N96- 14990

Integrated Intelligent Training And Job Aiding For Combustion Turbine Engines

Clifford M. McKeithan Jr.
Galaxy Scientific Corporation
2310 Parklake Dr, Suite 325
Atlanta, Georgia, USA
George H. Quentin, Ph.D.
Electric Power Research Institute
3412 Hillview Ave
Palo Alto, California, USA

INTRODUCTION

Over the past decade, combustion turbines have become a prominent source of electric power generation in the United States and around the world. Advanced gas turbines have offered substantial gains in firing temperatures, thermal efficiency, and electrical output. Therefore, a family of expert system diagnostic tools has been introduced to monitor their operating performance and their maintenance condition.

One particularly important feature of gas turbines is their rapid startup feature that has made these engines valuable for peaking duty, i.e. to provide short-term supplemental power when all baseload sources are insufficient to meet electrical demand peaks. Because such windows of opportunity for peaking power are brief, failure-to-start incidents must be rapidly diagnosed. This has led to the development of an expert system job aid, or startup advisor, to help technicians detect and overcome the impediment.

This paper describes an ongoing program to augment such an expert system gas turbine startup advisor, known as the EPRI SA°VANT^{tm} System, by including an intelligent training package. It will give a brief background on the SA°VANT development and an overview of its evolution into a full-blown Gas Turbine Information System (GTIS) for rapid access of on-line documentation, diagnostics, and training.

In particular, the paper will address:

(1) The conversion of the knowledge base used by the SA°VANT startup advisor so that it can be used for both training and job aiding; and
(2) The hypertext-oriented user manuals being incorporated into the system for rapidly accessing on-line documentation at the job site.

BACKGROUND OF SA°VANT SYSTEM DEVELOPMENT

EPRI began developing the SA°VANT system in 1984, initially to establish the viability of expert system job aids for field use by maintenance technicians. A simple pilot system enabled novice technicians to locate short circuits in turbine control systems in about sixty-five minutes, whereas experienced technicians without the job aid required about sixty minutes on average. Novices could not locate faults without the job aid or other support. The original computerized job aid was cumbersome, with built-in delays, and did not improve performance times for experienced technicians.

This led EPRI to develop a compact portable computerized job aid to eliminate built-in time delays and thereby expedite troubleshooting. Applying the job aid with this new SA°VANT system, the experienced technicians cut their problem solving time in half (to about twenty-five minutes). Surprisingly, the novice technicians made similar gains, solving the same problems in about twenty-six minutes.

From this baseline experience, EPRI chose to apply the SA°VANT know-how for helping technicians to diagnose failure-to-start incidents on large gas turbines in peaking service. Of the existing power generation fleet in the USA of roughly two thousand gas turbines generating 50,000 megawatts, approximately 90 percent of those engines are operated as peaking units. Generally, without special attention, starting reliability of peaking units has been at a level of 85% or lower due to a variety of reasons. By contrast, the power industry has set starting reliability goals of 95% or better.

By developing an effective startup advisor job aid, EPRI could help the power industry attain such startup reliability goals by avoiding aborted startups. This also avoids possible engine damage due to faulty restarts with its attendant hazards, as well as costly delays at a critical time.

As noted above, peaking gas turbine units supply additional power to meed peak demands, when normal baseload power plants are already operating at full capacity. Such gas turbines are often located at remote sites, far apart, not manned at all times, and are used infrequently. When such engines are needed, they must respond on short notice. This does not leave much time for troubleshooting of any problems which may occur. Since many of the problems involved in starting the turbines are not visible prior to starting the unit, and the units are not usually started until they are needed, operators and technicians must be able to rapidly troubleshoot and correct any impediments.

To compile the original knowledge base for the gas turbine startup advisor expert system, EPRI contractors gathered several gas turbine experts familiar with particular engine models, including utility troubleshooters. Using a roundtable format, the procedural expert system was generated, leading technicians carefully step-by-step from symptom to cause.

While the knowledge base was being finalized, the computer-interface though which technicians would access the expert system job aid was evolved. EPRI chose to incorporate several multimedia concepts in an audiovisual-oriented system, using drawings, pictures and video resources, as well as voice I/O. A block diagram of the system appears in Figure 1. The basic SA°VANT hardware was originally an MS-DOS compatible system based on dual 80386SX processors. Once central processing unit (CPU) controls a graphics display, while the other runs the expert system. As an alternative to the traditional keyboard/video display interface, the EPRI system included a speech synthesis/voice recognition interface to allow "hands-free" operation of the unit.

Once the expert system was developed, the SA°VANT startup advisor was field tested to determine its effectiveness. This system was tested by a broad range of technicians, ranging in experience for six months to sixteen years. The use of SA°VANT resulted in a 25% decrease in time required to troubleshoot the turbine, with an 81% decrease in calls for assistance and an 89% decrease in the number of aborted restart attempts (Quentin, 1991).

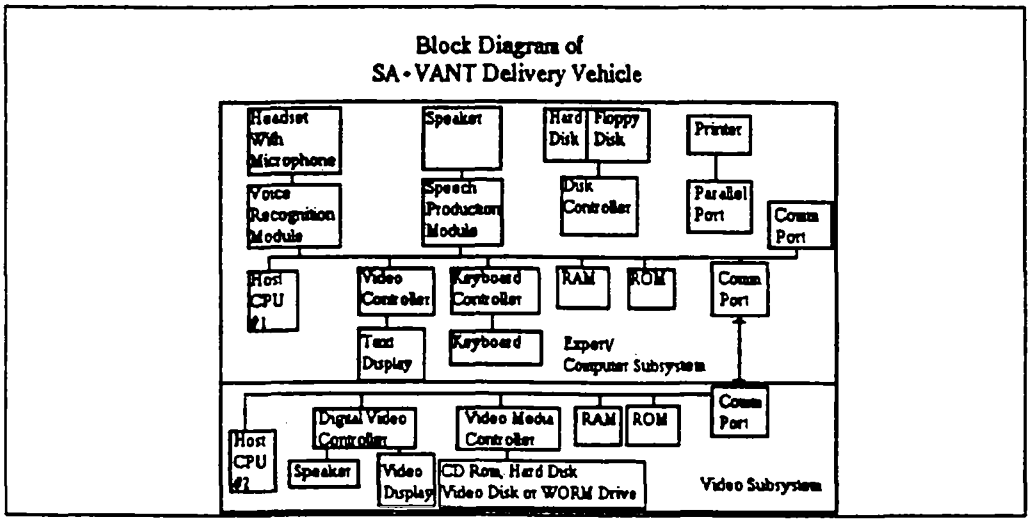

Figure 1: SA•VANT Block Diagram

THE GTIS: AN OVERVIEW

GTIS is a system composed of three major parts: an intelligent tutoring system (ITS) (Polson, 1988), a job aid, and a hypertext information manager. All three functions have been linked into a single, consistent, user environment. The GTIS runs on the same basic hardware as the SA°VANT system, with a few enhancements to both the hardware and the operating system.

Part 1: The Intelligent Tutoring System

The first major change to the SA°VANT system was the addition of the ITS. The goal of this addition to SA°VANT was to enable the technicians to practice solving troubleshooting problems during the time that they have available when the gas turbines are not being run. The hypothesis was that by practicing on a simulation-based training system, the technicians would be better prepared to troubleshoot failures within the real system when they occurred.

The simulation used by the GTIS is a surface level, static simulation. Most simulation elements maintain the same value throughout the diagnosis of a problem, with the exception of those elements that determine the ability to restart the turbine. When the trainer is started, the operator is presented with a description of the simulated failure. The operator is then free to troubleshoot the system, with the help of the intelligent tutor.

The ITS is made up of five components: the student interface, the instructional environment, the instructor model, the student model, and the expert model. These components are depicted in Figure 2.

The Student Interface

The student interacts with the ITS using a graphical user interface, depicted in Figure 3. The display consists of images and/or text on the screen. The operator can select items by pointing at the object of interest and clicking the mouse button.

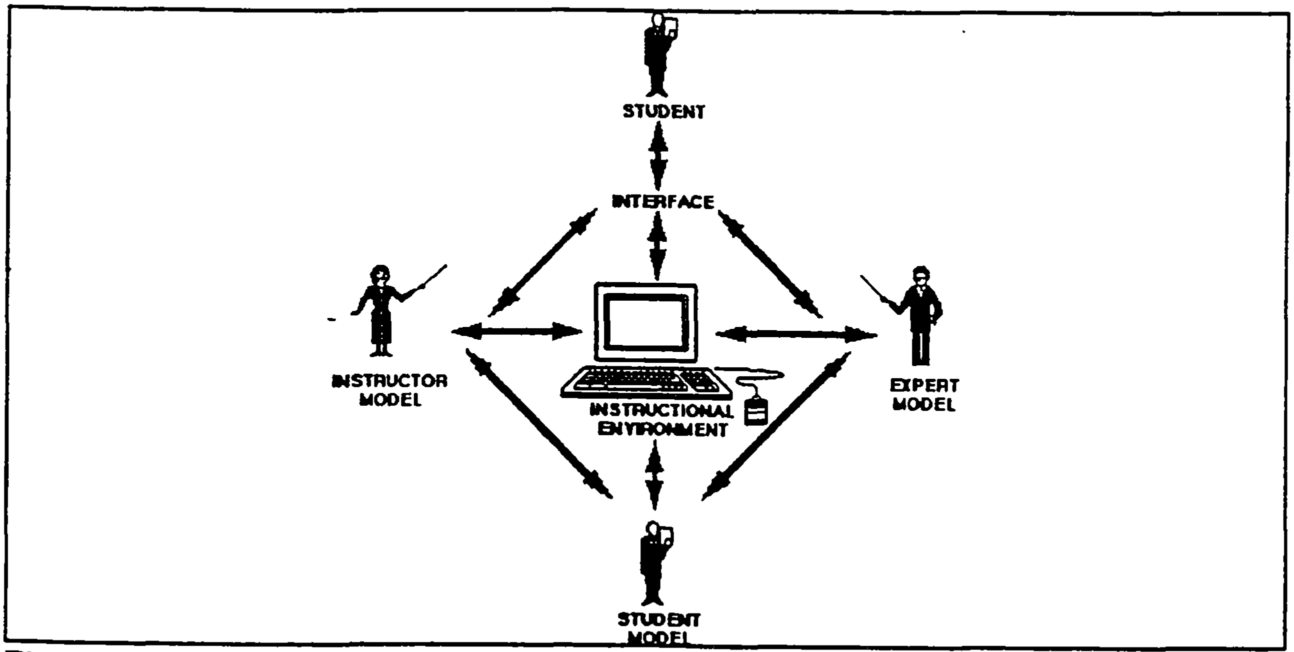

Figure 2: Intelligent Tutoring System Diagram

The Instructional Environment

As stated above, the ITS uses a static simulation as the instructional environment. In complex domains such as gas turbine engines, a combination of simulation and an ITS has proven to be effective in teaching troubleshooting skills (Johnson, 1988a). The images used in the simulation are high resolution, digitized images taken from a video recording of one of the gas turbine plants. By selecting areas on the screen that have been highlighted, the student can move from one area of the simulation to another. Within each area, the operator is able to get part descriptions, read gauges, test parts, and repair or replace parts. Each area also includes context sensitive help. The Student, Instructor, and Expert models all have access to the state of the simulation.

The Student Model

The student model tracks the actions taken by the student. This model keeps track of the number of errors made and the number of times the student has requested advice from the expert model. Access to the simulation enables it to determine which parts of the turbine have been examined, tested, or replaced. This information is used by the instructor model and the expert model to generate advice.

The Instructor Model

This model examines each action taken by the student to determine whether that action is an error. For example, if the operator attempts to replace a part that has been tested and determined to be fully functional, the instructor model would inform the student that the action is inappropriate. Any errors of this kind are logged in the student model. In addition, the Instructor Model compares each action the student takes with the recommended action from the Expert Model. This comparison allows the Instructor Model to determine if the student is progressing toward the solution. If no progress is being made, the Instructor Model advises the student to consult with the Expert Model.

The Expert Model

The expert model generates advice for the student when the student requests it. This model is passive, in that it does not give advice unless the student specifically asks for it. When advice is requested, the expert model checks the student model to see what information the student has acquired and the actions that the student has taken. It then makes use of the SA°VANT job aid knowledge base to determine the next action that the student should take to solve the problem.

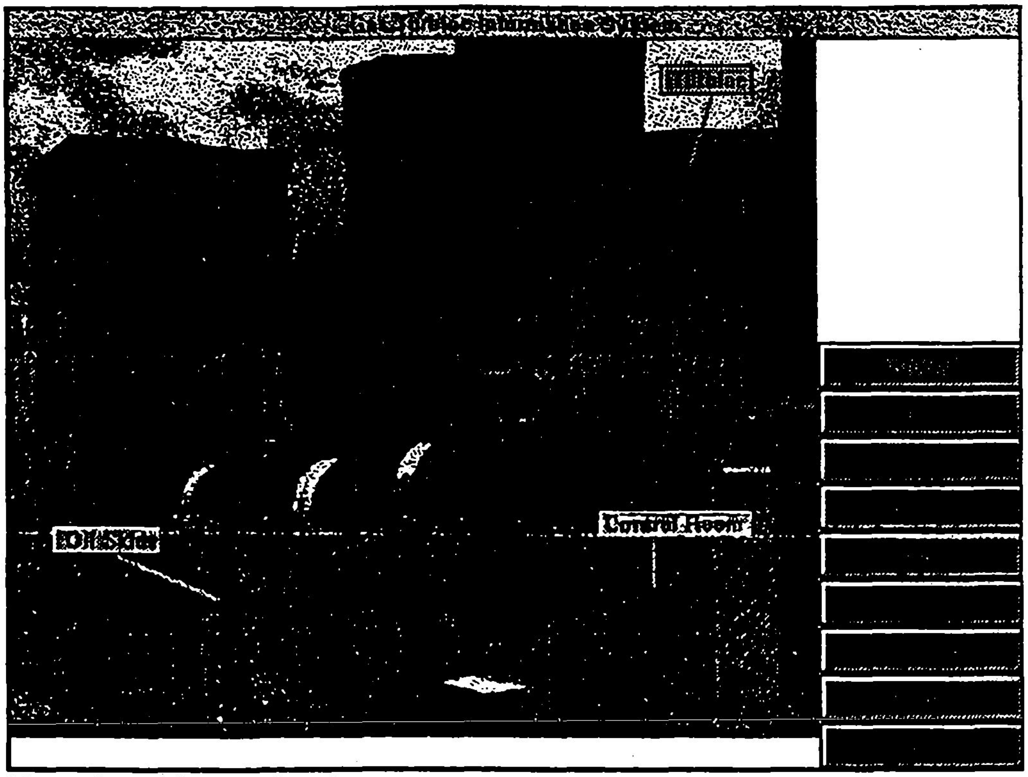

Figure 3: The Student Interface

The second part of the GTIS is the modified job aid. This consists of the SA°VANT system converted to function in the Windows environment. As stated above, the original SA°VANT interface used two CPUs. The first CPU handled the direct communication with the operator - receiving data, giving advice, supervising the speech synthesis, and recognizing voice commands. The second CPU displayed graphic images of the particular turbine part being examined. By moving the system to Windows, the need for the second CPU was eliminated. Graphic images can be displayed in another window on the same monitor that is displaying the advice, instead of using two monitors as was previously required.

Part 3: The Information Manager

The third part of the GTIS system is the information manager. The main element of this component is an on-line manual containing troubleshooting information used by the technicians when working on the job. This manual contains items such as alarm descriptions and computer access codes, and can be used both by the student when troubleshooting the simulated turbine, or by the technician in a real-time environment.

The operator also has access to information describing the parts of the turbine that make up the simulation. When the operator is examining a part, an "info" button is provided to gain access to a description of the part and its function. All the parts in the system are connected by hypertext links to enable the operator to learn of their relationships.

The final tool currently used by the GTIS information manager is digital video. Sequences of experienced technicians demonstrating troubleshooting techniques were videotaped. These sequences were then captured in digital format and incorporated into the system so that when an operator desires to test a part, he/she can view the video sequence describing that test.

It is intended that other information sources, such as operations operator manuals, will be available in addition to the troubleshooting manual. We are working on getting permission from the makers of the turbines to place their manuals on-line. When this occurs, the operator will have a powerful resource for information about the system.

REUSE OF THE SA°VANT KNOWLEDGE BASE

In developing the original SA°VANT system, a great deal of effort was invested in acquiring the knowledge to be used by the expert system. A panel of experts from several gas turbine sites was assembled to develop the rules to be used in the system. The first SA°VANT system had approximately forty-five-hundred rules compiled by this "round table" of experts. The development of these rules was both time-consuming and expensive (Quentin, 1991). Rather than trying to build a training system from scratch, EPRI decided the reuse the knowledge from SA°VANT as the basis for an ITS.

The Prototype

The approach to reusing of the knowledge base for the prototype GTIS was a simple one. The job aid knowledge was parsed "by hand" to extract sample problems which were then hard-coded into the software for the ITS. The simulation was then programmed to provide appropriate gauge readings and system settings for the gas turbine generator. Once this was completed, it was repeated for each additional problem added to the prototype. This process proved to be time-consuming in itself, as the knowledge base being parsed had several thousand rules combined into one ASCII file over two megabytes in length. It was clear that manual conversion of the knowledge base would not be appropriate.

Conversion of PML code.

The original knowledge base was written in PML, a high level, rule based description language developed by Honeywell (Cochran, 1991). This structure provides several classes of rules for building a rule based system. The classes of rules are as follows:

Tell Rule: Gives a list of other rules which will be executed in order.

Announce Rule: Provides for text output to the screen.
Record Rule: Provides for text output to a disk file.
Ask Rule: Displays a question and a list of possible answers to the screen, and then inputs the selection made into a variable.
If Rule: Compares the value of one or more variables to constant values and then executes another rule based on that comparison.
Set Rule: Sets a variable to a constant value.
When Rule: Sequentially compares one or more variables to constant values and then executes another rule based on which of the comparisons yielded a "true" result.

Each rule involving textual display is able to display two text fields and a list of selections (if appropriate).

A sample "ask" rule appears in Figure 4. Note that the rule is broken down into several sections. The "action" line gives the name for the rule. The "heading" lines allow for the insertion of comments into the code. The "ask" keyword indicates that this rule is an "ask rule", and that the following symbol, "?FLAME-SD-MACH-SPD," is the variable name into which the response is to be placed. The text enclosed in the quotes is the information to be displayed to the operator. The keyword "explain" allows for additional information to be presented to assist the operator in understanding why the question is being asked. Following the text field is a list of choices to be shown to the operator and the corresponding values to be stored in the variable.

```
ACTION- CHECK-FLAME-SD-UNIT-SPD
 DOCUMENTATION- "created by JU at 19:14:38 4/30/89
edited by BULLEMER at 22:57:51 6/5/89"
 HEADING- "5111 CHECK-FLAME-SD-UNIT-SPD *#@*"
 ASK- ?FLAME-SD-MACH-SPD
(FORMAT NIL "
What was the speed of the unit at ~
the time of the flame shutdown?
")
 (EXPLAIN
(FORMAT NIL
 "
If you did not reach 1200 rpms, ~
the unit probably tripped after the ~
ignition sequence timer expired.

Note: 1200 rpm is 30% of full unit ~
speed. ~
"
 ))
 ("Unit speed < or = 1200 rpm" :Unit-speed—or---1200-rpm
 "Unit speed > 1200 rpm"  :Unit-speed---1200-rpm
 "Don't know unit speed"  :Don-t-know-unit-speed)
```

: Sample "Ask" Rule

Accessing a specific rule requires either a time consuming textual search, or an indexing scheme that indicates the rule's position in the file. If the latter scheme is used, any changes to the knowledge base would require recomputing the values in the index. It was decided that in the interest of data access and modification, the knowledge base would be converted to a database. This would allow for direct access to the rules based on the rule name.

To facilitate this conversion, the tools LEX (a lexical analyzer) and YACC (Yet Another Compiler Compiler) were used along with a C++ program written for this purpose. LEX was used to analyze the PML code and break each rule down into its respective parts. YACC was used to take the values for these parts and send them to the C++ code. This code in turn organized the rules and saved them into the database. For example, the LEX generated code would locate the keyword "ACTION" in the sample above. YACC would then parse the action name "CHECK_FLAME_SD_UNIT_SPD" and call a procedure in the C++ program which would make a new record in the database for the new action. The remaining fields in the database record would be filled as the rest of the rule is parsed.

Designing the database records involved tradeoffs between complexity, space, and operator requirements. The simplest design for the records would have been to create one record with a field for every possible kind of rule. However, this would have wasted a great deal of space, as most of the records have only a small subset of the total possible fields. A second option involved a more complex database structure which greatly reduced the disk space required to store the record. As this system was being developed, an additional operator requirement forced another change in the database structure. The operators responsible for maintaining and updating the PML code wanted to be able to edit the rules in their original form. As a result of this need, the current form of the database was developed. Each rule is stored separately in the database as a single field, which then has to be parsed using LEX and YACC at runtime to break it into its component parts.

In order to navigate through the rule base, the calling program only needs to initially know the name of the starting rule. It can then recall this rule from the database and begin to parse it. The starting rule is generally a "tell rule" which yields a stack of additional rules to be interpreted. The rule currently on the top of the stack is then loaded from the database and is parsed and executed in its turn.

Once the rules were converted to the database, a tool was written in C++ to allow for the editing of the text fields in the database. This tool allows the operators to maintain the knowledge base without having to manipulate the database directly, and provides both a graphical interface and a text-based interface to edit the PML code.

CONTEXT SWITCHING

The GTIS provides the operator the ability to switch at any time between the job aid and the tutor. This allows the operator to access the on-line manuals, parts descriptions, equipment descriptions and diagrams, and any other information that the system provides. Thus the operator can get any additional information about the gas turbine that may be required for troubleshooting. It also furnishes a way for the operator to investigate why the job aid has asked for the particular data that it needs. Such context switching allows the operator to learn about the gas turbine without interrupting the job aid.

CURRENT DEVELOPMENTS

There are several extensions and enhancements which are being added to the GTIS as a result of the latest round of developments.

The first extension to the system is increasing the coverage of the ITS. At this time, only a few of the failures covered by the SA°VANT system are handled by the ITS. Now that the knowledge base has been converted to the database format, additional failures are being incorporated into the GTIS. In addition, more digital video sequences are begin incorporated into the system.

As the system continues to develop, we will be adding tools for creating and editing new turbine information systems. EPRI has generated knowledge bases on several different styles of gas turbine generators. The GTIS only addresses one of these knowledge bases at this time. With these conversion tools, all of the knowledge bases can be converted to database format and added to the GTIS.

A number of hardware improvements have been made to the system. The system has been modified so that it will function when running on a pen-based computer platform. The current SA°VANT system is a large unit, which cannot be easily carried from one location in the plant to another. This isä a significant problem if the unit is being used as a job aid. The technician must go to the console to get instructions, then leave the console to carry out the instructions. Once the operator has performed the required task, he/she must then return to the console and repeat the process. The current pen-based computer models are much smaller - about the size of a paper notebook. At less than four pounds, they are also relatively light. Porting the GTIS to a pen-based computer will allow the operator to carry the computer while performing maintenance. He/she will no longer have to go continually back and forth between the turbine and the computer, thus reducing the amount of time required to complete each task.

The enhanced portability of the pen based system has benefits for training as well. By bringing the tutor out to the turbine site, the student will be able to more easily relate the task being shown on the computer to the task required on the turbine.

CONCLUSIONS

The GTIS employs the same knowledge for use in both training and job aiding. As tools for knowledge base management improve, the ability to revise existing job aiding knowledge for other tasks becomes feasible. By converting these knowledge bases instead of recreating them, developers will realize a significant savings both in cost and time in the development of training systems. By consolidating information into a common knowledge base source for the GTIS, it should be feasible to create an overall system that appears "seamless" to the user. That is, the user should be able to move freely among the three elements in the system (i.e. documentation, diagnostics, and training) as needed, without necessarily crossing "hard system boundaries."

REFERENCES

Quentin, G.H., and Dolbec, A.C. (1991). "Expert Systems for Peaking Combustion Turbine Plants," ASME, Budapest, Hungary

Polson, M.C., and Richardson, J.J. (Eds.) (1988). Foundations of Intelligent Tutoring Systems. Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Johnson, W.B. (1988a). Developing expert system knowledge bases for technical training. In L.D. Massey, J. Psotka, and S.A. Mutter (Eds.), Intelligent Tutoring Systems: Lessons Learned (pp 83-92). Hillsdale, NJ: Lawrence Erlbaum Associates, Inc.

Cochran, T., Bloom, C., and Bullemer, P. (1991). Increasing End User Acceptance of Expert Systems by Using Multiple Experts. In K. McGraw and C. Westphal (Eds.), Readings in Knowledge Acquisition (pp 73-89). London, England: Ellis Horwood