Basden, A., Watson, I., & Brandon, P.S. (1991).The Evolutionary Development of Expert Systems. In: Research and Development in Expert Systems VIII, pp.67-81. Eds. I.M. Graham & R.W. Milne. Publ. Cambridge University Press, Cambridge, UK. ISBN: 0 521 41836 4

The Evolutionary Development Of Expert Systems

BASDEN, A†., WATSON, I.D*., & BRANDON, P.S*.

University of Salford
†*Information Technology Institute*
**Department of Surveying*

Abstract

This paper describes a project that will study the evolutionary development of Expert Systems (ESs) within small & medium sized enterprises (SMEs). It is posited that such organisations cannot gain access to ES technology because of its high entry price and because the tools and methodologies now available were developed for and by large organisations. Within this context evolutionary development encompasses the entire life cycle of an ES from initial conception to implementation (termed Evolutionary Development Part 1 or ED1) and then to the eventual maintenance and updating of the system (Evolutionary Development Part 2 or ED2). The central premise is that the methodology should be "Client-Centred," emphasizing "what " the client can see rather than being "technology-centred," emphasizing how " the knowledge engineers work. Discussion includes how this methodology addresses many criticisms of prototyping ESs.

The project aims to study ED1 by developing two ESs for the construction industry, and to study ED2 by monitoring the use of an existing ES that has over 200 users. Accepting that the knowledge in an ES evolves means that the design process of the ES must recognise that the system will be updated. Hence, in addition to its advantages during knowledge acquisition, an intermediate representation should form the basis of system documentation. This can be supported by tools enabling the maintainer of the ES to refer to knowledge sources, program overviews and program code via structured links. However, although the project intends to develop software tools to support this approach, it is the functionality of the methodology rather than the specific software tools that the research will emphasise.

Keywords:

Expert Systems, Knowledge Based Systems, Methodologies, Intermediate Representations, Tools, Knowledge Acquisition, Life-Cycles Methodologies, Client Centred Methodology, Hypertext.

INTRODUCTION

This paper will describe the aims and objectives of a collaborative research project, funded under the IEATP, between the Information Technology Institute and the Department of Surveying 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 current Expert System (ES) development methodologies are "technology centred" and are not suited for use by small and medium sized enterprises (SMEs). 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 context of SMEs. More importantly, few SMEs can capture their expertise in an ES due to the high price of hardware, software, and the need to make experts available.

The EDESIRL project will involve approximately 20 SMEs and will result in a deeper understanding of the requirements for evolutionary development of ESs that are specific to SMEs. It will thus provide easier entry to the ES arena for SMEs thereby allowing them to capture their expertise rather than relying on packages supplied to them by others. This will facilitate a more rapid spread of good practice and professional expertise throughout SMEs in the UK.

A "methodology" is understood as a system of principles and methods brought together in fulfilment of a unifying goal: the production by SMEs of usable ESs that fullfil 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 methodology to guide evolutionary development of ESs for SMEs,
- to propose techniques based on the methodology,
- to develop a package of software tools to aid the methodology.

The evolutionary development of ESs is different from prototyping although it does encompass it. Rather, it is a recognition of the fact that the development of an ES is never finished and that in particular the knowledge in an ES continues to evolve after the system has come into use. There are two main parts to the methodology:

1. a life-cycle methodology (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,
2. a Knowledge Acquisition methodology (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 will develop a package of tools to aid the methodology, the techniques and the functionality of the tools are of primary importance. Moreover, we do not intend to "re-invent the wheel"; thus, where an existing tool exhibits the required functionality we shall use it providing we can integrate it sufficiently with the other software.

This paper first discusses the LCM and the KAM, then outlines the functional requirements of the software tools, and concludes with an outline of the plans for this project

LIFE-CYCLE METHODOLOGY

The LCCM to be developed in this project begins with Basden's Client-Centred Methodology (CCM) [1989]. The CCM has had successful use in industry,and is not merely an ad-hoc collection of experience; rather, is based on several key 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 an holistic approach throughout the project, including the consideration of usability and saleability from the start;
- the recognition that functioning software is better than paper models as a specification and discussion point;
- 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 know 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 an SME who may have some experience of using spreadsheets or databases but is not an IT professional).

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

- the client as above,
- the primary user who presses the keys or mouse buttons,
- those who must supply information to enable the primary users to operate the ES,
- those who use the information derived from it,
- the manager(s) of the primary user,
- those responsible for software/hardware support,

- the experts from whom the expertise is acquired in the building of the ES,
- customers of the organisation,
- suppliers to the organisation,
- and the wider society and environment.

To this end the methodology 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.

Though it is a staged structure, the CCM is designed to overcome the problems normally associated with linear structures insofar as it allows for evolutionary development. Moreover, it also 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 the knowledge engineers and domain experts but not necessarily those of the users.
- Prototyping can lead to uncontrolled growth of the system causing difficulties in maintenance.
- Because elicited knowledge is often translated directly into code, there is no complete and explicit statement of the knowledge included in the system. This also makes maintenance very difficult.
- Over reliance on the iterative refinement of ES's can result in their inability 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 in particular recognises that "people" should be at the centre of a methodology 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. Thus, the client is not fully involved and the above problems result.

The EDESIRL project intends to use the CCM to address these problems by maintaining a client-centred rather than a technology-centred focus. In the CCM the stages (see Figure 1) are named not after activities ("how") but after deliverables ("what") that the client can expect throughout the project. It ensures that an holistic client-centred view is established at the start. This encourages closer cooperation with all the stakeholders throughout the project rather than just at the early and the late stages.

While it might appear that these stages are rather artificial means of breaking up incremental development, in fact they have their own distinct purposes, likely techniques and deliverables.

Start (5 Hurdles)

The purposes of the first stage are as follows:

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

The first stage involves considering 5 Hurdles; these are questions designed to focus initial discussion and to ensure that nothing relevant or important is overlooked:

1. Is the problem suitable for IT (i.e. would it be better to have a person perform the task)?

2. Is the problem suitable for ES technology (e.g. could a spreadsheet or some other "conventional" program provide the same benefits)?

3. Is the knowledge for the ES available (e.g. willing and available experts)?

4. Will the ES be worth it (i.e. what are the benefits and costs to each stakeholder)?

5. How can we ensure the ES will be acceptable to all stakeholders(i.e. what affect will the ES have on working practice and organisational structures)?

If the project falls at one hurdle, the problem domain may be divided and the 5 Hurdles re applied to determine if ESs are an appropriate solution to part of the problem.

Various methods can be employed here, including Checkland's Soft Systems Methodology [1981] and SWOT Analysis. Questionnaires are being developed to guide this analysis along with an ES to serve as an advisor. Identifying roles [Basden, 1983] is important since these determine how the last three hurdles are overcome. The deliverable of stage 1 is an explanation on how the 5 Hurdles have been overcome, 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.

Skeleton System

This system shows a sequence of screens to give a "feel" of the 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, which can subsequently guide the project,
- to ensure the clients and the domain experts understand what ESs can be expected to do,
- and to help the knowledge engineer(s) obtain an initial picture of the domain and of the types of information that is handled.

The deliverable can be seen as an embodiment of the conventional top-level functional description, but since it is active, the experts (who often have little idea of what to expect from ESs) get a better "feel" for the system than they would from a paper specification, before knowledge acquisition starts in earnest. Some domain conceptualisation occurs as a result, which can form the basis for the next stage.

Demonstration System

In the third stage a significant amount of knowledge is acquired and encapsulated in an ES that can be demonstrated. Most domain conceptualization and structuring is completed. Its 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;
- and to provide a go/no-go decision point: Will the ES as proposed bring relevant benefits? Can the knowledge be encapsulated as an ES.

The methods employed in this stage are the usual ones of 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 a small number of experts for the knowledge acquisition and others for validation.) The holistic picture obtained during the 5 Hurdles 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 experience the use of the system.

Working System

While the 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 proceeds. 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. 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, and easy methods of data entry. It is advised that knowledge acquisition and validation be employed to determine what usability features are needed; the sources of knowledge should be the primary users of the ES. Too often usability features are approached as a set of features to be added to a system to make users happy. However, we are suggesting 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 should usually overlap with stages 3 and 4 to some extent. For instance, usability features like "what-if" and "dump-and-restore" can be useful to streamline the validation process, and a number can be added at carefully chosen points so that the client receives visual indication of progress instead of merely a report that, "The accuracy has increased from 85% to 90%." This stage is deliberately placed after stage 4 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. This involves attending to details of the User Interface, the wording of screens, a Help system, 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 whole 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 such 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 METHODOLOGY

The methodology the EDESIRL project will develop differs from most other knowledge acquisition methodologies in that it is based on an explicit model of expertise as 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], and states that it is insuffficient 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 out the U from the CPS element. This is achieved through three major activities:
- obtain expert problem solving heuristics (using well known elicitation techniques);
- seek the U that lies behind the heuristics,
- by asking 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; rather, they are performed as necessary, with the second and third activity forming the bulk of the work. This methodology 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 a fruitful provider of high quality knowledge.

INTERMEDIATE KNOWLEDGE REPRESENTATIONS

Because of the problems inherent with prototyping and the requirements of maintainability (i.e. ED2), 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].

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

- logically testable for consistency;
- speed of production of the intermediate representation;
- clarity and understandability 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.

We present here the functional requirements of the ideal intermediate representation and a comparison of 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 a domain expert are poor.

- Prototyping is not a representational formalism, but it is included because many researchers and methodologies support it. It scores highly in terms of 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 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 constitute 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 understand 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 tools that are technology-centred. The KnAcq is testable, expedient and can generate program code for several commercially available ES Shells. However, it can only handle certain types of knowledge, is not supported by an all embracing life cycle methodology, 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] but were then simply an example of another paper model, and were therefore slow to create, untestable, and not directly translatable. This project intends to use inference nets to explore the functionality of intermediate representations. Experience has shown that domain experts find them flexible and easy to understand. A prototype tool to assist in their creation and to test them for correctness has been implemented and will be further developed. They are fully supported by the CCM and in conjunction with the tools outlined in Figure 3 can provide useful system documentation. The research is addressing the translation from inference net into program code.

DISCUSSION

The description above is of the methodology as it is at present, and the EDESIRL project's purpose is to research and develop this. In particular, attention will be focused on the following issues. The first five have a high profile in the software engineering or knowledge engineering communities and have not yet been covered in the methodology explicitly, though some have been covered implicitly:

- **Risk**. Many project methodologies (such as Boehm's Spiral Model and the KADS methodology [Breuker & Wielinga, 1985] that derived from it) are couched in terms of risk. Risk handling is implicitly incorporated in the CCM in three ways:

    1. the 5 Hurdles are a mechanism of reducing the risk of failure,

    2. the iterative approach to building the ES reduces risk due to changing requirements,

    3. and the KAM's requirements attempts to uncover underlying understanding and context that minimise the risk of ES fragility.

- **Metrics**. Neither part of the methodology discusses metrics (measuring the accuracy, acceptability, impact, costs, etc. of the ES) because the methodology 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 methodology. The reason is perhaps historical in that the roots of the methodology lie in the era just prior to the upsurge in interest in QA. The whole aim of the methodology, however, is identical to that of many QA approaches, and it has been based on similar principles. It is intended that explicit QA principles and practice will be grafted into the methodology developed during this project.

- **Procedures**. The methodology as it currently stands states what ought to happen but does not yet show how to make it happen. Procedures will be developed concerning all activities involved in building ESs. It is not our intention, however, to prescribe a single procedure.

- **Documentation**. Owing to the importance of documentation and the frequency with which user documentation is inconsistent with a product, the methodology includes explicit guidance on documentation being produced as the ES is built, and so intends to define a set of stages for documentation similar to those of the CCM.

ED2. The project plans to develop a methodology for ED2 as well as ED1, but the current version has only been applied to ED1. We intend the methodology for ED2 to be based on the same principles as those of ED1. We are fortunate in having two ESs to study immediately (ELSIE and ELI) for the purposes of ED2, and the ESs developed during this project will undergo some ED2 before the end of the project. Nonetheless, it remains to be seen whether this experience, even enriched by that of others, will be enough to develop a mature ED2 methodology in the three years of the project.

CONCLUSION

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

1. a Client-Centred life-cycle methodology that places the client at the centre of the development process and informs them in terms of "what" they can see rather than "how" it was achieved,

2. and a Knowledge Acquisition methodology that obtains the understanding that informs heuristics.

However, the methodology requires development. Concepts such as risk, metrics, and quality assurance must be attended to. The methodology 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 methodology must be applied to ED2, whilst the information gathered from studying ESs in use must be re-incorporated into ED1.

This project has two and a half years to run, and we believe this is sufficient time to overcome many of the problem areas identified in this paper and thereby to produce usable methodology containing procedures to guide the evolutionary development of ESs.

ACKNOWLEDGEMENTS

The EDESIRL project would like to acknowledge the DTI for their funding contribution, and The Royal Institution of Chartered Surveyors, Imaginor Systems, and Inference Europe for their contributions to the research.

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.

Basden , A. (1983). On the application of Expert Systems, in IJMMS, 19:pp461.

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.

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.

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

Checkland, P. (1981). Systems Thinking, Systems Practice.

Chen, P.P. (1976). The Entity-Relationship Model -- Toward a Unified View of Data, in ACM Transactions on Database Systems 1(i): pp9-36.

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

Hines, J.G., & Basden, A. (1986). Experience with the use of computers to handle corrosion knowledge, in British Corrosion Journal, 21(iii):pp151-156.

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.

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.183-198.

Watson, I.D., Brandon, P.S., & Basden, A. (1991). The Development of Knowledge Based System for Predicting Building Maintenance, in

Basden, A., Watson, I., & Brandon, P.S. (1991).The Evolutionary Development of Expert Systems. In: Research and Development in Expert Systems VIII, pp.67-81. Eds. I.M. Graham & R.W. Milne. Publ. Cambridge University Press, Cambridge, UK. ISBN: 0 521 41836 4

Microcomputers in Civil Engineering, special issue on Artificial Intelligence and Construction, in print.

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.