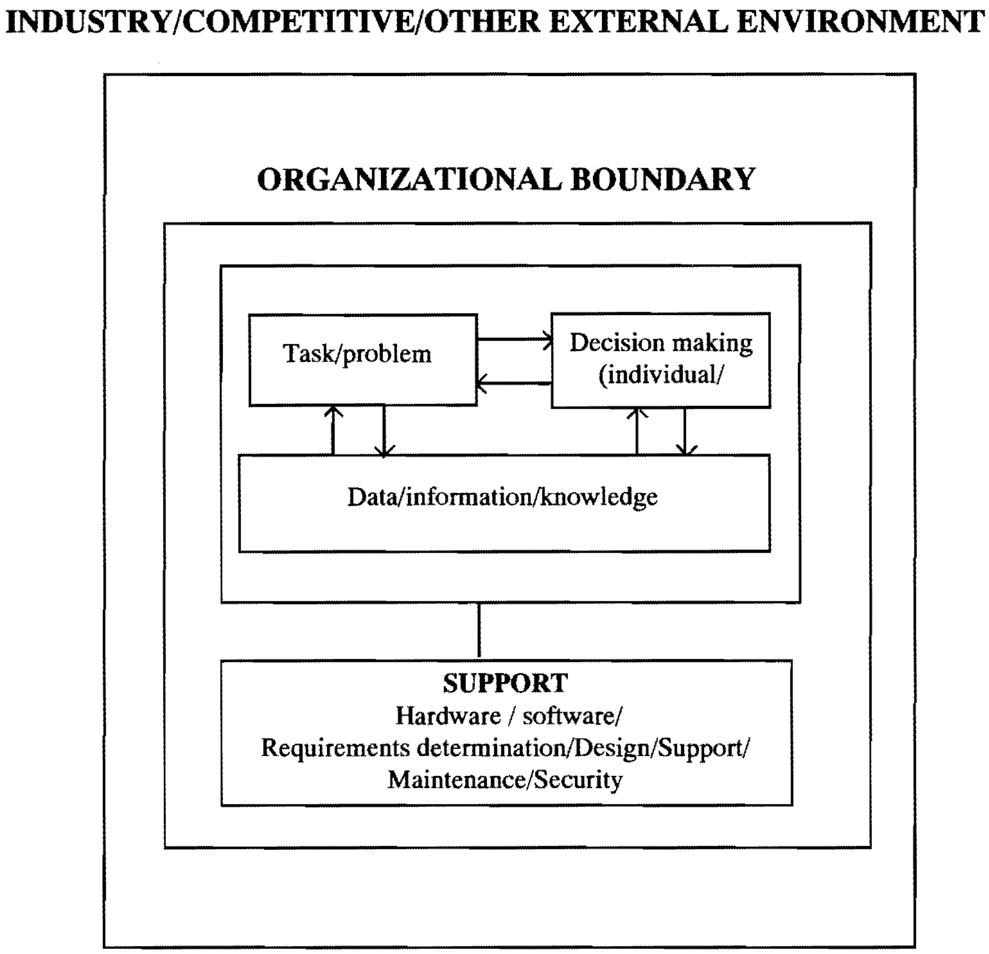


FIGURE 5.1 Components of conventional information systems environments

to a large extent, determine the kinds of data, information and knowledge required to solve a particular problem or to make a reasonable decision. As is known from prior information systems research, the database is crucial to effective information systems support for decision making. (Note at this point information systems are described in a generic sense without delineating between various components such as MIS, DSS and so on.) Somewhat secondary to the three major systems components—task problem domain,

PROBLEM/TASK DOMAIN:
General business domain/problems
Specific domain:
- Operational
- Planning
- Control

DATA/INFORMATION/KNOWLEDGE:
Data
Information
Knowledge

DECISION MAKING DOMAIN:
Routine, structured vs unique, complex
Judgment support
Delegation
Monitoring
Conflict resolution
Consensus building
Training/technology transfer

SYSTEMS DOMAIN:
Transaction processing systems - TPS
Management information systems - MIS
Decsions support systems - DSS
Executive support systems - ESS
Knowledge based systems - KBS
Expert systems - ES
Neural nets - NN

ORGANIZATIONAL ENVIRONMENT:
Power
Control
Goal Congruence
Political

DESIGN/IMPLEMENTATION/SUPPORT:
Requirements determination
Design
Implementation
Support
Maintenance
Security
Auditability

HARDWARE DOMAIN:
Standalone PC's, workstations
Networked environments
Distributed environments
Large, central computing: minis, mainframes

SOFTWARE DOMAIN:
User view
Designer view
Maintainer view
Management (not necessarily the user) view
Developer view: software engineering issues

FIGURE 5.2 Key attributes for consideration in information technology integration decision making domain and data/information/knowledge domain—is the support domain which includes the software, hardware, requirements determination/design, support, maintenance, security and other systems design/implementation/support components.

5.2.2 Detailed Characteristics: A Views Model

Figure 5.2 expands the components in Figure 5.1 to include more detailed attributes of the information system. These attributes are not intended to be exhaustive of all the attributes of an information system but rather to focus on attributes that are of key importance as the integration of various types of systems is considered. As seen in Figure 5.2, eight major categories of attributes are presented. These groupings of attributes represent views from which the integration of conventional and intelligent technologies can be discussed. View 1 is the problem/task/decision domain. This domain is considered from the classic Gorry, Scott-Morton (GSM) (1971) framework.

View 1: The Problem/Task/Decision View

The GSM framework provides a starting point for considering the task/problem and decision making attributes which may require support by an information system. As shown in Figure 5.3, the GSM framework shows structured, operational control type task/decision environments in the upper left hand corner of the diagram and unstructured, strategic planning decisions in the lower right hand corner of the diagram. As has been discussed previously in the literature, different levels of tasks and decision making require different kinds of information systems support. Originally, Gorry and Scott-Morton (1971) suggested that the information characteristics of interest for effective systems design are source, scope, level, time horizon, currency, require accuracy, frequency of use. Thus, these characteristics of information would likely differ among structured, operational tasks and unstructured, strategic tasks. This basic framework has been useful in providing perspective for management information systems (MIS) and decision support systems (DSS). MIS were conceptualized to be more apropos to support the problem domains represented by the upper left hand triangle of Figure 5.3 and DSS were conceptualized to be more apropos to the problem domains represented by the lower right hand triangle. Although basic characteristics of information were provided in the GSM framework, they are not completely sufficient for many problem domains and tasks. Thus in the next section we provide a discussion of View 2: The Data/Information/Knowledge View.

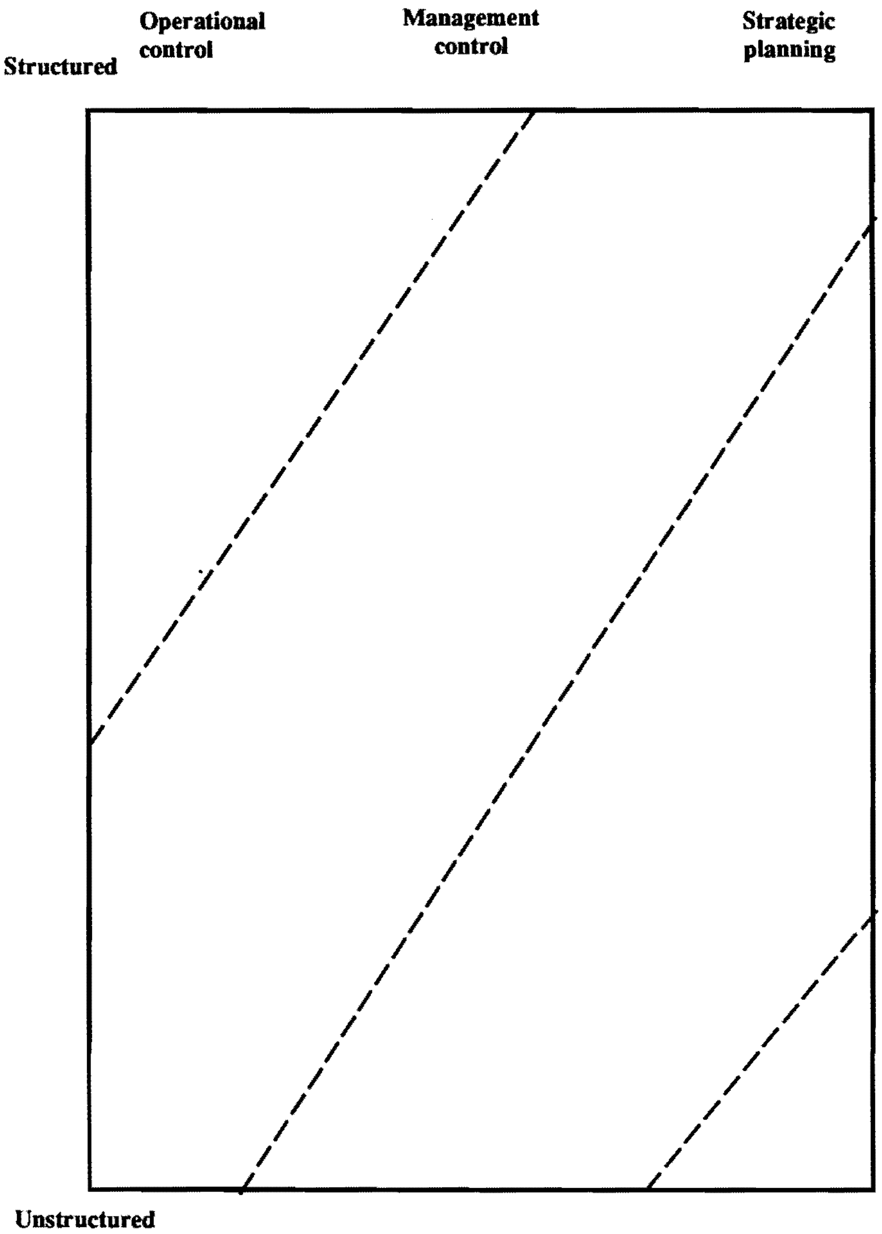


FIGURE 5.3 Integration View 1: Problem/Task/Decision View

View 2: Data/Information/Knowledge View

Definitions of data and information are fairly well known in the information systems literature. Table 5.1 presents general definitions for data, information and knowledge. The distinction between information and knowledge is important for several reasons. First, in view of the GSM framework above, information alone may be insufficient for supporting a give class of problems/decisions in the framework. For example, consider movement from the upper left hand corner of Figure 5.3 to the lower right hand corner of the figure in view of data/information/knowledge requirements for the following items:

- Operational, structured decisions may require more numeric, data driven approaches (e.g., linear programming) whereas unstructured, strategic planning decisions may require more symbolic knowledge and less data.
- Operational, structured decisions may be more data intensive, and require structured static knowledge (e.g., accounts receivable) whereas strategic planning, unstructured decisions may be more expert knowledge intensive (mergers and acquisitions or R&D planning).
- Operational, structured decisions may require more well defined and predictable data and knowledge whereas unstructured, strategic planning decisions may require less well defined, unpredictable knowledge necessary to solve the decision problem.
- Operational, structured decisions typically utilize stable data and knowledge (e.g., approaches to accounting for accounts receivable) whereas unstructured, strategic planning decisions may only have available unstable knowledge (mergers and acquisitions) necessary to solve the decision problem.

TABLE 5.1 Definitions for data, information and knowledge from a systems point of view

Data
Data are basically facts, primarily numerical in nature, which form the basis of much of the contents of databases in business organizations.

Information
Transformed data into a form that is useful to the decision maker and/or organization. Information may be items input into or developed by the system which may be helpful for decision making but may often suggest isolated or unrelated situations or facts.

Knowledge
Applies to the understanding of a set of facts or principles or of information. Thus knowledge transcends facts and information and provides a means of intelligently utilizing the facts and information in decision making.

"She has acquired much information but has little knowledge"

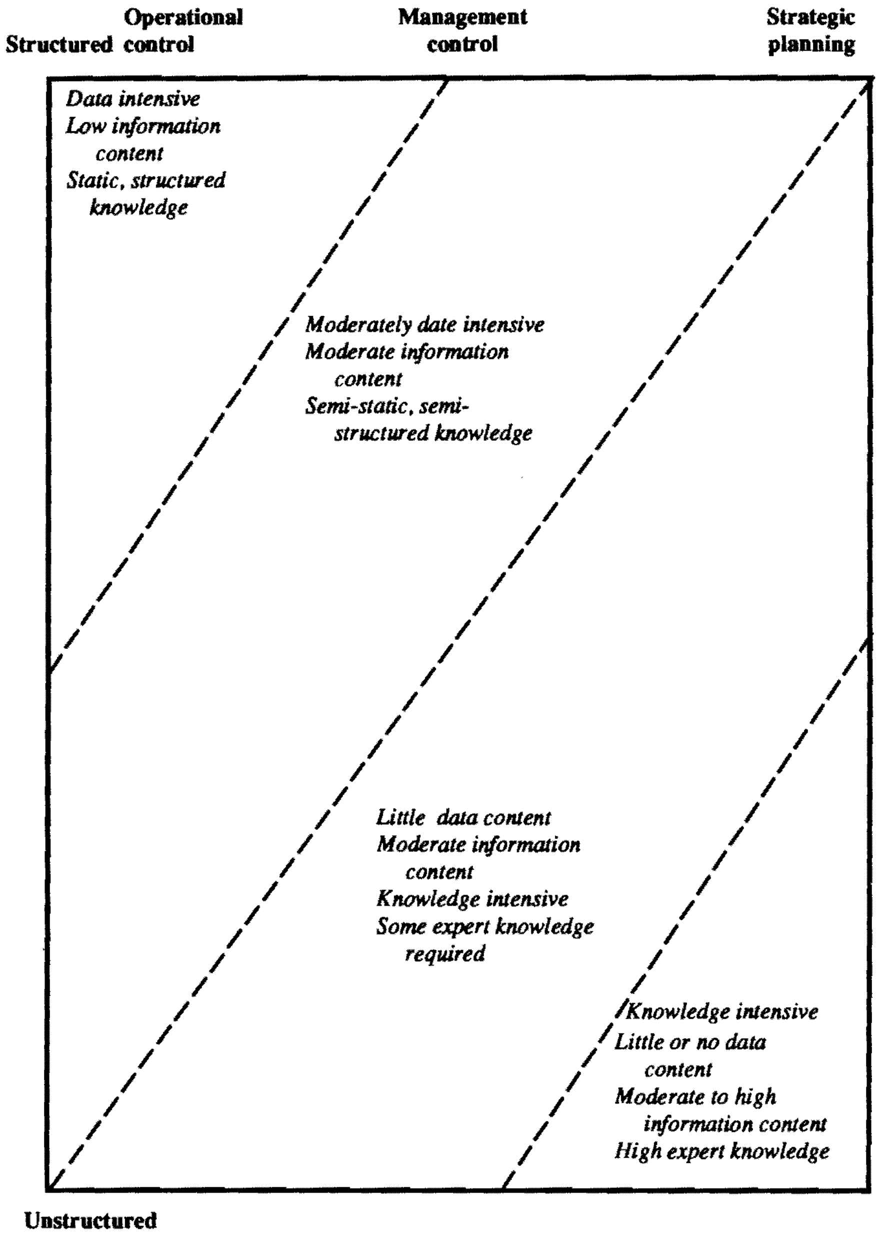


FIGURE 5.4 Integration View 2: Data/Information/Knowledge View

Knowledge, then, is how information is applied or utilized in a given task domain and the more unstructured, complex the domain is, the more the requirement for expert knowledge. The more structured and routine the task domain is implies more structured, less expert based knowledge. Knowledge, as stated before, may be a set of rules or principles, perhaps in the form of policies, procedures, requirements, conditions and so on. This type of knowledge is potentially useful for a class of decisions as indicated on Figure 5.4. This class of decisions requires data, information and knowledge about how to organize and apply the data and information. Another definition of knowledge is expert knowledge; that is, the experiences and organized knowledge structures of an expert in a particular area which can be captured and transferred to another source, such as an expert system. This type of knowledge is located on Figure 5.4 in the lower right-hand corner (unstructured, complex decision making).

One contribution of this conceptual model is the recognition of the distinction between data, information and knowledge and that there is a wide class of problems for which knowledge is desirable; especially to augment current data and information based systems. Knowledge can be applied to structured, operational decisions as well as unstructured, strategic planning decisions. The type of knowledge varies from static, structured rules for the structured task domains to more dynamic, adaptive, conceptual knowledge for the unstructured domains. Thus one type of integration summarized by View 2 is the integration of knowledge with data and information.

View 3: Decision Making View

View 1 described the problem/task/decision view. Decisions in View 1 were structured–operational, structured–management control, structured–strategic planning, semi-structured–operational, semi-structured–management control, semi-structured–strategic planning, unstructured–operational, unstructured management control, and unstructured–strategic planning. In addition to these decision areas, View 3 is concerned with individual or group judgment support (different from decision support), delegation in the decision making process, monitoring of decisions, conflict resolution, consensus building, and training/technology transfer.

Purpose of the System Systems designed to solve a specific task or support a specific decision may also provide decision making information that can be integrated into other decision making domains. For example, in one Fortune 500 firm (TRW) with which the authors are familiar, output from a system designed to detect intruders into an MIS system proved to be quite useful in the generation of data about the clients of a database service, for marketing purposes. Another system developed by a New York bank to audit

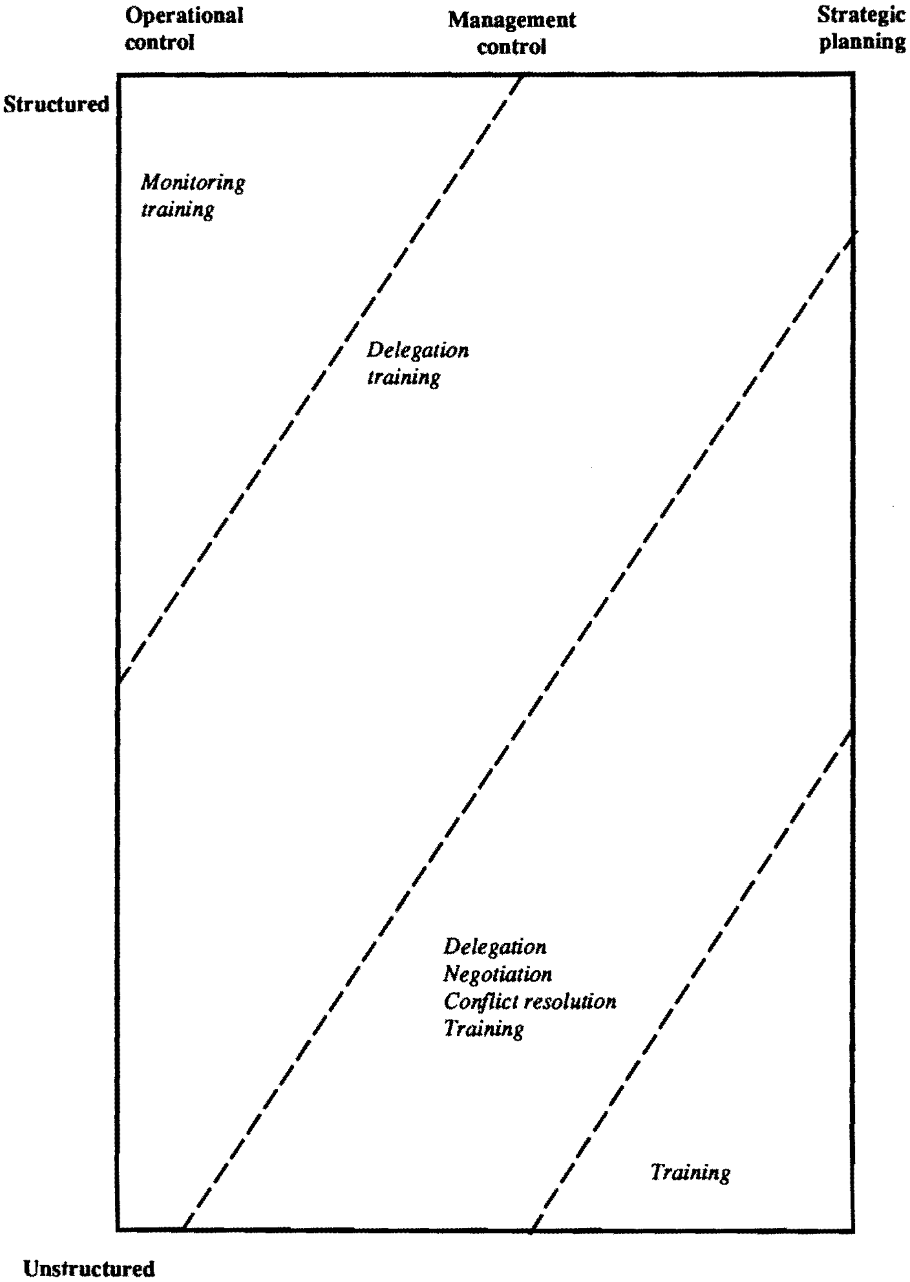


FIGURE 5.5 Integration View 3: Decision Making View

foreign currency trading also was found to be useful as a management tool (O'Leary and Watkins (1989a). Thus, in Figure 5.5, an operational–semi-structured decision environment provided output which then could be integrated into a system providing semi-structured–managerial control support.

Delegation In many situations the role of delegating tasks to subordinates by managers can be replaced or augmented by appropriate integration of delegation based knowledge systems. For example, routine, semi-structured tasks may be augmented by a task delegation system where the system replaces the traditional managerial delegation task as indicated in Figure 5.5.

Monitoring Oftentimes, monitoring is a relatively stable activity that intelligent systems can perform in an efficient manner (e.g., Denning, 1987). Accordingly, intelligent systems can be embedded in other systems in order to monitor access, examine input or output for errors, and a variety of other functions. Monitoring systems may be more appropriate at the structured–operational control level as shown in Figure 5.5.

Negotiation —Conflict Resolution Increasingly intelligent systems offer the opportunity to computerize negotiation processes. Such approaches are beginning to be integrated into decision making processes. For example, at Westinghouse:

> With OptionFinder, each meeting participant has a small numeric key pad, with which he or she votes into the OptionFinder Software, which instantly tallies them and produces a graph of the results... *Because everyone must participate, must push their button, the tools encourage teamwork* (Wilkinson,1989).

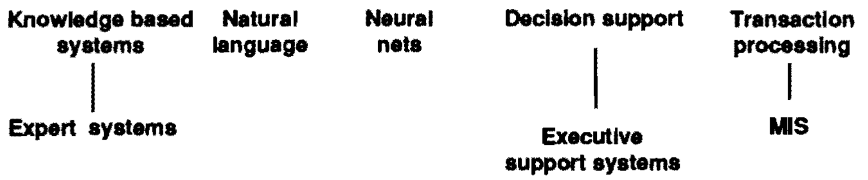
Thus, negotiation type augmentation may be more at the semi-structured, planning level as shown in Figure 5.5.

Training Training, although not specifically identified with decision making, can change the manner in which decisions are made and encourage more rapid and effective technology transfer by embedding appropriate training modules into existing systems. For example, in Figure 5.5, training modules can be embedded into all of the problem/task/decision supporting areas from very structured to unstructured tasks and thus provide benefits in terms of training integration. This may enhance decision making performance over time and allow managers to provide more realistic learning environments for new hires and employees who have been assigned to new tasks.

View 4: Systems View

The systems view looks at the various intelligent and conventional systems and describes their characteristics from a standalone and integrated point of view. Figure 5.6 provides a hierarchical view of conventional and intelligent information systems. This view is a non-integrated view, that is, each system is conceptualized as a standalone system. The definitions of transaction processing systems (TPS), management information systems (MIS), decision support systems (DSS) and executive support (or information) systems (ESS) are generally well known, although there may be disagreement as to the precise definition of DSS, and ESS. The definitions on Figure 5.6 are to provide a point of reference and perspective, not necessarily to argue strongly that a given definition is precise and unconditionally accepted.

Definitions of knowledge based systems (KBS), expert systems (ES), natural language (NL) and neural nets (NN) are less well known in the systems literature and are now briefly discussed. Knowledge based systems form the broad category of systems that derive from applied artificial intelligence. The argument is that much of intelligence in any system, human or machine, is based on knowledge. Knowledge based systems then may contain detailed, factual knowledge which has little or no expertise component or they may contain a great deal of expert knowledge. For this chapter, expert systems are considered to be a sub-set of knowledge based systems wherein a high level of *expert* knowledge about a given task domain is present. KBS, then, contain knowledge, but not necessarily expert knowledge. Natural language systems are well known in the systems literature and have been proposed as front ends to database systems and for user interface technologies for decision support systems. Little more will be said about these systems in this chapter. Neural nets are artificial intelligence derivatives which have application in a wide variety of settings. Of particular interest for business information systems are NN that are focused on pattern recognition of certain kinds of activities such as accounts receivable collections, or mortgage or commercial loan applications. The NN technology is based on scanning patterns of characteristics of the task over a large history file and generating a template or master pattern which can be applied to current task settings. A logical extension of NN is to interface these with other intelligent technologies such as KBS or ES for interpretation and decision making activities. In the systems view of integration, the goal is to integrate the standalone systems with each other for a particular problem/task/decision and to support the appropriate integration of data/information/knowledge. Thus various combinations of systems can be envisioned as appropriate. For example, ES-DSS, ES-MIS, ES-TPS, KBS-DSS, KBS-MIS, KBS-TPS, ES-NN-TPS, ES-NN-TPS, KBS-NN-MIS, KBS-NN-TPS and so on. The emphasis should be placed on the data/information/knowledge needs



DEFINITIONS

Knowledge Based Systems: Systems which have factual, procedural and little or no *expert* knowledge for a given task or problem domain. Knowledge is represented primarily in the form of rules. Has an inference mechanism. Knowledge base may be relatively static (accounts receivable collectibility system) and need little or no maintenance or for some domains may need frequent updates (e.g. tax rules adviser).

Expert Systems: Systems whose primary type of knowledge is obtained from experts for a given problem or task domain. Knowledge may be represented in a variety of ways. Has varied inference mechanisms. Knowledge base may need modification in view of new experiences.

Neural Nets: Specialized processing networks which are adaptive based on pattern recognition of task histories. An example would be an equities trading program where the neural net looks at volumes of stock market data and correlates data patterns with probable stock market behaviors.

Transaction Processing: System oriented towards processing high volumes of transactions, e.g., bank check clearing systems.

MIS: System which focuses on generation of reports for a variety of management support activities. May be part of the transaction processing system.

DSS: Focused on supporting decision making. Key element is database and may have other components such as a models manager and so on.

ESS: Executive support system. Focus is on support for top executives and provides graphics and other presentation views for executive support. May have mail systems and dialog systems interconnected.

FIGURE 5.6 Hierarchy of intelligent and conventional systems of View 2 within the context of the problem/task/decision domain of Views 1 and 3. The section on View 8, Software View, provides additional issues for the systems integration view. The primary focus of the remainder of this chapter is on the integration of KBS and ES into conventional systems. Although conceptually, the organizing concept is knowledge rather than a specific system, the fact that separate KBS and ES are being developed in organizations suggests that this discussion focus on the concept of knowledge as well as KBS and ES.

In Figure 5.7, the various types of systems are shown in the context of the problem/task/decision domain and the data/information/knowledge domain of Views 1 and 2.

View 5: Organization Environment View

The organization environment view may be evaluated from the perspective of Figure 5.1 where the organizational boundary encompasses the information systems activity within the organization. Theorists such as Katz and Kahn (1966, 1986) and others commonly use at least the following seven categories in discussing organizations:

- Organization structure
- Centralization/decentralization
- Leadership and power
- Organization roles
- Communications and communications flow
- Corporate culture

All of the above seven categories can have an impact on the problem/task/decision domain, the data/information/knowledge domain and the decision making domain. Much of the discussion that follows views integration in the context of standalone versus integrated or non-standalone systems.

Organization Structure An environment where applications are integrated as opposed to standalone, requires different organizational structure. If the systems are standalone systems then this likely would encourage independent departments either to develop their own intelligent systems or to develop a department that would specialize in such systems. On the other hand, if intelligent systems are integrated into other types of information systems then the department where expertise on intelligent systems is resident is likely to be integrated with the expertise on the development of those other systems.

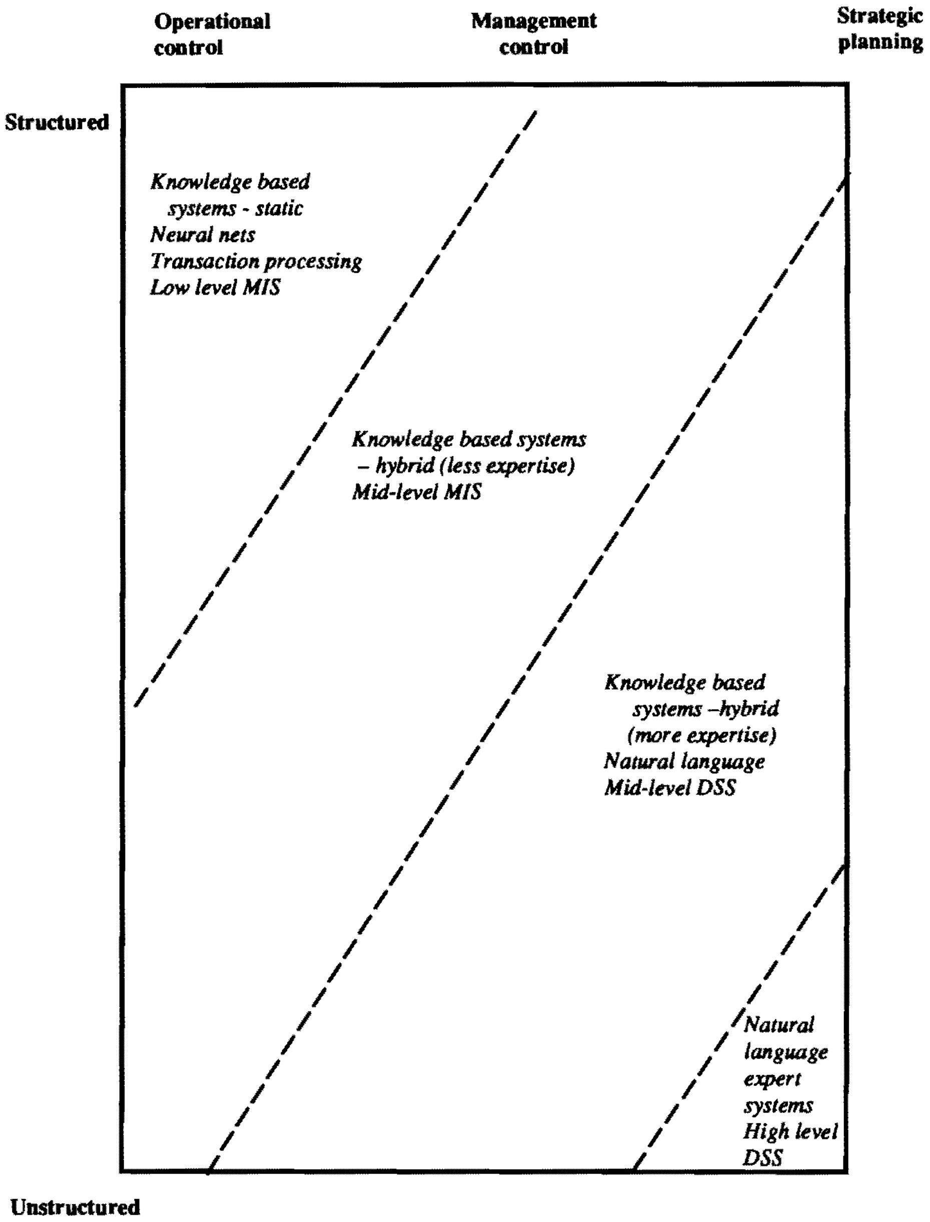


FIGURE 5.7 Integration View 4: Systems View

Centralization/Decentralization If all knowledge and expertise is resident in only humans or standalone systems then it becomes more difficult to centralize decision making since the decision making resources are too dispersed. Similarly, if knowledge and expertise is integrated into other systems, then it becomes easier to centralize, since the decision making resources also are integrated.

Leadership and Power Several researchers (e.g., Katz and Kahn, 1966, 1986) have noted that both leadership and power at least partially derive from expert knowledge and access to information. In standalone systems the expertise, and thus the power for expertise, remains distributed with particular individuals or groups of individuals. However, in integrated systems, the expertise is implicitly available to all. Thus, it appears that an important variable in determining the feasibility of integrating expertise into a system is the *willingness* to lose the potential power of distributing the expertise. As noted by Mowshowitz (1985, p. 102), *When real money and power are at stake the realization of a redistribution of control takes more than mere technological possibility.*

Alternatively, some may argue that by integrating intelligence into a system that power can be broadened. For example, if integrating expertise into a system means limiting the feasible set of options of the users then power of management can be increased. For example, many of the systems designed to support auditors improve the control that management would have over auditors in the field (e.g., O'Leary and Watkins, 1989a).

Organization Roles As expert and knowledge based systems are developed, some new roles can be added to the organization. One pivotal role in stand alone expert systems is the expert system manager (O'Leary and Watkins, 1989a). This manager is responsible for monitoring and maintaining the performance of the system, as it evolves over time. However, if the system were embedded or integrated into a larger information system that role might change to one of less emphasis on a single application and more emphasis on the system.

Communications and Communications Flow If an intelligent system is developed for a standalone environment, then clearly that system is physically accessible to only a few members of the firm and inhibits communications flow.

Integrating intelligent systems within traditional types of systems can facilitate communications. For example, a module representing expertise can provide communication from the expert to the group using the system. O'Leary and Watkins (1989b, c) discuss an integrated expert system designed to facilitate communication of technical updates. The system was integrated with a database of analyst updates issued by management to clarify changes in technology. Communication in the other direction could be facilitated by a system that tracked what was done with expertise recommended by the system.

Corporate Culture The corporate culture also can play a critical part in the adoption of technology, such as integrating intelligent systems. In Wilkinson (1989), corporate culture is credited with being a critical part in the development of a group DSS designed to streamline meetings.

Thus, organizational environment issues can have a major effect on the integration effort and both mandate organizational structural changes as well as force technology into a particular role in order to meet organizational needs.

View 6: Design/Implementation/Support View

Integration issues arise as systems and components of systems are designed, implemented and supported through maintenance functions, security and audits. Since characteristics of intelligent systems may differ from conventional systems, care must be taken in the integration effort to insure recognition and compatibility of the differences that arise as divergent yet complementary technologies are integrated.

Requirements Determination Prototyping is the promulgated basis of requirements determination in expert, knowledge based and decision support systems. Its use as a tool in standalone systems has been investigated by a number of authors in transaction systems (Earl, 1982), decision support systems (Henderson and Ingraham, 1982) and expert systems (O'Leary, 1988a). Jenkins and Fellers (1986) have accumulated a substantial bibliography on prototyping.

Although the use of prototyping in traditional systems has been investigated empirically by Alavi (1984), generally, more structured *top-down* or *bottom-up* approaches are used or promulgated in traditional TPS or MIS. As a result, there can be a conflict in promulgated methodologies when ES, KBS or DSS are integrated with TP or MIS. If a structured approach is used then the less structured prototyping approach may provide inadequate inputs at various points in time in the requirements determination process. Unfortunately, the potential difficulties of the integration of alternative approaches to requirements definition has received little attention.

Design/Implementation The design and implementation of intelligent systems is somewhat different than that of conventional systems. In many cases, the problem/task/decision to be supported by intelligent systems is very narrow and thus becomes a very specialized project. One of the major difficulties

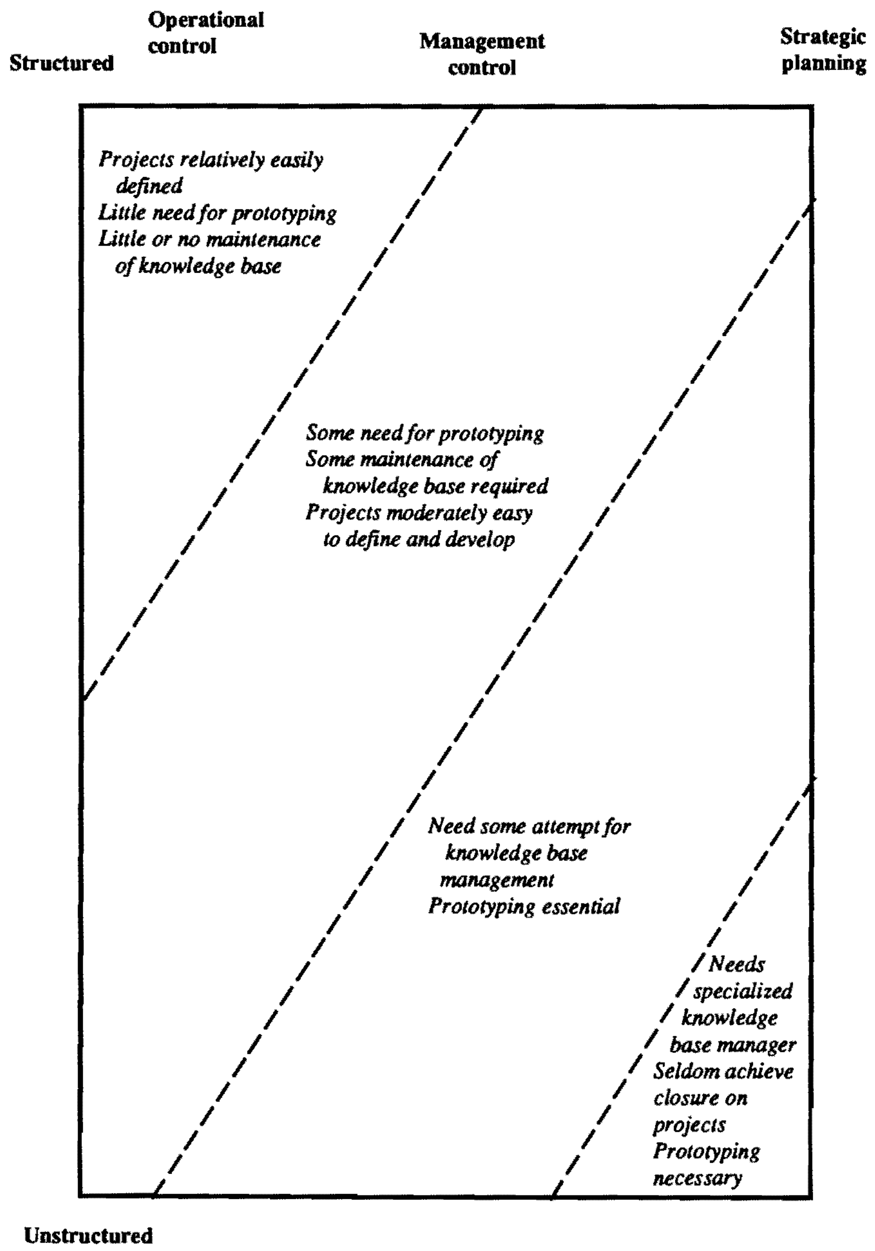


FIGURE 5.8 Integration View 6: Design/Implementation/Support/View

with the design/implementation of high-level KBS and ES is that they are seldom ever finished (Tener, 1988). That is, the knowledge base component is somewhat dynamic and requires constant maintenance and adjustment which in turn may have impact on the system. In many cases, high level intelligent systems are designed and built on a *trial and error* basis and are in a constant state of evolution and fine tuning. Thus, it may be difficult to say that closure has been achieved on a given KBS, ES project which makes it difficult for information systems management and other management to accept. Conventional information systems usually can be specified in advance with a definite and finite project development plan and timetable. High level KBS and ES can rarely be well specified in advance and timetables are rarely estimable.

KBS in support of structured tasks where there is little expert knowledge and where little maintenance of the knowledge base is required are more like conventional systems in the design phase and maintenance activity. As intelligent systems are developed in support of more unstructured tasks requiring *expertise* it tends to become less easy to completely specify the design in advance and provide reliable estimates of the development cycle.

Figure 5.8 provides some perspective for these differences in design/implementation/maintenance for the various types of decisions supported.

Compatibility Another key issue in the integration of intelligent technology into existing technology is in the area of systems compatibility and the feasibility of the integration effort. For example, the authors are aware of a major defense contractor who is attempting to integrate advanced expert systems technology with a first generation database and is finding the task almost all but impossible given the differences between the two technologies. Thus, although integration of intelligent technologies into existing technologies may be desirable from a conceptual point of view, it may be somewhat undesirable from a technical compatibility point of view. In industry there exist many older technologies simply because of cost benefit considerations in changing to new technology, e.g., the first generation database which is very large and would have tremendous costs for conversion, upgrading to newer technology.

Support/Maintenance The support and maintenance of integrated systems may be quite different from that of conventional systems. New job descriptions may have to be created and new training methods developed for employees in these positions. For example, in high level KBS and ES, a knowledge base management position is required. Analogous in concept to a database manager, in reality the job description for a knowledge base manager may be quite different (Tener, 1988). The knowledge base manager must not only have knowledge about the contents of the knowledge base of the intelligent system but must also have expertise in the problem/task domain to which the KBS, ES is to be applied. This is because the KB manager must be able to assess the impact of changes in the knowledge base to the manner in which the problem domain will be affected when the knowledge base modifications are utilized by an end user. In addition the KB manager must know what the impact of the knowledge base changes will have on the other components of the integrated system. For example, changes in the knowledge base may affect certain assumptions that the rest of the system relied on or conversely, changes in the environments or assumptions of aspects of the conventional systems could cause major problems for the knowledge base. Thus, knowledge based management is an area that requires a great deal of expertise and attention, yet very little is known about the qualities needed for successful knowledge base management.

Security Software engineers (e.g., Pressman [1987, p. 185]) define security as ... *the degree to which system and information access is protected...* Integrating intelligent systems into alternative computing environments can provide increased security risks or move toward mitigating security difficulties.

A recent conversation with a consultant indicated some of the security problems that can occur with the integration of expert systems into a computing environment that allows potential access to other software or databases. The consultant asked if he could examine a particular expert system. After providing the consultant with a brief introduction to the system, the consultant asked if he could *try out* the system on his own. As with many expert systems, the system had an easy-to-use-and-learn interface that allowed the experienced user to rapidly learn what was required of the user. Unfortunately, the system was so easy to use that the consultant was able to access classified database information. The system allowed the user to circumvent the security inherent in the database system. The user and the system effectively had different security levels.

Alternatively, since the law of requisite variety (Ashby, 1963) indicates that it takes equivocality to remove equivocality, this suggests that intelligent systems can be integrated into existing systems to assist in securing the system. Denning (1987) discusses such a system that is designed to detect intrusions to a computer system. As discussed in Tener (1988) intrusion-detection systems also can be integrated into a other applications.

Auditability As noted by Weber (1988, p.1), *EDP auditing is a function that has been developed to assess whether computer systems safeguard assets, maintain data integrity, and achieve the goals of the organization.* Integrating intelligent systems into other computer-based systems can either assist in the process of auditing the resulting systems or add to original systems complexity, and thus, the audit difficulty.

The work by Hansen and Messier (1986) is an example of the first situation. EDP-Xpert is a frequently referenced knowledge based system that is designed to assist in the audit of advanced EDP systems. AY/ASQ, developed by the accounting firm Arthur Young (now Ernst and Young), is a decision support system designed to assist in the audit of transaction-based systems.

Alternatively, by embedding an intelligent system into an existing system, auditing the resulting system becomes more difficult. Since expert systems and other intelligent systems generally are designed using different software and different development methodologies, including symbolic representation of knowledge, etc., they are conceptually different than the transaction processing software in which they are embedded. As a result, if an investigator were to audit the system they would find that alternative processes likely would be required to audit the embedded system. In addition, integrating such systems into other transaction processing systems means attending to the interfaces with those systems, thus, further complicating the resulting audit.

View 7: Hardware View

Integration of intelligent systems may present special challenges from a hardware point of view. Many high level expert systems have, in the past, been developed and implemented on standalone Lisp workstations. Rather than recode the systems to run in a production code such as COBOL or C, some firms have attempted to integrate the Lisp workstations with the conventional systems hardware. In many cases the issues of hardware compatibility and integration are no different from conventional systems issues. On the other hand, hardware capabilities can provide some major difficulties in the integration effort. For example, in a microcomputer environment, either standalone or networked, end-users may have some applications that they wish integrated with KBS or ES. The capabilities of the micros may be insufficient to run the KBS or ES efficiently. On the other hand, cost benefit analysis may suggest that specialized hardware is too costly and thus compromises in efficiency have to be made which can in turn affect the decision making environment. Figure 5.9 shows the effects of this integration in the conceptual model.

View 8: Software View

KBS and ES and other intelligent technologies are, after all, software products. The issues of software integration are therefore very important and-somewhat dominate the integration issues. Thus far in this chapter integration has been discussed at a non-software level. As software integration is now discussed it is useful to further refine the notion of integration.

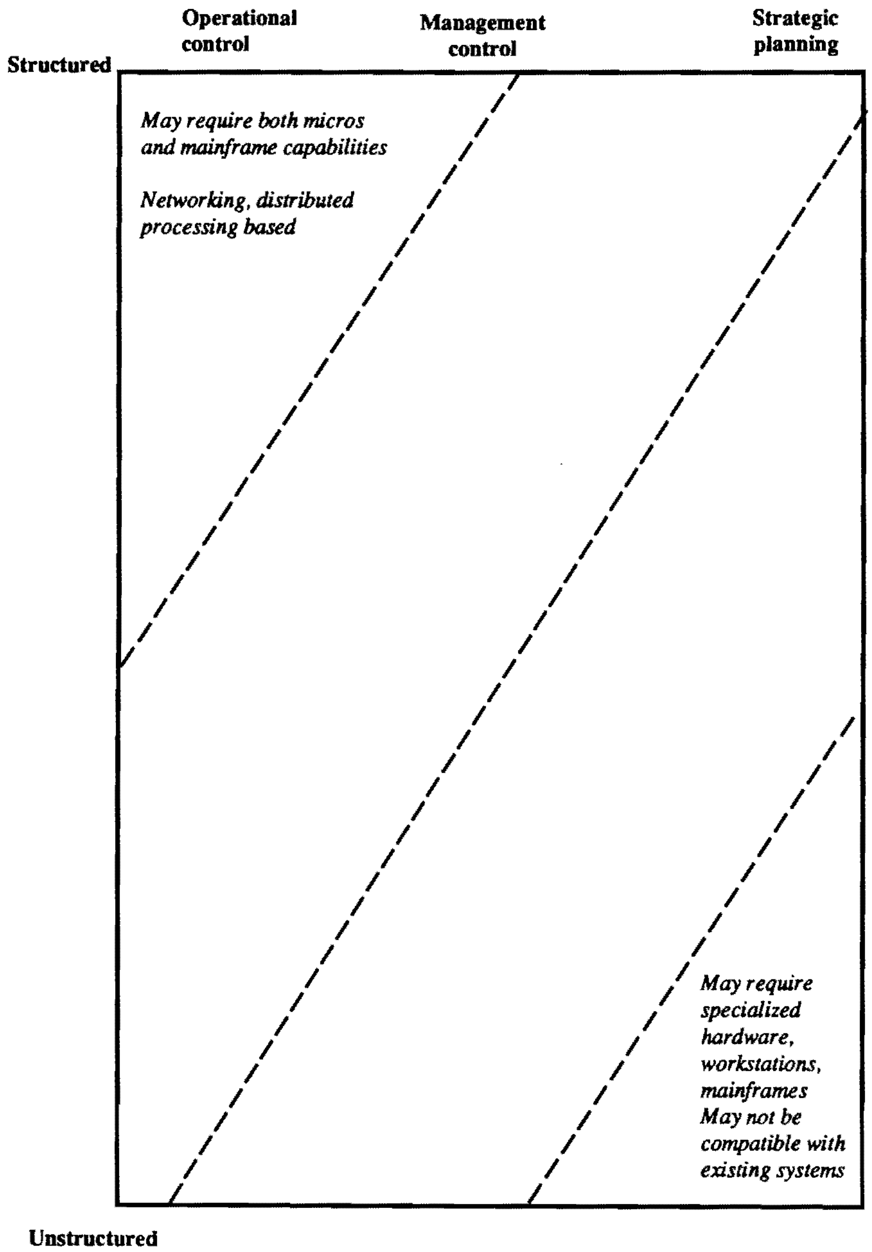


FIGURE 5.9 Integration View 7: Hardware View

Concepts of Integration Integration of systems is a somewhat nebulous concept. For purposes of clarity, a distinction is made between integration and the embedding of systems with each other. Embedded systems are integrated systems but integrated systems are not necessarily embedded.

Embedded systems are those that generally have been considered in a top-down design approach and have been built into the system as part of the overall programming effort. Examples of these type of systems are most commonly found on standalone applications for microcomputers where the software vendor has built in, say, a rule based, KBS component to the program but where to the user the program is a more friendly, powerful program. Examples include Borland Pardox, Symantec Q & A and Lotus Agenda where KBS are embedded into the software package as a seamless, overall strategic part of the software development effort. Another example would be the embedding of ES into CASE technology programs for re-engineering COBOL code in organizations.

Integration, which may include embedded systems, may simply mean augmenting two or more diverse technologies for the overall benefit of the organization. For example, Figure 5.10 shows the integration of a KBS into a conventional information system for the diagnosis of application software and hardware problems for communications applications between terminals and workstations of the user communities and the company networked mainframes. As shown in Figure 5.10, a knowledge base, rules and an inference engine are augmented to a conventional MIS which has various database components and files. Not shown in Figure 5.10 is that the conventional system is in a networked environment shared by multiple communications consultants. To the user, the KBS component is seamless, but to the developer and maintainer of the systems, the KBS component is simply integrated and needs separate attention and maintenance and specialized attention apart from the conventional MIS component. These views of the system will be in a subsequent paragraph.

Another example of integration is where a standalone ES is hooked up to an existing MIS to analyze outputs and render judgments based on those outputs. Turban and Watkins (1986) provide several examples of this type of integration. Engart (1989) describes the need for an ES component for LAN management and numerous other examples may be found.

There seem to be varying degrees of integration of intelligent systems and TPS, MIS, DSS and ESS. At one end of the spectrum are embedded systems and at the other end are manual transfers of information from one system to another. In large, mainframe environments, embedded systems are more prevalent in TPS whereas manual or non-seamless transfers of information are made between intelligent and conventional systems at the DSS level. On the other hand, with microcomputing, the standalone, vendor supplied packages, many of which are DSS in nature, tend to have the embedded intelligent systems whereas the TPS systems, such as accounting

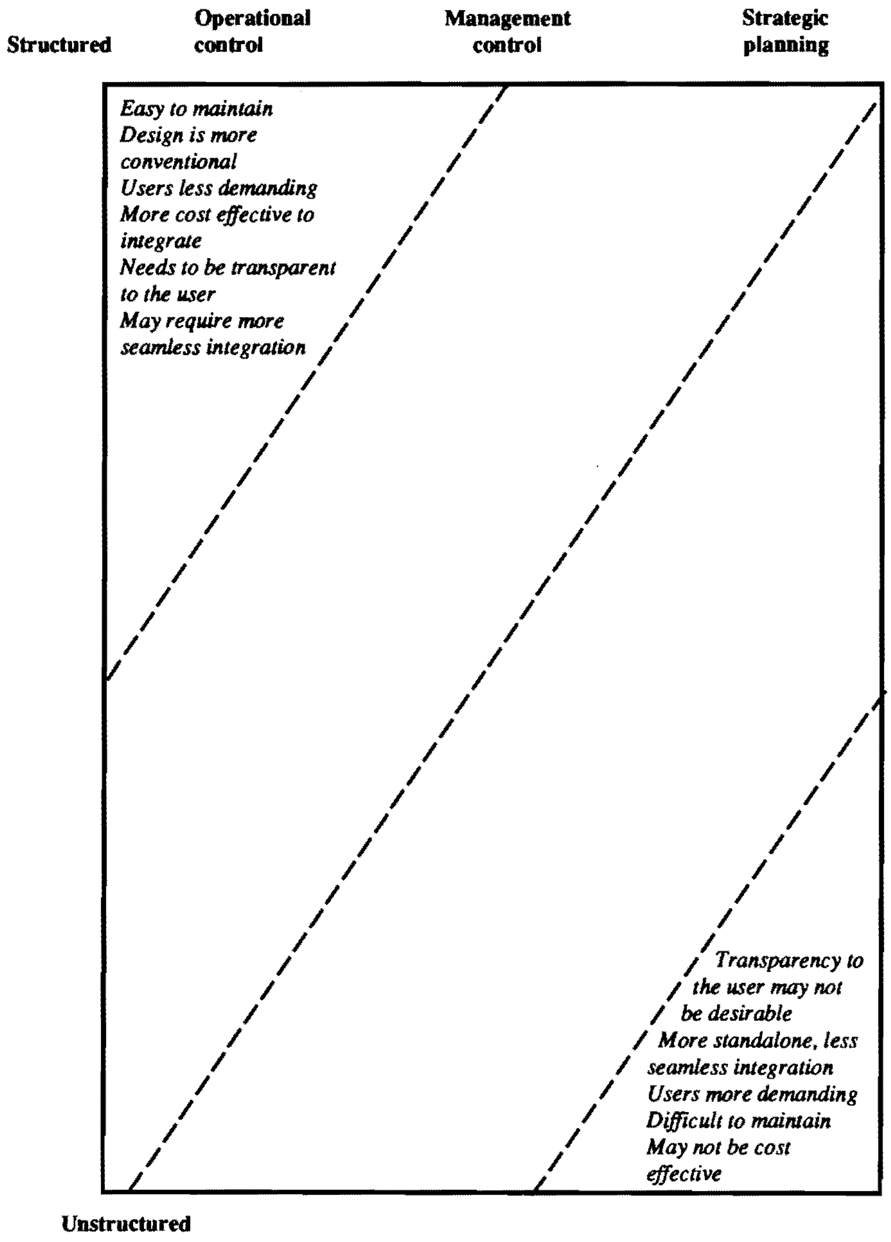


FIGURE 5.10 Integration View 8: Software View

packages, don't have embedded intelligence and require manual transfer or non-seamless transfer between them and intelligent systems.

As stated before, the focus of the integration, embedded systems activity, should be motivated and driven by the problem/task/decision considerations and the type of knowledge augmentation required in given problem/task/decision domains.

From the software integration view, several perspectives or views are worth discussing. These are the user, designer, maintainer, manager (not necessarily the user) and the developer's point of view. Each of these are now briefly described.

From the user's point of view, embedded or integrated systems are desirable if they solve a problem or make the task at hand more easily dealt with. The user wants results and the appearance of a system that solves a problem is the primary concern of the user. How the integration takes place and whether it is embedded or not is usually of no interest to the user.

The designer's point of view can vary depending on what is being designed. If a standalone intelligent application is being designed for a workstation, then the designer should be concerned with providing hooks or other means of interface to other technologies. On the other hand, the designer of a conventional application may wish to embed KBS components in the software to make the program more functional and provide more features to the user. In a large mainframe or distributed environment, designers need to provide for the possibility of a variety of software integration issues. For example, the authors are aware of a commercial software product (ES shell) for a mainframe environment that two Fortune 500 firms have been attempting to integrate into their mainframe conventional software systems. Both firms have been working on the projects for over a year now and are having major problems due to the lack of good design features in the shell which make the integration effort feasible.

The maintenance of the software may require different skills in an integrated environment. Conventional applications programmers may not have the expertise to maintain the KBS or ES components of the systems and thus a team approach may be necessary for effective maintenance. Figure 5.10 shows the maintenance effort may be a function of the type of problem/task/decision and data/information/knowledge circumstances. Structured support systems may be much easier to maintain than unstructured support systems.

The developer's view or software engineer's view may be quite different from the other views discussed. Software engineering is aimed at using engineering approaches in order to develop efficient and effective software in a cost beneficial manner. The integration of software engineering methods in information systems has been discussed in, e.g., O'Leary (1989a). In the case of software that is not complex it generally is easier to employ engineering principles in a formal manner. However, as the software becomes more complex, more *ad hoc* approaches seem to be employed. This is clearly the case with intelligent systems at one end of the spectrum, for which prototyping typically is promulgated, and transaction processing at the other end of the spectrum, for which more structured methods generally are promulgated. Unfortunately, integrating the *ad hoc* and the structured is an issue that has received little attention.

Figure 5.10 shows the impact of the problem/task/decision and data/information/knowledge views on the integration of software. For example, structured problems may be easier to integrate (e.g., accounts receivable) than unstructured tasks which are more difficult to integrate (R&D Planning). Accounts receivable inputs and outputs are stable, and thus are easier to integrate. On the other hand, activities such as R&D planning and mergers and acquisitions have less stable inputs and outputs.

Some applications may be necessary to integrate where others are less necessary to integrate. This is a function, in part, of the *closeness* of the application to the database. For example, accounts receivable provides information to a number of users, while R&D planning provides little immediate data to other users.

Applications may range from those which are embedded to those which are integrated but make knowledge transparent to user (e.g., assumptions, procedures, etc.). Because the knowledge employed in applications like accounts receivable are stable and the inputs and outputs are stable, such applications can operate with little intervention. On the other hand, even if applications such as mergers and acquisitions were integrated into an overall information system, generally, the user would be concerned about what assumptions are being made, what data is available, what kinds of reports should be prepared, and so on. (Note that this argument is for sufficiency—but it is not necessary to embed all such systems, e.g., ALDO. However, in a truly intelligent system, ALDO would be embedded, since such a system would track previous decisions, store information about those decisions in a database, be adaptive, etc.)

Another consideration is those applications which are cost effective to integrate to those which are not cost effective to integrate. Because of all these concerns it becomes less and less cost effective to integrate as you move from structured tasks in Figure 5.10 to unstructured tasks.

5.3 SUMMARY AND CONCLUSIONS

This chapter has focused on the major views of the integration of intelligent technologies into conventional information systems. As was discussed, eight views of integration were presented: problem/task/decision, data/information/knowledge, decision making, systems, organizational environments, design/implementation/support, hardware and software. Within these eight views are many sub-views and issues that were described. The fundamental theme of this paper is the conceptual model which shows that the problem/task/decision domain and the data/information/knowledge domain are the driving and principal forces in the model. That is, most integration issues can be put into perspective by considering these two domains. Other issues remain in the integration effort and subsequent research will address these issues in more detail. One effort of future research will be to look at combinations of existing systems and evaluate commonalities and differences as they apply to integration: such as DSS-ES or TPS-KBS and so on. The goal will be a comprehensive framework of integration attributes and components.

REFERENCES

Alavi, M. (1984), An Assessment of the Prototyping Approach to Information Systems Development, *Communications of the ACM*, June.

Ashby, R. (1963), *An Introduction to Cybernetics*, John Wiley, New York.

Denning, D. (1987), An Intrusion Detection Model, *IEEE Software Engineering*, Vol. **SE-14**, 3, pp. 252–261.

Earl, M. (1982), Prototyping Systems for Accounting and Control, *Accounting, Organizations and Society*, Vol. 3, No. 2, pp. 161–170.

Gorry, G. and Scott-Morton, M. (1971), A Framework for Management Information Systems, *Sloan Management Review*, Vol. 13, No. 1, pp. 55–70.

Hansen, J. and Messier, W. (1986), A Knowledge-based Expert System for Auditing Advanced Computer Systems, *European Journal of Operational Research*, September, pp. 371–379.

Henderson, J. and Ingraham (1982), Prototyping DSS: A Critical Appraisal, in (ed) M. Ginzberg, W. Reitman and E. Stohr, *Decision Support Systems*, North Holland, New York.

Jenkins, A. and Fellers, J. (1986), *An Annotated Bibliography on Prototyping*, Graduate School of Business, Indiana University.

Karlinsky, S. and O'Leary, D. (1988), A Framework for Taxation-Based Computer Decision Systems, *Expert Systems Review*, Vol. 1, No. 2, pp. 13–16.

Katz, D. and Kahn, R. (1966/1986), *The Social Psychology of Organizations*, Wiley, New York.

Ko, C. and O'Leary, D. (1987), The Impact of Culture on Expert Systems, *Proceedings of Pan-Pacific Conference IV*, Taipei, Taiwan, May.

Mowshowitz, A. (1985), On the Social Relations of Computers, *Human Systems Management*, pp. 99–110.

O'Leary, D. (1986), Expert Systems in Accounting in a Personal Computer Environment, *Georgia Journal of Accounting, Special Issue on Microcomputers*, 107–117.

O'Leary, D. (1986a), Expert Systems Prototyping as a Research Tool, in *Applied Expert Systems*, (ed) E. Turbin and P. Watkins, North Holland.

O'Leary, D. (1986b), Software Engineering and Research Issues in Accounting Information Systems, *The Journal of Information Systems*, Spring, pp. 24–38.

O'Leary, D. (1989), Security in Expert Systems, *IEEE Expert*, Forthcoming.

O'Leary, D. and Turbin, E. (1988), The Organizational Impact of Expert Systems, *Human Systems Management*, Vol. 7, pp. 11–19.

O'Leary, D. and Watkins, P. (1989a), Expert Systems in Internal Auditing, *Institute of Internal Auditors*, Altamonte Springs, Florida.

O'Leary, D. and Watkins, P. (1989b), Knowledge Acquisition for a Diagnosis-Based Task, *Under Review*, February.

O'Leary, D. and Watkins, P. (1989c), *An Integrated Expert System for Diagnosing Communications Problems*, Unpublished paper presented at the ORSA/TIMS meeting in Vancouver, May.

Pressman, R. (1987), *Software Engineering*, McGraw Hill.

Tener, W. (1988), *Integrating Expect Systems Into MIS*, paper presented at The First National Symposium on Expert Systems in Business, Finance and Accounting, September.

Turban, E. and Watkins, P.R. (1986) Integrating Expert Systems and Decision Support Systems, *MIS Quarterly*, Vol. 10, No. 2, pp. 121–138, June.

Watkins, P. and O'Leary, D. (1987), Knowledge Acquisition for Small Scale Expert Systems from Consulting Experts: A Field Study Approach, *First European Conference on Knowledge Acquisition for Knowledge Based Systems*, Reading, United Kingdom, September.

Weber, R. (1988), *EDP Auditing*, McGraw-Hill, New York.

Wilkinson, S. (1989), Decision Support Tool Streamlines Meetings, *PC Week*, February 6, p. 49 & 51.