ACKnowledge:
An integrated workbench supporting Knowledge Acquisition.

Jean-Charles Marty - Fano Ramparany
Marie-Stephane Doize - Cleo Jullien

Cap Gemini Innovation - ZIRST
7, chemin du vieux chene, 38240 Meylan FRANCE

Tel. +33 76 76 47 36 - e-mail marty@capsogeti.fr

Abstract

Knowledge Acquisition is a critical and time-consuming phase in the development of Knowledge Based Systems. The ACKnowledge project aims to improve the efficiency of the knowledge acquisition process. The approach is to analyze and evaluate a variety of existing knowledge acquisition techniques, including machine learning methods. Taking into account their complementarities, we integrate these techniques into a Knowledge Engineering Workbench that supports the Knowledge Engineer in his various tasks. This approach is tested on real life applications, simple ones (e.g. analysis in metal fractures) and more complex ones (e.g. failures in the Spanish data communications network).

Key words: Knowledge Acquisition (KA), Integration of KA techniques, Knowledge Engineering Workbench, Integration of Knowledge.

1 Introduction

Currently, much work in the AI community is related to the development of **Knowledge Based Systems (KBS)**. We already know that the real bottlenecks in the development of such systems are the acquisition and modelization of the expert's knowledge. Knowledge acquisition refers to the initial phase of KBS development where the **Knowledge Engineer** gathers, analyzes and models the expert's knowledge. The efficiency of knowledge acquisition and the quality of the resulting model of expertise are crucial for the subsequent development phases.

Much work aims at supporting the knowledge engineer in this knowledge acquisition process. Some interview techniques have been developed, some of them together with a software support. But most of them only address particular problems and are not general enough to cover the entire process.

During the European Knowledge Acquisition Workshop 89 (EKAW), one of the conclusions drawn was that the scientific community should concentrate more on putting these tools together and studying their synergy rather than developing new and more sophisticated tools. It was also mentioned that the two fields of "Knowledge Acquisition" and "Machine Learning" should be more closely related.

This is perfectly in line with the work that was going on in the ACKnowledge project[1] [1]. We believe that the efficiency of the **Knowledge Acquisition (KA)** process will be improved if we take advantage of the complementarity of existing techniques. We are thus developing a framework that allows the fruitful combination of a large range of existing KA techniques, including machine learning techniques.

The mere inclusion of these techniques into a single system constitutes the first step for the construction of a **Knowledge Engineering Workbench (KEW)**. But this is not enough. If we want our workbench to be useful, we also need some knowledge about the usability of these techniques. With this new knowledge about the nature and applicability of the techniques, the KEW will actively assist knowledge engineers in their tasks, carrying out some parts of the work by itself, and providing the user (knowledge engineer) with some advice and guidance.

In this paper, we first describe in detail our vision of the KEW, the functions it must provide and its general architecture. We then explain the knowledge integration process, central theme in our approach. Relation to other work in the domain is described. Finally, we give some information on the real-life applications used for the validation of KEW.

2 The Knowledge Engineering Workbench

In this section, we describe several aspects of the workbench that we are developing in the project.

2.1 The KEW Vision

As we said above, the goal of ACKnowledge is to construct a workbench, that **incorporates different knowledge acquisition tools** within a single integrated system. No new knowledge acquisition tools will be developed within ACKnowledge. However, new implementations will be needed to provide a coherent integration of techniques. Moreover, the workbench will have an open architecture which will leave room for the addition of new tools to the system.

The construction of such a workbench meant that the project had to develop a conceptual framework adequate for characterizing, in a systematic and principled way, the function of

---

[1] The ACKnowledge (ACquisition of Knowledge) Project is a 3 year project partially funded under the auspices of the ESPRIT Programme of the Commission of the European Communities as project N°2576. The project Consortium consists of the following institutions; Cap Gemini Innovation(F), Marconi Command and Control Systems(UK), GEC-Marconi Research Centre(UK), Telefonica(E), Universidad Politecnica de Madrid(E), Computas Expert Systems(N), Veritas Research(N), the University of Amsterdam(NL), Sintef(N), and the University of Nottingham(UK).

Figure 1 Basic knowledge acquisition cycle

various techniques and tools. The workbench must be able to deal with a wide range of tools, input and output types, and user interactions.

Ideally, the user obtains a certain type of information with one tool and other kinds of information with other tools. Furthermore, we wanted the information acquired by one tool to be usable by other tools. This required research into the maintenance of knowledge base coherence and methods to effect **transformations between various knowledge representations**. Part of this research is presented in section 3.

The knowledge engineer must consider the knowledge acquisition process as an **incremental process** [2]. At each iteration of the process, information acquired so far is evaluated. Based on the result of this evaluation, an appropriate technique is selected and applied to obtain new information from sources of domain knowledge such as human experts and documents. This information is intended to refine and enrich the current version of the integrated knowledge base.

Further decomposition of one iteration of the KA process reveals the following basic operations:

- **select:** identify which part of the acquired knowledge to evaluate.
- **check:** check the selected piece of knowledge in order to find defects in quality[2].
- **problem identification:** identify and select which of the defects to pursue at the current stage of the KA process.
- **technique selection:** find the adequate KA technique for fixing the problem.
- **technique application:** apply the KA technique to the piece of knowledge selected in the first step.
- **result assimilation:** find out how results of the technique application can be assimilated into the entire existing knowledge.

Obviously from the above decomposition, the control of the KA process is mostly **cyclic** as described in figure1.

2.2 Levels of KEW support in the KA process

We envisage workbench support to the KA process at four different levels:

- The first level provides the knowledge engineer with a support for his **clerical activities**. Editing examples for KA tools, reporting on the current status of the KA process, and all this kind of activities are gathered in this level.
- The second level concerns the **execution of the various KA tools** included in KEW. Data inputs and presentation of outputs of these techniques are also considered at this level.
- The third level gathers the support functions that allow the user to assimilate the results obtained with one particular tool to other knowledge obtained previously. Section 3 explains this assimilation of knowledge in more details.
- The fourth level is the control part of KEW that will provide the user with **advice and guidance**. For instance, it will give information about what technique or sequence of techniques is best adapted to the current phase of the KA process.

Each of these levels enriches the preceding ones: If one considers the environment including the functions of the first level only, the knowledge engineer has basically a tool for his project management. With the second level, tools become available either for his interviews with the expert or for his analysis of these interviews. By adding the third level, the synergy among all the tools appears and a real integration of the knowledge obtained from these various tools is possible. Finally, the fourth level gives intelligence to the workbench. The knowledge engineer is not in front of a range of tools without really knowing which one to apply: instead, he receives advice on what the possibilities are at this stage.

2.3 The KEW Architecture

The above analysis of the KA-process has been used as a basis for the KEW system architecture, which is shown in figure 2.

Central to the architecture is the **Common Information Repository (CIR)** which stores all information needed during the Knowledge Acquisition process: partial or intermediate piece of knowledge, such as protocol transcript or structured cases; project management documents; and the **Integrated Knowledge Base (IKB)** which is the final and complete knowledge base that the KA process aims to produce. Different **Knowledge Acquisition tools (KA-tools)**[3] support the KEW user during his basic KA-tasks. One such task yields a **Tool Specific Knowledge Base (TSKB)** whose content is to be assimilated later on into the IKB using appropriate **Knowledge Assimilation (K-assimil)** mechanisms. "Tool Specific" means that the knowledge is represented using a language that is specific to the KA-tool. Knowledge assimilation mechanisms will be detailed further in section 3. The **reasoning** component embodies an interpreter of the IKB knowledge representation and allows the user to perform reasoning on the knowledge currently contained in the IKB for testing or simulation purposes.

The **User Interface (UI)** allows the user to browse through the CIR and access its constituents (via the **CIR server**), to

---

[2] By qualities we mean properties like completeness, correctness, comprehensibility, modularity, etc...

[3] The list of the tools included in KEW is given in figure 4.

use the KA-tools (possibly under the control of the Advice and Guidance module), and to reason on the knowledge acquired so far.

An **Advice & Guidance (A&G)** module provides assistance in decomposing the overall KA-process into KA-activities and further on into basic KA tasks, in selecting the appropriate KA-tool for the current KA task. This module includes a planner component which instantiates the generic KA-cycle presented in section 2.1 and which feeds an **agenda** containing remaining operations of the current KA-cycle plan, and activatable operations ordered with respect to their relevance. The agenda content is derived by using knowledge about the KA process (KA activity breakdown, constraints due to domain/application characteristics), knowledge of generic models (models of problem solving methods, domain structures and content,...), knowledge of the KA-tools themselves (functionalities, input constraints...) and knowledge on the results of KA activity (evaluative knowledge: consistency, coverage, generalization). This knowledge is stored in the **Knowledge Engineering Knowledge Base (KEKB)**. More details about this component can be found in [3].

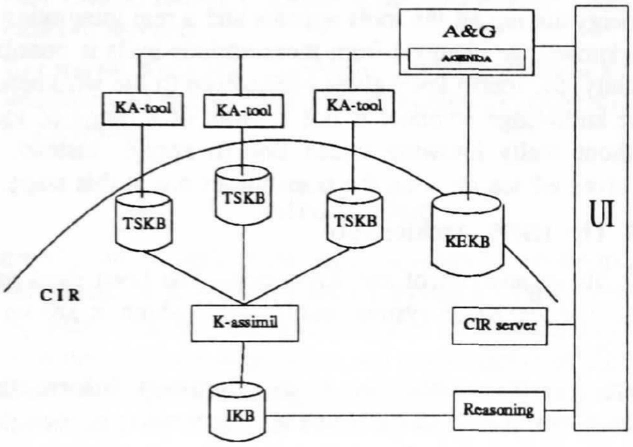

Figure 2 KEW architecture

3 Knowledge Assimilation

Knowledge assimilation is a key issue addressed in the project [4], [5].

As mentioned in section 2.3, the KA process generates a collection of Tool Specific Knowledge Bases, resulting from the different KA technique sessions. Due to the multiplicity of KA techniques those TSKBs will have heterogeneous knowledge representation. In order to assimilate the contents of those knowledge bases, we need functions which transform expressions written in the technique specific knowledge representation formalism into a common knowledge representation formalism. This first transformation step produces **Temporary Knowledge Bases** (Temp KB) that are images of TSKB represented in the common formalism. Functions are also required which integrate Temp KB into the central **Integrated Knowledge Base**.

The contribution of both **knowledge transformation** (K-transf) and **knowledge integration** (K-integr) functions with regard to the knowledge acquisition process is shown in figure 3:

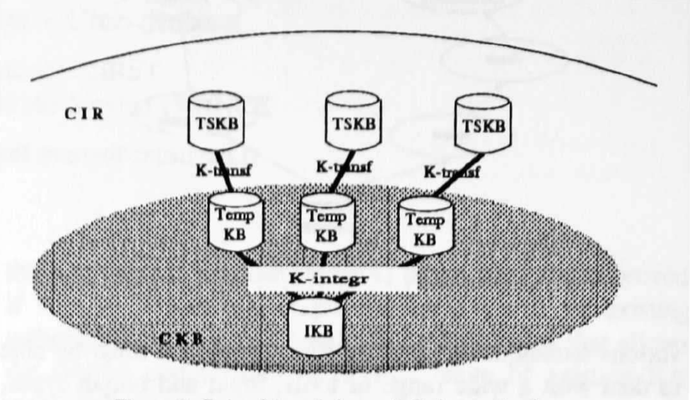

Figure 3 Role of knowledge assimilation mechanisms

The common information repository has been partitioned in two areas:

- The non shaded area contains heterogeneous knowledge bases that contain data used and produced by knowledge acquisition tools.
- The shaded area is called **Core Knowledge Base (CKB)** and contains knowledge bases expressed using the common knowledge representation language.

Transformation functions produce temporary knowledge bases as part of the Core Knowledge Base, by transforming the contents of tool specific knowledge bases into common knowledge representation structures. Integration functions then merge the different temporary knowledge bases into the integrated knowledge base.

In the following sections we analyze the main issues involved in knowledge transformation and knowledge integration.

3.1 Knowledge transformation

We have based our analysis of knowledge transformation on the type of data used and produced by the KA tools that are to be integrated within KEW. Figure 4 enumerates those techniques and their corresponding data types.

The diversity of knowledge representations is quite striking. As one may guess, some knowledge transformation are straightforward, due to the similarity (formally and functionally) of the expressions. For example, entailment rules produced by the repertory grid are easily transformed into CKB production rules. Transformation mechanisms mainly involve syntactic manipulation.

On the other hand, when knowledge representations are very different from one another (mapping between entities defined within the two representation formalisms is not trivial), additional knowledge is required to bridge the semantic gap. Usually, this implies user interaction.

More generally, we distinguish two aspects of knowledge transformation: The syntactic aspect and the semantic aspect.

The **syntactic aspect** of knowledge transformation refers to how terms and constructs from one language are transformed

KA Techniques	*Knowledge representations*
protocol editor	fragments of text conceptual model
concept editor	hiearchies of concepts
laddered grid	hierarchies of concepts
repertory grid	cases grids ratings entailment rules
similarity based learning tool	structured cases frame hierarchies classification rules
case based reasoning tool	cases attributes relations causal net
case sort tool	hierarchies of concepts identifiers
core knowledge base editor	frames production rules relations

Figure 4 Various KA techniques and their specific knowledge representation

in the other language. At this level one is concerned mainly with symbol manipulation. For languages based on the same paradigms, knowledge transformation consists mainly of syntactic transformation.

The **semantic aspect** of knowledge transformation refers to what primitives from one representation means with regard to the other representation. In a more practical view, semantical aspect of the transformation concerns the way basic primitives of the original representation scheme will be interpreted by the target representation scheme. For example, will a knowledge source be transformed into a set of rules or into a procedure within the common knowledge representation language?

Purely syntactic transformation is automatic and does not require the interaction of the user, whereas semantical considerations often require user interaction to guide the transformation.

Once a piece of knowledge resulting from a KA technique session has been expressed in the common knowledge representation language (via the appropriate knowledge transformation application), it is ready for integration into the integrated knowledge base.

3.2 Knowledge integration

We define knowledge integration as the operation of merging two different pieces of knowledge to form a unique and coherent knowledge base.

The main requirements for knowledge integration are that the semantics of pieces of knowledge to be integrated be preserved and that some quality criteria be fulfilled. This requirement means that if two KA tools are to use the same piece of knowledge (refer to an unique object in the real world), it should have the same meaning within each tool framework.

Within KEW, we have tackled this requirement by designing an algorithm for integration [4]. The algorithm involves four steps:

- **Ordering:** Ordering the knowledge pieces into an integration order;
- **Recognition:** Identification of the knowledge in the initial base relevant to the knowledge to be integrated;
- **Elaboration:** Modification of the new knowledge according to expectations provided by relevant knowledge in the KB;
- **Adaptation:** Modification of the knowledge base to accommodate the elaborated information.

4 Relation to other work

The KBS development process can be looked at from three viewpoints: **activity-oriented, result-oriented** and **internal view**. These views enlighten particular aspects of the process that must be considered for building a support tool [2].

The ***activity-oriented view*** is similar to the KBS life-cycle view. The KBS development process appears as a set of activities ordered in time. Examples of such activities are: *task analysis, static domain analysis, expertise modeling or target knowledge base construction.* Activities give a macroscopic view of the knowledge engineering process.

Activity support tools are dedicated to one or more activities. Expert system building tools such as *KEE, ART* [6] *or Knowledge-Craft* [7] belong to this class of tools. Obviously, all the tools supporting a KBS development methodology are also activity support tools (e.g. KPT [8] and more recently Shelley [9] both supporting the KADS methodology [10]).

KEW covers the **activity-oriented view** because it gives global guidelines for structuring the KBS development process. It provides the knowledge engineer with advice on which activities are to be performed and in which order. It also indicates which techniques are available for achieving a particular activity.

The ***result-oriented view*** is concerned with the results to be achieved during the development process. *Lexicon, conceptual model* or *knowledge base* are examples of such results. In this view, we are mainly concern with the contents and representation of these results. Indeed, the same result can be represented in different ways. For instance, a knowledge base may be represented with rules, classification trees or logical expressions.

Model/result driven tools guide the user with models of the results to be obtained in the KBS development. *Teiresias* [11], *Roget* [12], *Opal* [13] or *Mole* [14] belong to this class of tools. For instance, the model of rules to be obtained at the implementation level is known in Teiresias.

In KEW, the **result-oriented view** is used as the basis for representations that KEW has to maintain during the KA process. In particular, KEW is equipped with models of results that should be achieved during certain stages of the KBS development.

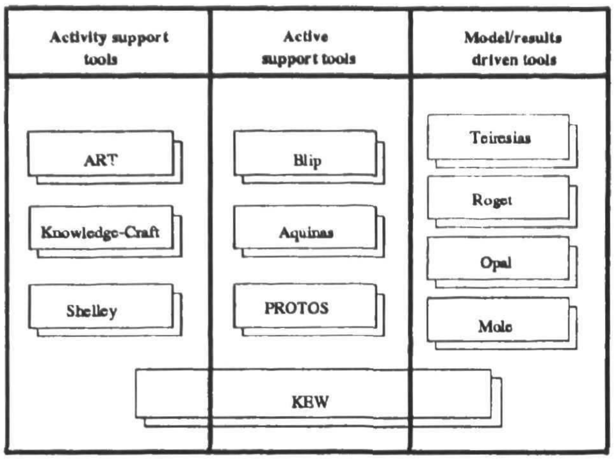

Figure 5 Classification of knowledge engineering support tools

The *internal view* is a microscopic view of the process in the sense that it shows its internal structure. It highlights the elementary tasks that compose the global process and how they interact in order to achieve activities and results. Examples of such tasks are *vocabulary construction, concept editing inference modeling.*

Intelligent support tools use some knowledge about the knowledge engineering internal process itself and are able to perform parts of that process autonomously. *Blip* [15], *Aquinas* [16] or *PROTOS* [17] belong to this category. For instance, Blip provides support by using knowledge about generalization and integration processes in KB construction.

From this internal view, the KA process can be split into elementary tasks that allow the identification of the appropriate KEW components to be activated.

Figure 5 shows a summary of the mentioned tools classification.The originality of KEW is that it is based on a synthesis of the KA process views. This is a good ground for obtaining a more complete and generic workbench.

5 Applications

In our project, we are testing and validating our KEW on several applications. We have chosen real life applications in order to make a valuable evaluation of our KEW.

5.1 Simple applications

We first investigated the area with several "simple" applications, in order to provide data for testing our first prototype. This first prototype was a collection of KA tools running in the same environment but without any integration (level 2 of section 2.2).

Different partners provided the consortium with such applications. We have worked on various domains such as analysis of metal fracture, object recognition and the "Tomato Plant Pathology domain" [18].

The **analysis of fracture damages** in steel constructions is used for instance in shipping where there is a wide range of equipment prone to such failures.

Due to its large set of cases, the application on **recognition of objects** (viz cars/trucks etc.) in 2D images was especially suitable for the machine learning component of the KEW.

The diversity of these applications gave us a clear idea of the kind of data usually available to the Knowledge Engineer.

5.2 The telecommunication network failure application

The application that is being used throughout the project is more complex. It deals with the Spanish Data Communications network [19]. The final KB system must **assist the operators in initiating corrective actions** when network anomalies occur. For making such decisions, the operators usually use network diagnostic aids and tools.

The operators are in charge of ensuring good network efficiency at all times. This problem is difficult because operational procedures are complicated and cases often arise which are not completely described in the manuals. Furthermore, modern telecommunications networks are increasing in complexity. Effective maintenance thus requires assimilation of vast amounts of operation and maintenance knowledge.

This application often guided the specifications of our final KEW. For example, the list of tools to be incorporated in the KEW was drawn up from this application requirements.

5.3 The Ship Bridge application

A new application is going to be implemented during the last year of the project. The area we have selected is new within classification of ship and deals with **bridge design.** The rules for this classification are aimed at verifying the design of a ship bridge in order to facilitate the shipmaster's work and thereby make the ship movements as safe as possible.

There exists only a handful of experts in the area of bridge design, and they spend much of their time traveling around the world, teaching local surveyors how to use the new rules. This is both a tedious job for the experts and quite expensive for the company. If a computer could do at least parts of this job, it would be a major achievement.

The rules for bridge design are implemented as an electronic rulebook. These rules can be used as a skeleton for adding new knowledge, and/or be the object of improvement itself. This application will thus be a very good testbed for the behavior of KEW in the **refinement phase.**

6 Status of work and conclusion

Viewing KA as an incremental construction process based on various elementary tasks and intermediate results allows us to design a knowledge engineering workbench that actively supports knowledge engineers.

KEW is much more than a simple tool box that allows the execution of a collection of KA techniques. KEW provides support for various integration mechanisms that allow an effective combination of the tools: storage and representation of acquired knowledge, translation and integration facilities for partial results obtained with different tools, advice and guidance for the selection of techniques taking into account both applicability and purpose of KA techniques, as well as advice for scheduling KA tasks.

From what we have already achieved, we believe that our approach to the problem is viable and that the main functionalities of the KEW can be implemented shortly.

Currently the project is at mid-course. Until now, we have devoted most of our effort to the conceptual framework and to the development of two prototypes written in Sun Common Lisp [20]. The first gathers some KA techniques without integration of knowledge. The second has allowed us to test some of the KA transformations that we intend to perform in the final KEW. Simultaneously we have analyzed a large range of existing KA techniques [21]. The difficult part of our project remains the "Advice and Guidance module". Most of the knowledge needed on the applicability or efficacy of tools is still to be acquired. Evaluation work is therefore going on for this purpose. We have recently delivered the functional specifications of the final KEW to the CEC. This final KEW will be demonstrated in October 91 at ESPRIT Conference, in Brussels.

Acknowledgements

We are grateful to members of the ACKnowledge team for the help in developing the ideas in this paper. In particular, we would like to thank Jorunn Brække, Sergio Calabretta, Jorun Eggen, Francisco Garijo, Inge Nordbø, Nigel Shadbolt and Bob Wielinga. Particular thanks to Didier Bronisz, Thomas Grossi, Bertrand Rousseau, Maurice Schlumberger and Philippe Vauquois for their valuable comments on this paper.

References

[1] J.-C. MARTY, N. SHADBOLT, and B. WIELINGA, "The acknowledge project improving the knowledge acquisition process.," in *Results and Progress of ESPRIT Projects in 1989, XIII/416/89*, November 1989.

[2] B. WIELINGA and al., "Conceptualisation of a knowledge engineering workbench," Report ACK-UvA-T1.4-DL-010-A, ACKnowledge project, ESPRIT, september 1989.

[3] N. SHADBOLT and B. WIELINGA, "Knowledge-based knowledge acquisition: The next generation of support tools.," in *Current Trends in Knowledge Acquisition*, (Amsterdam), June 1990.

[4] J. EGGEN, A. LUNDTEIGEN, and M. MEHUS, "Integration of knowledge from different knowledge acquisition tools," in *Current Trends in Knowledge Acquisition*, (Amsterdam), June 1990.

[5] I. NORDBO and J. EGGEN, "Information pool for knowledge engineering support - the common information repository.," in *ESPRIT Conference 90*, November 1990.

[6] B. CLAYTON, "Art, automated reasoning tool, programming tutorial," 1985.

[7] C. Group, "Knowledge craft 3.0. reference manual," 1985.

[8] A. ANJEWIERDEN, "Knowledge acquisition tools," *AI Communications*, vol. 0, pp. 29–38, August 1988.

[9] A. ANJEWIERDEN, J. WIELEMAKER, and C. TOUSSAINT, "Shelley - computer aided knowledge engineering," in *Current Trends in Knowledge Acquisition*, (Amsterdam), June 1990.

[10] G. SCHREIBER, B. GREDEWEG, P. D. GREEF, P. TERPSTRA, B. WIELINGA, E. BRUNET, N. SIMONIN, and A. WALLYN, "A KADS approach to KBS design," Tech. Rep. Delivrable B6 Esprit Project P1098, University of Amsterdam, September 1989.

[11] R. DAVIS, "Interactive transfer of expertise," *Artificial Intelligence*, vol. 2, no. 12, pp. 121–157, 1979.

[12] J. BENNETT, "ROGET : a knowledge-based system for acquiring the conceptual structure of a diagnostic expert system," *Journal of Automated Reasoning*, vol. 1, pp. 49–74, 1985.

[13] M. LIQUIERE and J. QUINQUETON, "OPAL : object-oriented programming and learning," Rapport de Recherche 32, Centre de Recherche en Informatique de Montpellier, Montpellier, France, Janvier 1987.

[14] L. ESHELMAN and J. McDERMOTT, "MOLE : a knowledge acquisition tool that uses its head," in *Proceedings of the 5th National Conference on Artificial Intelligence*, (Philadelphia, PA), pp. 950–955, August 11-15 1986.

[15] K. MORIK, "Sloppy modelling," in *Knowledge Representation and Organisation in Machine Learning* (K. Morik, ed.), Springer Verlag, 1989.

[16] J. H. BOOSE and J. M. BRADSHAW, "Expertise transfer and complex problems: using aquinas as a knowledge acquisition workbench for knowledge-based systems," in *Knowledge acqusition for knowledge based systems (vol. 2)* (J. H. Boose and B. Gaines, eds.), pp. 39–64, Academic Press, 1988.

[17] E. BAREISS, B. PORTER, and C. WIER, "PROTOS : an examplar-based learning apprentice," in *Proceedings of 2nd AAAI Workshop on Knowledge Acquisition for Knowledge-Based systems*, (Banff, Canada), pp. 1–16, October 1987.

[18] G. COLL and al., "Simple application contexts: Xfrac, danae, vehicle image and tomato plant pathology domains," Report ACK-CX-T5.1-DL-002-A, ACKnowledge project, ESPRIT, June 1989.

[19] L. SOLORZANO and al., "Complex application context for kew-1," Report ACK-TID-T5.2-DL-001-A, ACKnowledge project, ESPRIT, February 1990.

[20] F. RAMPARANY, "KEW: a Knowledge Engineering Workbench that integrates cooperative Knowledge Acquisition Techniques," in *Proc. Third Intern. Symposium on AI*, (Monterrey, N.L. Mexico), October 22-26 1990.

[21] M. V. SOMEREN and al., "A classification of techniques for knowledge acquisition," Report ACK-UvA-T1.2-DL-004-C, ACKnowledge project, ESPRIT, august 1989.