Watson, I.D., Basden, A., & Brandon, P.S., (1992). A Client Centred Approach to the Development of Expert Systems. Expert Systems, Vol. 9 (iv): pp.181-88.

A Client-Centred Approach to the Development of: Expert System

Ian Watson*, Andrew Basden†, & Peter Brandon*

*The EDESIRL Project
Department of Surveying
University of Salford
Salford, M5 4WT, UK.
☎ +44 61-295-5227 🗎 +44 61-295-5011
I.WATSON@CONSTRUCT-IT.SALFORD.AC.UK

†Information Technology Institute
University of Salford
Salford, M5 4WT, UK.

Abstract

This paper describes a project that is studying an expert system (ES) development method tailored to meet the requirements of small and medium sized enterprises (SMEs). It is posited that many such organisations cannot use ES technology because of its high entry price and because the available tools and methods were developed for and by large organisations. Within this context our development method encompasses the entire life cycle of an ES from initial conception to implementation through to maintenance and updating. The central premise is that the method should be "client-centred," emphasising "what" the client can see rather than being "technology-centred," emphasising "how" the knowledge engineers work. Discussion includes how this methodology addresses many criticisms of current ES development methods including rapid prototyping and KADS.

INTRODUCTION

This paper describes the aims, objectives, and findings of a collaborative research project funded under the IEATP between the Department of Surveying and the Information Technology Institute at Salford and headed by the Royal Institute of Chartered Surveyors. It is concerned specifically with the Evolutionary Development of Expert Systems in "Real Life" (EDESIRL).

The project has started from the observation that many current expert system (ES) development methods are "technology centred" and are not suited for use by small and medium sized enterprises (SMEs — defined by the DTI as an organisation with less than 501 hundred employees). While a few off-the-shelf ESs are available, most were designed for and by large companies and are thus not suitable in the different contexts of SMEs. Moreover, few SMEs can capture their expertise in an ES due to the relatively high price of hardware, software, and consultancy.

The project also recognises that there has been a proliferation of ES development methods, but no single method can be appropriate in all situations. We therefore suggest a contingent approach that enables practitioners to use their own techniques and tools within a guiding framework. Our approach is similar to those being generated by research into information systems development [Avison & Wood-Harper, 1991].

The EDESIRL project involves approximately twenty-five SMEs and is investigating the requirements of SMEs that want to develop ESs. It intends to provide easier entry to the ES arena for SMEs by providing a contingent development approach tailored to their needs. This will help the spread of good practice and professional expertise throughout SMEs in Europe.

A "method" is understood as a system of principles and techniques brought together in fulfilment of a unifying goal: the production by SMEs of usable ESs that fulfil a beneficial role in the organisation concerned. In this project the context is that of the construction industry, particularly surveying.

The EDESIRL project has the following goals:

- to generate a model of the dynamics of the evolutionary development of ESs within SMEs,
- to develop a method to guide evolutionary development of ESs for SMEs, and
- to propose techniques based on the method,

There are two main elements to the method:

- A life-cycle method (LCM), concerned with managing the whole life of the ES, from its initial conception, its development, its bringing into use, its development in use, and eventually its demise as a useful tool.
- A knowledge acquisition method (KAM), that involves obtaining a structured and understandable expression of the expertise that can be encapsulated in the computer to form the ES.

Though this project is developing tools to aid the method, the techniques and the functionality of the tools are of primary importance. Moreover, we do not intend to "reinvent the wheel"; thus, when an existing tool exhibits the required functionality we shall use it providing we can integrate it sufficiently with other software.

This paper discusses the LCM and the KAM. It then outlines the functional requirements of intermediate representations, and concludes with proposals for future work.

LIFE CYCLE METHOD

The LCM developed by this project began with Basden's Client-Centred Approach (CCA) [1989]. The CCA has been used successfully in industry and is not merely an ad-hoc collection of experience. It is based on several principles:

- the importance of spending time and effort at the early stages of an ES project in defining roles, benefits, objectives, and so forth;
- the importance of a holistic approach throughout the project, including the consideration of usability and saleability from the start;
- the recognition that functioning software is useful both as a discussion point, and for deriving a specification;
- the need for a staged approach to project progression so that clients can plan resource provision;
- the need for an iterative approach to knowledge acquisition and representation because of the well known problems of devising a specification in ill-structured areas;
- the close involvement of the client and other stakeholders throughout the process;
- the need to speak the client's language;
- and the usability of the methodology by lay people (i.e., those within a SME who may have some experience of using spreadsheets or databases but are not IT professionals).

"Client" means the person or organisation for whom an ES is being built; the client owns the problem that motivates an ES project and provides or authorises resources for the project. "Stakeholder" refers to any person who is or will be affected by the creation and use of an ES, and there can be many of these including:

- clients as above,
- knowledge engineers,
- primary users who press keys or mouse buttons,
- those who supply information to enable the primary users to operate the ES,
- those who use information derived from it,
- managers of the primary users,
- those responsible for software and hardware support,
- experts from whom the expertise is acquired to building an ES,
- customers of organisations,
- suppliers to organisations,
- and the wider society and environment.

To this end the method proposes a seven-stage structure to ES projects, from initial conception through to the ES embedded in use within its proper organisational and human context (see Figure 1). The middle stages concern the conventional activities of knowledge acquisition, representation, and validation.

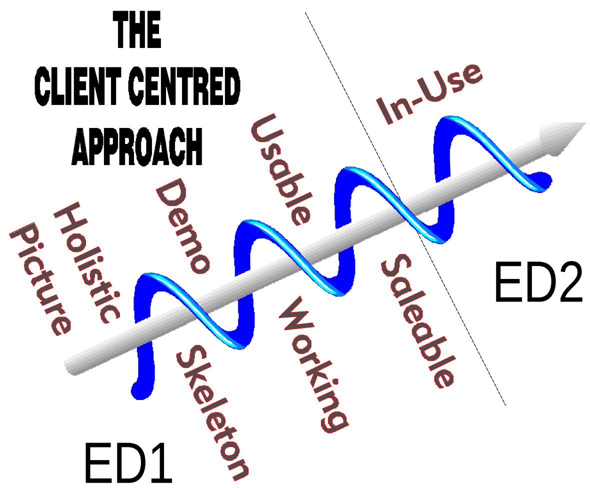

Figure 1.

Though it is a staged structure, the CCA overcomes problems normally associated with linear structures because it allows for evolutionary development. Moreover, it overcomes problems normally associated with rapid prototyping. These problems have been identified by Berry and Broadbent [1986] and Diaper [1989] as follows:

- Systems may satisfy the expectations of knowledge engineers and domain experts but not necessarily those of the users.
- Prototyping can lead to uncontrolled growth of systems causing difficulties in maintenance.
- Because elicited knowledge is often translated directly into code, there is no complete and explicit statement of knowledge included in systems. This also makes maintenance very difficult.
- Over reliance on iteratively refining ESs can make them unable to solve problems occurring only rarely in the domain [Roth & Woods, 1989].

- The final "delivery" system is often a tidied up version of the prototype that was being worked upon when either the project's time or money ran out.
- Because of its cyclical nature prototyping is very difficult to audit.

However, although Diaper recognises that "people" should be at the centre of a method, he does not identify the cause of the above problems, namely that they are derived from being technology centred [Basden, 1989]. Both linear and rapid prototyping methodologies are described in terms dictated by the technology (e.g., elicit, represent, debug), and many omit to establish an initial holistic view of the project. Thus, the stakeholders are not fully involved and the above problems result.

The criticism of being technology centred also may be levelled at KADS, which abounds with technical jargon describing activities in terms of layers of knowledge and models. Such an approach will inevitably exclude the stakeholders unless they are willing to undergo extensive training.

The EDESIRL project uses the CCA to address these problems by maintaining a client-centred rather than a technology-centred focus. In the CCA the stages (see Figure 1) are named not after activities ("how") but after deliverables ("what") that the client can expect throughout the project. This ensures that a holistic client-centred view is established initially, resulting in closer co-operation with all stakeholders throughout the project rather than just at the early and the late stages.

While it might appear that these stages are an artificial way of breaking up incremental development, they have their own distinct purposes, techniques, and deliverables.

The Holistic Picture

The purposes of the first stage are as follows:

- to perform a feasibility study that includes business benefit, limitations and risk;
- to scope the ES domain;
- to identify all stakeholders;
- to obtain a holistic picture of a problem by involving stakeholders;
- to obtain their commitment to the project;
- to help the client plan resource provision; and
- and to provide retrospective documentation of top-level decisions.

Because our approach is contingent, various methods can be employed here, including aspects of Checkland's Soft Systems Methodology [1989] or Mumford's ETHICS [Mumford, 1986] The deliverables at this stage are documents giving a clear, holistic picture of the objectives of the proposed ES, its human and organisational environment, its benefits and limitations, its acceptability, its interface, its eventual market niche, and its knowledge sources. An analysis of potential threats to the success of the project, using Boehm's spiral method [Boehm 1986 & 1989] are prepared along with a project plan including resource allocation. However, we recommend an alternative view of the spiral model as discussed below and in greater detail in Watson [1992].

Skeleton System

This shows a sequence of screens to give stakeholders a "feel" of the systems' overall functionality. It contains little domain knowledge, but serves the following purposes:

- to gain a clear picture of the overall functionality of the proposed ES including its inputs and outputs,
- to ensure stakeholders understand what ESs can be expected to do,
- to help knowledge engineers obtain an initial understanding of the domain and of the types of information that is handled, and in what sequence, and
- to obtain resource commitment.

The deliverable is an embodiment of the conventional top-level functional description, but since it is active, the stakeholders (who may have little idea of what to expect from ESs) get a better "feel" for the system than they would from a paper specification. Some domain conceptualisation occurs as a result, which can form the basis for the next stage. The project has prepared a software tool to guide this stage.

Demonstration Systems

In the third stage a significant amount of knowledge is acquired and encapsulated in an ES that can be demonstrated. Most domain conceptualisation and structuring is completed, as is the functional specification and design. There may be several demonstration systems; their purpose is as follows:

- to keep the client informed and involved;
- to provide the knowledge engineers with an understanding of the types of knowledge in the domain;
- to decide upon suitable representations and software, and
- to provide a go/no-go decision point: Will the ES as proposed bring relevant benefits? Can the knowledge be encapsulated in an ES?

The methods employed in this stage are knowledge acquisition (as described below in the KAM), knowledge representation, and validation but as a cyclical rather than as a linear process. (The method adopted successfully on the ELSIE project [Brandon, et al., 1988] employed several experts for knowledge acquisition and others for validation.). The holistic picture obtained during the first stage ensures that individual parts of the structure are always related to the whole, but this picture may evolve, once experts and several primary users start to use the system.

Working System

While the final Demonstration System gives reasonable results in unexceptional conditions, the Working System gives "correct" results in "all" situations in which the ES is expected to be used. The techniques employed are the same as in the third stage, but validation assumes ever greater importance as the project continues. Sometimes parts of the knowledge base must be reworked during this stage, owing to an over-simplified picture having been obtained earlier. This becomes clear as experts use the ES and highlight deficiencies. In our experience it is seldom possible to predict all such requirements and knowledge structures in advance.

Usable System

The Working System gives correct results but is too cumbersome in its use to bring business benefit. This stage ensures that the ES is usable, by considering usability features such as "what-if," explanation facilities, easy methods of data entry, and integration with other systems. It is advised that knowledge acquisition and validation are used to decide what usability features are needed. Too often usability features are approached as a set of features to be added to a system to make users happy. However, we suggest that the determination of the type and structure of these features is worthy of knowledge acquisition.

The deliverable is a Usable System. Note that this stage may actually overlap with stages three and four to some extent. For instance, usability features like "what-if" and "dump-and-restore" can be useful to streamline the validation process. These should be included at carefully chosen points so that the client receives visual indication of progress instead of reports that, "The accuracy has increased from eighty five to ninety per cent." This stage is deliberately placed after stage four to emphasise the overriding importance of getting the core knowledge right. This is discussed in more depth in Watson et al.[1991].

Saleable System

The purpose of this stage is to prepare the ES for wide distribution to those who might be less sympathetic to it. This involves attending to details of the interface, the wording of screens, a help system, tutorial, user documentation, installation procedures, and preparing for the launch and subsequent activity. The deliverable is the final ES.

Note that giving too much effort prematurely to saleability features is a common mistake since developments can render them redundant. However, saleability and such things as documentation should be kept in mind throughout the project, and we intend that user documentation should follow the ES in parallel stages from Skeleton to Saleable.

Embedded in Use

The seventh stage brings the ES into regular beneficial use. This involves training users, providing help-lines and so forth, but it also means earlier planning of organisational changes required for effective use of the ES. The timing of changes will have been determined during the first stage. The deliverables at the seventh stage are the business benefits that accrue from using the ES.

KNOWLEDGE ACQUISITION METHOD

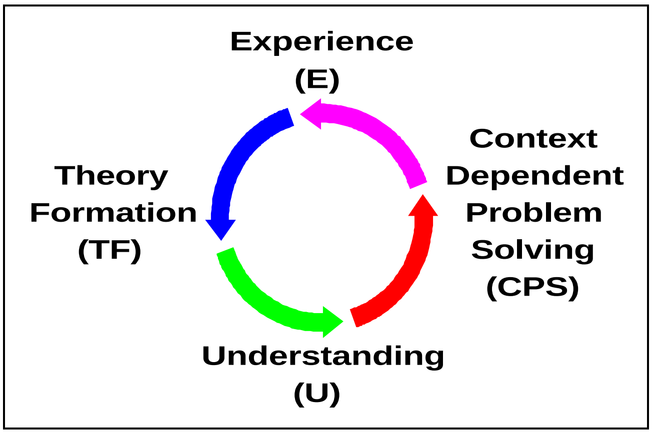

Figure 2. A Model of Experience & Understanding

The method the EDESIRL project uses differs from many other knowledge acquisition methods in that it is based on an explicit model of experience and understanding (shown in Figure 2, and discussed in detail in [Attarwala & Basden, 1985]). This model has been used on several successful ES projects [Jones & Crates, 1985; Hines & Basden, 1986; Brandon, et al., 1988]. It states that it is insufficient merely to elicit problem solving heuristics because these are subjective and context sensitive, being composed of underlying understanding (*U*) and Context Dependent Problem Solving (*CPS*). *CPS* includes:

- the problem solver,
- the problem context, and
- the problem solving method.

The knowledge acquisition process should separate the *U* from the *CPS* element. This is achieved through three activities:

- obtain expert problem solving heuristics (using well known elicitation techniques);
- seek the *U* that lies behind the heuristics,
- ask the question, "Why?" in several ways; and
- make explicit the *CPS* that is implicit in the heuristics, by asking the questions, "What else?" and, "When not?" in several ways.

These activities are not necessarily performed in the above order; they are performed as necessary, with the second and third activity forming the bulk of the work. This method has been found to overcome several problems endemic with ESs. It reduces fragility of the knowledge base and improves knowledge base maintenance and explanation of results due to in-built understanding. It also turns disagreement between domain experts into high quality knowledge.

INTERMEDIATE KNOWLEDGE REPRESENTATIONS

Because of the problems inherent with prototyping, the requirements of maintainability (i.e., ED2), and quality assurance, this project is researching intermediate knowledge representations. Many researchers advocate using an explicit and complete representation intermediate between the expert's knowledge and the implemented system code [Alexander et al., 1986; Wielinga & Breuker, 1986; Berry & Broadbent, 1986; Butler & Chamberlin, 1987; Edwards, 1987; Johnson, 1987 & 1989; Recogzei & Plantinga, 1987; Young, 1987; Diaper, 1987, Watson et al., 1989; Kalos, 1992].

The functional requirements of an ideal intermediate knowledge representation are as follows [Watson, 1989]:

- logically testable for consistency;
- speed of production of the intermediate representation;
- clarity to experts, so they can validate it;
- direct usability as system documentation to assist ED2;
- high expressive power (flexibility);
- translate into program code ideally by automatic means; and
- integration with a LCM.

Functionality / Representation	Logically Testable	Speed of Production	Clarity to Experts	Flexibility	Translation into code	Documentation	Life Cycle Support
The Perfect Intermediate Representation	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4	4 4 4 4 4
Rapid Prototyping	8	4 4 4 4 4	8	4 4 4 4 4	8	8	4 4 4 4
Paper Models	8	4 4 4	4 4 4 4	4 4 4 4 4	8	4	4 4
Formal Notations (e.g "Z")	4 4 4 4 4	4 4	8	8	4 4 4 4	4 4 4 4	4 4 4 4
Conceptual Graphs	4 4 4 4 4	8	4	4 4 4 4	4 4 4	4 4 4	4 4
The KnAcq	4 4 4 4 4	4 4 4 4 4	4 4 4	4	4 4 4 4 4	4 4 4	4
Inference Nets	4 4 4 4 4	4 4 4 4	4 4 4 4 4	4 4 4	4 4 4	4 4 4	4 4 4 4

Table 1. A Functional Comparison of Intermediate Representations

Table 1 presents the functional requirements of the ideal intermediate representation and compares this with several alternatives:

- Formal notations (e.g., "Z") score highly on testability and documentation, but their ability to represent different knowledge types (their flexibility) and their clarity to domain experts is poor.
- Prototyping is not a representational formalism, but it is included in Table 1 because many researchers and methods support it. It scores highly for speed (hence Rapid Prototyping) and flexibility (because there is no syntax, it is infinitely flexible).
- Paper Models have been used by many researchers and are treated here as a generic term for any method of representing domain knowledge on paper. This could either show simple overviews or considerable detail. Because, these are on paper they cannot be logically tested. They usually can be created quickly and are also infinitely flexible; however, they are not usually supported by procedures for creating program code from them, and consequently they do not form reliable system documentation. However, they can be supported by life cycle methodologies.
- Conceptual graphs [Sowa, 1984], which evolved from Entity Relationship diagrams [Chen, 1976], can be tested, are sufficiently flexible, make good documentation of the knowledge in the system, and can be understood by domain users (with some tuition) [Watson et al., 1989; Watson, 1989]. However, their construction is time consuming, and they are not supported by a LCM.
- The KnAcq™ is representative of knowledge acquisition methods and tools that are technology-centred [Mott & Brooke, 1987; Brooke & Jackson, 1991]. The KnAcq is testable, expedient and can generate program code for several commercially available ES shells. However, it only handles certain types of knowledge, is not supported by a life cycle method, and can only serve as documentation for the knowledge in the ES.
- Inference nets were used successfully on the ELSIE project [Brandon et al., 1988], and domain experts find them flexible and easy to understand. However, on the ELSIE project they were then simply an example of another paper model, and were therefore slow to create, untestable, and not directly translatable into code. We have subsequently implemented a prototype tool for testing sets of inference nets [Watson & Norman, 1992], and have implemented an object oriented architecture (called the Agenda Manager) that supports the implementation of inference nets into the ES shell Kappa-PC™.

DISCUSSION

The description above is of the method as it is now, and the EDESIRL project's purpose is to research and develop this. In particular, attention will be focused on the following issues:

Risk

Some project management techniques such as Boehm's spiral model [Boehm, 1986 & 1989] and that used by KADS [Breuker & Wielinga, 1985; Bright et al., 1991] are couched in terms of risk. Risk handling is implicitly incorporated in the CCA in three ways:

- obtaining a holistic picture of the project at the start reduces the risk of failure;
- the iterative, but managed, approach to building the ES reduces risk due to changing requirements; and
- the KAM uncovers underlying understanding and context that minimises the risk of ES fragility [Hart, 1988].

While we do not believe that KADS is always suitable for smaller organisations (this view is accepted by Professor Wielinga of the KADS II consortium), their project management techniques, and in particular their handling of risk, are appropriate. Their project management method is based upon Boehm's spiral model and provides a useful tool for managing the complexity and unpredictability of an ES project of any size [Bright et al., 1991].

However, criticism of the depiction of a project's development as a spiral (i.e., implying to clients that the project is going round in circles) has caused us to represent development as a helix.

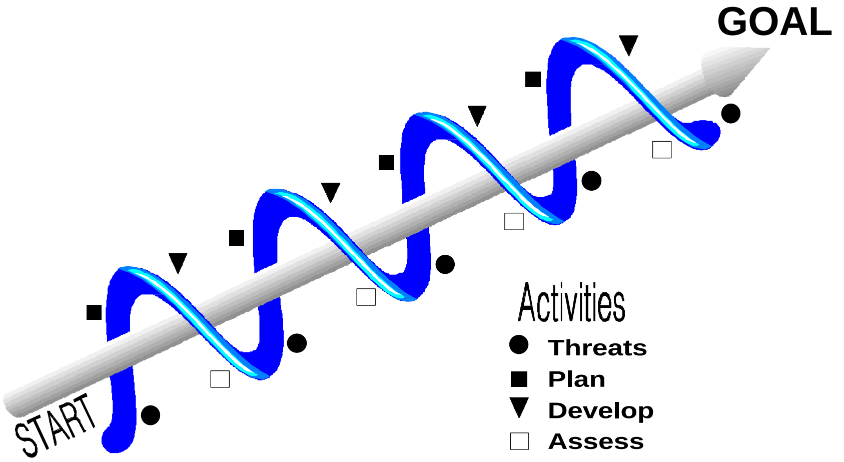

Figure 3. A Helical Model of Risk Management Based on the Spiral Model

At one level this is just a cosmetic change, but we feel it important that the client sees that the project is progressing towards a goal as shown in Figure 3. We recommend for smaller projects that one cycle of assessment, risk analysis, planning and development is carried out for each stage of the CCA. Larger projects could include more cycles per stage as necessary. The helical model is described in more detail in Watson [1992].

Metrics

Neither part of the method discusses metrics (measuring the accuracy, acceptability, impact, costs, etc. of the ES) because the method is designed to accommodate any metric approach or method. However, it is recognised that work is required in this area, and it is noted that another IEATP project is specifically involved with ES Metrics, which we do not intend to duplicate.

Quality Assurance (QA)

QA is a vogue topic, so it may be surprising that it is not explicitly included as an issue in the method. The reason is perhaps historical in that the roots of the method lie in the era just before the upsurge in interest in QA. The whole aim of the method, however, is identical with that of many QA approaches, and it has been based on similar principles. In particular we believe that the uses of an intermediate representation that can be tested and translated into code by a verifiable process assists QA. Also of importance is the validation (or endorsement) of the knowledge included in an ES by a recognised authority (e.g., a professional institution). In support of this a recent study of the users of the ELSIE system showed that its endorsement by the Royal Institution of Chartered Surveyors was the key factor that influenced people to buy the system [Castell, et al., 1992].

Documentation

This comprises three separate types, each of which is fully specified at each stage:

- Project documentation includes the details of the feasibility study, and all decisions taken during each stage of the project and management cycles.
- System documentation includes the scope and functionality of the system, archived elicited knowledge, the intermediate representation and functional design specification, and the system code itself.
- User documentation includes all topics one would expect in a comprehensive user manual. The project is preparing a template to represent "best practise" in this area. Owing to the importance of user documentation and the frequency with which it is inconsistent with a product, the method includes explicit guidance on documentation being produced as the ES is built.

ED2

The EDESIRL project has developed a maintenance method for ED2 which is compatible with that used in ED1 and is based on the same principles as those of ED1. We were fortunate in having two commercially available ESs to study, and a forthcoming paper will describe our findings [Watson et al., 1992]. The ESs developed during this project also will undergo some ED2 before the end of the project.

It is important to recognise that maintenance considerations should be built into an ES during ED1. These include good project and system documentation, clear functional design and modularity, and the continual involvement of clients, experts, and users.

CONCLUSION

This project is exploring our understanding of the development of ESs within SMEs. We believe that two underlying problems of currently available methods and tools are that they are "technology centred" and prescriptive. The approach that forms the basis for this project has been shown to work in practice, having been used in the building of three successful ESs to date, and explicitly addresses these problems. It comprises two parts:

- a Client-Centred life-cycle method that places clients at the centre of the development process and informs them in terms of "what" they can see rather than "how" it was achieved;
- an approach that is "contingent," thereby enabling people to use techniques and tools that they trust within a guiding framework; and
- a knowledge acquisition method that obtains the understanding that informs heuristics.

However, the method requires development. Areas such as tool selection, metrics, and quality assurance, must be incorporated. The method seeks to combine the strengths of both the software engineering linear approach and the iterative approach of AI while overcoming many of their weaknesses. Finally, the principles of the method must be fully applied to ED2, while the information gathered from studying ESs in use must be re-incorporated into ED1.

ACKNOWLEDGEMENTS

This research was funded by the IEATP Program, Project No.IED4/1/2062. We would also like to acknowledge our collaborators: The Royal Institution of Chartered Surveyors, Inference Europe Ltd., and Imaginor Systems, and our project user group.

BIBLIOGRAPHY

Alexander, J.H., Freiling, M.J., Shulman, S.J., Staley, J.L., Rehfuss, S., & Messick, S.L. (1986). Knowledge Level Engineering: Ontological Analysis, in AAAI-86 vol.2: pp.963-968.

Attarwala F.T., & Basden, A. (1985). A methodology for constructing expert systems, in R & D Management, 152: pp.141-149.

Avison, D.E., & Wood-Harper, A.T. (1991). Information Systems Development Research: An Exploration of Ideas in Practice. The Computer Journal, Vol.34(ii): pp.98-112.

Basden, A. (1989). A Client Centred Methodology for Building Expert Systems, in People and Computers, V. Sutcliffe, A., & Macaulay, L. (Eds.), Cambridge University Press, Cambridge, UK.

Berry, D.C., & Broadbent, D.E. (1986). Expert Systems and the Man-Machine Interface, in Expert Systems, Vol.3(iv): pp.228-231.

Boehm, B. (1986). A spiral model for software development and enhancement. IEEE Computer, May 1988.

Boehm, B. (1989). Software Risk Management. IEEE Computer Society Press.

Brandon, P., Basden, A., Hamilton, I., & Stockley, J. (1988). The Strategic Planning of Construction Projects, RICS Publications, London, UK.

Breuker, J.A, & Wielinga, B.J. (1985). KADS: Structured Knowledge Acquisition for Expert Systems, in Proc. 5th. Int. Workshop on Expert Systems & Their Applications.

Bright, C., Martil, R., Williams, D., & Rajan, T. (1991). The Kads-II Framework for ES Project Management. Proc. Ist SGKBS Int. Workshop on Knowledge-Based Systems Methodologies.

Butler, A., Chamberlin, G. (1987). The Aries Club -- Experience of Expert Systems In Insurance & Investment, in Research & Development in Expert Systems IV, pp.246-257, Moralee, D.S. (Ed.), Cambridge University Press, Cambridge, UK.

Castell, A.M., Basden, A., Erdos, G., Barrows, P., & Brandon, P.S. (1992). Knowledge Based Systems In Use: A Case Study, in Proc. of the 12th. Annual Conference of the British Computer Society Specialist Group on Expert Systems (in print).

Checkland, P. (1989). Soft Systems Methodology. Human Systems Management, 8: pp.273-89.

Chen, P.P. (1976). The Entity-Relationship Model -- Toward a Unified View of Data, in ACM Transactions on Database Systems 1(i): pp.9-36.

Diaper, D. (1989). Knowledge Elicitation: Principles, Techniques & Applications. Ellis Horwood Ltd.

Hart, A. (1988). Expert Systems: an introduction for managers. Kogan Page

Hines, J.G., & Basden, A. (1986). Experience with the use of computers to handle corrosion knowledge, in British Corrosion Journal, 21(iii): pp.151-56.

Johnson, N. (1987). Mediating Representations in Knowledge Elicitation. In, Proc. 1st. European Workshop on Knowledge Acquisition for Knowledge Based Systems. Addis, T., Boose, J., & Gains, B. (Eds.).

Johnson, N. (1989). Mediating Representations in Knowledge Elicitation, in Knowledge Elicitation: Principles, Techniques & Applications, pp.179-194, Diaper, D. (Ed.), Ellis Horwood Ltd., Chichester.

Jones, M.J., & Crates, D.T. (1985). Expert Systems and Videotext: an application in the marketing of agrochemicals, in Research and Development in Expert Systems - procs. of the 4th. Technical Conference of the British Computer Society Specialist Group on Expert Systems', Bramer, M.A. (Ed.), Cambridge University Press, Cambridge, UK.

Kalos, A. (1992). Knowledge Engineering methodology for Industrial Applications. In, Proc. of EXPERSYS-92, Niku-Lari, A. (Ed.), pp.593-98.

Mott, P., & Brooke, S. (1987). A Graphical Inference Mechanism. Expert Systems, 4(ii)

Mumford, E. (1986). Using Computers for Business Success: The ETHICS Method. Manchester Business School.

Regoczei, S. & Plantinga, E.P.O (1987). Creating the domain of discourse: ontology and inventory, in J. Man Machine Studies, 27: pp.235-250.

Roth, E.M., & Woods, D.D. (1989). Cognitive Task Analysis: An Approach to Knowledge Elicitation for Intelligent System Design, in Topics in Expert system Design: Methodologies & Tools. Guida, G., & Tasso, C. (Eds.), pp.233-264. North-Holland, New York, US.

Sowa, J.F. (1984). Conceptual Structures: Information Processing in Mind and Machine. Addison-Wesley Publishing Co. Inc., Reading, Massachusetts, US.

Watson, I.D. (1989). A Methodology for Knowledge Analysis, Ph.D. Thesis, University of Liverpool, UK.

Watson, I.D., Shave, M.J., & Moralee, D.S. (1989). A Knowledge Analysis Methodology Using an Intermediate Representation Based on Conceptual Graphs, in Proc. 9th. Int. Workshop on Expert Systems & their Applications, 1: pp.183-98.

Watson, I.D., Brandon, P.S., & Basden, A. (1991). The Development of Knowledge Based System for Predicting Building Maintenance, in Microcomputers in Civil Engineering, special issue on Artificial Intelligence and Construction, 6**:** pp.303-10.

Watson, I.D. & Norman, M. (1992).
ProTest: A Knowledge Analysis Tool for the Development of Expert System*s***.** In, Proc. 4th. International Conference on Software Engineering and Knowledge Engineering, IEEE Computer Society Press.

Watson, I.D. (1992).
Project Management for the Evolutionary Development of Expert Systems. . In: Research and Development in Expert Systems IX, Bramer, M. (Ed.), Cambridge University Press, Cambridge, UK. (in print).

Watson, I.D., Basden, A., Brandon, P.S. (1992). The Client Centred Approach: Expert System Maintenance. Expert Systems, 9.

Wielinga, B.J. & Breuker, J.A. (1986). Models of Expertise, in Proc. of the European Conference on AI, pp.306-318.

Young, R.M. (1987). The Role of Intermediate Representations in Knowledge Elicitation, in Research and Development in Expert Systems IV, pp. 287-288. Moralee, D.S. (Ed.). Cambridge University Press, Cambridge, UK.