



# WHITE PAPER: EXPERT SYSTEMS TECHNOLOGY

by Henry Brasza

## What is Expert Systems technology?

Expert Systems technology is more than a collection of hardware and software products. The techniques used to develop an Expert System are a critical component of the technology. These techniques, particularly in the areas of knowledge engineering and rapid prototyping, differ substantially from conventional development techniques.

## Why use Expert Systems technology?

Expert Systems technology and conventional systems technology are best-suited to different types of situations. Neither technology is a universal solution, nor does every situation demand the exclusive use of one technology or the other. The following table lists aspects of applications which suggest either an Expert System or a conventional approach:

Expert Systems Applications	Conventional Systems Applications
Knowledge is fragmented and implicit, is difficult to communicate except in small "chunks", and is often distributed amongst individuals who may disagree.	Knowledge is complete and explicit, and is easily communicated with formulas and algorithms.
Rules are complex, conditional, and often defined as imprecise "rules of thumb".	Rules are simple, with few conditions.
Problem-solving demands dynamic, context-driven application of facts, relationship and rules.	Problem-solving requires predictable and repetitive sequences of actions.
The finished system captures, distributes and leverages expertise.	The finished system automates manual procedures.
System performance is measured in degrees of accuracy and completeness where explanations may be required to establish correctness.	Simple criteria are used to determine accuracy and completeness.

# Why are Expert Systems development tools useful?

The measure of a development tool is how easily the developer can translate requirements and knowledge into an application. For example, the requirements of most business applications are more easily translated into COBOL than FORTRAN. In the context of scientific applications, however, FORTRAN usually provides a more natural translation.

In a problem situation conducive to the application of Expert Systems technology, the central task of application developers is to capture a precious but elusive commodity: expertise. As a result, the primary requirement for an Expert Systems development tool is that it provides an efficient and natural means to translate expertise into an application. This translation is easiest when the development tool represents knowledge in a manner that closely resembles the form in which knowledge is obtained from an expert.

Expert Systems development tools provide superior knowledge representation capabilities in two areas. First, all these types of development tools include an inference engine which enables the developer to use "chunks" of knowledge (rules) as building blocks for the overall system. Many Expert Systems development tools also include object oriented capabilities, which allow the developer to represent concepts and relationships between concepts.



Not only do these capabilities help the developer to transfer knowledge into an application, they also help the developer communicate the knowledge back to the expert(s) in order to verify its correctness.



## Are there any similarities between Expert Systems and conventional systems?

There are some loose parallels between Expert Systems and conventional systems, although the terminology used in describing Expert Systems is unique. The following list provides the most closely related conventional concept for a number of Expert Systems concepts, along with a brief comment concerning the key difference between the two:

Expert Systems	Conventional Systems	Brief Comment
Inferencing	Program Flow	Inferencing is non-sequential.
Knowledge Base	Database	Knowledge Base contains data and strategies.
Object Class	Relational Table	Object structure is logical, not physical.
Object Instance	Relational Table Record	Represents data only, not procedures.
Object Attribute	Relational Table Field	Object attributes are inherited, not redefined.
Rule	If...Then Statement	Rule execution is not sequential.

## **How does knowledge engineering differ from conventional systems analysis?**

As discussed in Question 2, an Expert System is appropriate in situations where knowledge is fragmented, implicit, complex and not easily communicated. For this reason, the knowledge engineering task is substantially more involved than most conventional systems analysis. Expertise is not easily communicated in the form of simply stated requirements or problem-solving methods. Instead, the knowledge engineer participates in a joint process of questioning and discovery with the expert(s). Understanding evolves in "chunks" over time, and the system specifications evolve along with that understanding. The knowledge engineer plays a critical role in ensuring that knowledge is not only obtained, but also represented and structured for optimal use and reuse within the Expert System. As discussed in the next question, the responsibilities of a knowledge engineer are likely to encompass tasks, such as testing, not typically associated with a systems analyst.

## **How does the Expert Systems life cycle differ from the traditional systems development life cycle?**

As discussed in the previous question, the requirements and behavior of an Expert System are extremely difficult to state in advance during formal analysis and design stages. Instead, the knowledge, captured by the Expert System, increases in depth and scope over time through the knowledge engineering process. Rapid prototyping enables the Expert System to develop incrementally in the same manner that knowledge is obtained incrementally. Each prototype facilitates additional analysis, design and verification. Because of the overlap between these activities, the knowledge engineer is involved in virtually all phases of the Expert Systems life cycle, not just analysis.