In the last subcase with two edges with neighbors attributed to B according to eq 18, we show that half of the contribution of the edge AB and the edge to the nonneighbor cancel with each of the contributions according to eq 18. Thus,

$$(1/\sqrt{3c} - 1/\sqrt{4(c+1)} - 1/(4c\sqrt{c})) + (1/\sqrt{6} - 1/\sqrt{8})/2 - 1/(2\sqrt{32}) \quad (29)$$

is an upper bound for one half of the contribution of the edges at B to $\chi_R(G')-\chi_R(G)$. We use eq 25 once more and obtain the monotone upper bound

$$(1/\sqrt{3} - 1/2)/\sqrt{c} + (1/\sqrt{6} - 1/\sqrt{8})/2 - 1/(2\sqrt{32}) \quad (30)$$

whose value $\sqrt{6}/4 - 7\sqrt{2}/16 \approx -0.0063$ for $c = 2$ is negative. Since the constants $2(1/\sqrt{6} - 1/\sqrt{8}) - 1/\sqrt{32} \approx -0.067$ in eq 27 and $1/(12\sqrt{3}) + (1/\sqrt{6} - 1/\sqrt{8}) - 1/\sqrt{32} \approx -0.074$ in eq 28 are less than the constant $(1/\sqrt{6} - 1/\sqrt{8})/2 - 1/(2\sqrt{32}) \approx -0.063$ in eq 29, we have shown that $\chi_R(G')-\chi_R(G)$ is negative in all cases, and the proof of the theorem is complete.

REFERENCES AND NOTES

(1) Kvasnička, V.; Pospíchal, J. Canonical Indexing and Constructive Enumeration of Molecular Graphs. *J. Chem. Inf. Comput. Sci.* **1990**, *30*, 99–105.
(2) Wiener, H. Structural Determination of Paraffin Boiling Points. *J. Am. Chem. Soc.* **1947**, *69*, 17–20.
(3) Randić, M. On Characterization of Molecular Branching. *J. Am. Chem. Soc.* **1975**, *97*, 6609–6615.
(4) Rouvray, D. H. The Modelling of Chemical Phenomena Using Topological Indices. *J. Comput. Chem.* **1987**, *8*, 470–480.
(5) Rouvray, D. H. The Limits of Applicability of Topological Indices. *THEOCHEM* **1989**, *54*, 187–201.
(6) Morgan, H. L. The Generation of a Unique Description for Chemical Structures—A Technique Developed at Chemical Abstracts Service. *J. Chem. Doc.* **1965**, *5*, 105–113.
(7) Pospíchal, J. Private communication, 1991.

Domain-Oriented Knowledge-Acquisition Tool for Protein Purification Planning

HENRIK ERIKSSON

Department of Computer and Information Science, Linköping University, S-581 83 Linköping, Sweden

Received August 22, 1991

Knowledge acquisition for expert systems can be facilitated by appropriate supporting tools. By providing knowledge-acquisition tools that closely resemble the experts' own conceptual model of the domain, the knowledge-acquisition bottleneck can be widened significantly. P10 is a domain-oriented knowledge-acquisition tool specialized to the problem area of protein purification planning. P10 enables specialists in protein purification to develop expert systems for this task without a mediating knowledge engineer. Knowledge structures entered in a graphical interface are transformed into appropriate knowledge base constructs by P10.

1. INTRODUCTION

Knowledge acquisition (KA) has long been recognized as a major bottleneck in the development of expert systems. Computer-based tools that supports different aspects of the KA process have been suggested and, to a varying degree, implemented and used for practical purposes. However, little has been published on KA tool support for chemical informatics. Commercial expert systems shells offer a level of abstraction (e.g., rules and objects) suitable for knowledge engineers, but current shells are largely unsuitable for domain experts.

P10 is a knowledge-acquisition tool that provides specialists in protein purification planning with means to enter, edit, and review their planning knowledge according to a conceptual domain model supported by the tool. The user is required to fill in forms and to enter *partial plans*[1] (i.e., planning schemata or recipes) in a graphical user interface. The knowledge acquired by P10 is automatically transformed through a knowledge base generator into knowledge bases that can be used immediately for testing. Thus, experts can use P10 to develop the bulk of an expert system without an intermediate knowledge engineer.

Earlier KA tools have been based on a particular knowledge elicitation technique (e.g., ETS[2]) or a model of the problem-solving method used (e.g., MORE,[3] MOLE,[4] and SALT[5]). Knowledge acquisition tools such as P10 draw their power from an abstract model of the domain in question rather than other (more general) models such as models of classification, planning, and design. The major advantages of this approach are that specialized tool support is provided, which typically leads to better understanding of and acceptance for the KA tool among experts, and that KA environments can be tailored to the requirements from a small group of experts. The rest of this article develops the idea of domain-oriented KA tools for protein purification and reports the domain model supported by P10 as well as the P10 implementation.

2. BACKGROUND

Research in biochemistry requires isolation of proteins that are present in only trace amounts from biological material. Such an isolation is performed in a number of stages of which purification by high-resolution chromatography techniques is the last one. The outcome of the purification depends on the order in which different chromatography techniques are applied and on the running conditions for each of them. A complete purification plan is more or less impossible to construct in advance, since the effect of each operation cannot be fully predicted (at least when important properties of the protein is unknown). Thus, planning protein purification requires experience and skill.

2.1. Planners for Protein Purification. In an earlier project we have studied computer-supported planning for protein purification. As part of this project an expert system (P8) that supports a chemist in planning protein purification was implemented. P8 supports liquid chromatography techniques applied to membrane-bound proteins. The system gives advice on which techniques to use, the running conditions for each step, and measurements to be taken between each purification step. The P8 planner is based on a set of *partial plans*,[1] i.e., skeletal solutions that could be refined and combined into a

0095-2338/92/1632-0090$03.00/0

proper plan. This way of reasoning with partial plans is close to the conceptual model of standard recipes common among experts in protein purification planning. Planning aspects on protein purification and the P8 system are described in further detail elsewhere.[1,6] A related approach is reported by Ananda et al.[7]

2.2. Problem Definition. Although knowledge-based planners such as P8 can be used to come up with initial plans, their performance is ultimately dependent on the quality and scope of their knowledge bases. Therefore, knowledge acquisition is an important issue for the development and maintenance of these planners. Development through "conventional" means, i.e., through knowledge engineers interviewing domain specialists, is a laborious task and, consequently, a limiting factor that hampers the performance of the planner. More specifically, difficulties occurred in the P8 project as the size and complexity of the knowledge base grew, and it became apparent that we had to move toward computer-supported knowledge acquisition.

Part of the problem is that rules, objects, and other symbol-level constructs[8] provided by expert system shells are in many senses inappropriate media for expressing planning knowledge for protein purification. Indeed, the kind of rules (and objects) supported by commercial expert systems shells are inappropriate for many fields since much domain knowledge cannot easily be expressed in such forms. In some cases domain knowledge has been forced into the form of rules, which resulted in systems difficult to extend, debug, and maintain. Overconfidence in the services provided by shells has resulted in many failures and partial failures in the implementation of expert systems.

Some universal purification rules can be extracted, but much of the reasoning is centered around more practical structures, e.g., recipes and partial plans. KA tools in this field are required to handle such abstract structures and to enable experts to formulate knowledge on their own.

2.3. Tool-Supported Knowledge Acquisition. Several classes of KA tools that support the transfer of expertise have been reported in the literature.[9] Although their support is limited, *rule editors* provided by various shells can be viewed as a simple form of a KA tool if they are used directly by experts. Other, more advanced, KA tools are based on specific knowledge-acquisition and structuring *techniques*, e.g., different grid techniques. There are also KA tools that acquire knowledge for specific *problem-solving methods*, e.g., various forms of classification and design methods.

The most specific class of KA tools are the ones that are tailored to a particular application, i.e., *domain-oriented* KA tools.[10-12] These tools are typically based on some form of conceptual model of the domain, e.g., models of cancer treatment plans[10] or protein purification planning. The advantages of such KA tools are that they are easy to use for domain experts and that they provide efficient support for the application in question. Domain-oriented KA tools can also be used as a documentation tool to report difficult cases, exceptions, etc., for later inspection.

2.4. P10 Project. The P10 project was formed to provide support for knowledge acquisition for the P8 knowledge base. The overall purpose of the project was to (a) explore how specific models of domains can be used to facilitate knowledge acquisition for expert systems and (b) find suitable architectures for KA tools supporting such models. In particular, our goal has been to find a conceptual domain model for protein purification planning and to develop the KA tool support for it.

The project has gone through the following stages: (1) adoption of the P8 planner to knowledge acquisition systems and refinement of the conceptual domain model in P8,[13] (2) development of a domain-oriented KA tool (P10) that acquires knowledge for the above-mentioned expert system, and (3) evaluation of the P10 system. This paper reports stage 2. The results from the evaluation of P10 are briefly mentioned in Section 3.5, and the details are reported elsewhere.[14]

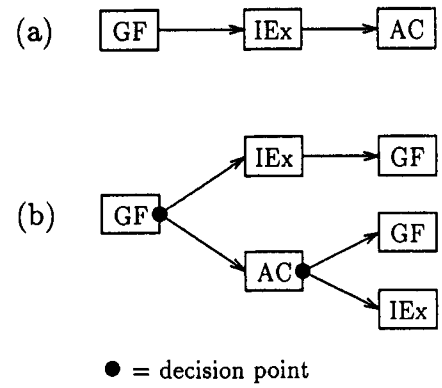

Figure 1. Linear recipe (a) and nonlinear partial plan with alternative paths (b).

3. P10: A DOMAIN-ORIENTED KA TOOL

P10 is a knowledge-acquisition tool that provides specialists in protein purification with the means to formulate and edit their recipes and knowledge about purification strategies. The scope of P10 is liquid chromatography techniques, and P10 supports gel filtration, ion exchange, and affinity chromatography techniques. P8 was initially limited to membrane-bound proteins, but this restriction was removed in P10. P10 features graphical editors for partial plans for protein purification and various decision points in the process. These editors are highly specialized to protein purification and liquid chromatography techniques in order to provide experts with a tailored environment suited for their needs. The rest of this section describes the P10 system and its implementation in detail.

3.1. Knowledge-Acquisition Model. P10 is a domain-oriented KA tool since it is based on a conceptual model of protein purification and it adopts the ontology of liquid chromatography. The underlying assumption for P10 is that experts should take an active part in formulating knowledge. P10 is organized around a set of editors in which experts can *actively* enter their domain knowledge. These editors provide a layer that completely abstract away symbol-level constructs, such as individual objects, rules, etc., in the knowledge base. P10 provides the user feedback by displaying previously entered knowledge and by test-running newly entered knowledge (i.e., running prototype versions of P10-generated knowledge bases). Hence, the intended user category is experts trained in the usage of P10.

P10 is complete in the sense that an expert alone can develop a complete knowledge base in the class of problems supported. Note that P10 is *not* primarily intended as a tool for knowledge engineers. (Although it can be used by knowledge engineers to enter structures according to the model of protein purification in the same way as an expert would do.)

3.2. Conceptual-Domain Model. A convenient way to think of planning knowledge for protein purification is the notion standard recipes for typical cases.[15] Figure 1a exemplifies such a (linear) recipe for liquid chromatography, i.e., a sequence of gel filtration (GF), ion exchange (IEx), and affinity chromatography (AC).

Often, purification recipes are reported in the literature for relatively well-known proteins, but for largely unknown proteins, linear recipes are inappropriate due to difficulties in predicting the properties and the behavior of the proteins. Consequently, the conceptual model that P10 supports allows

Parameters	Branch A	Branch B
No. of SDS bands	≥2	<2
Pure enough	No	Yes
	Run IEx	Run GF

Figure 2. Sample decision point representation in P10. If the number of SDS-PAGE (sodium dodecyl sulfate polyacrylamide gel electrophoresis) bands is ≥2 and the target protein is pure enough, branch A should be taken (i.e., run ion exchange). On the other hand, if there are less than two SDS-PAGE bands, branch B should be selected (i.e., run gel filtration).

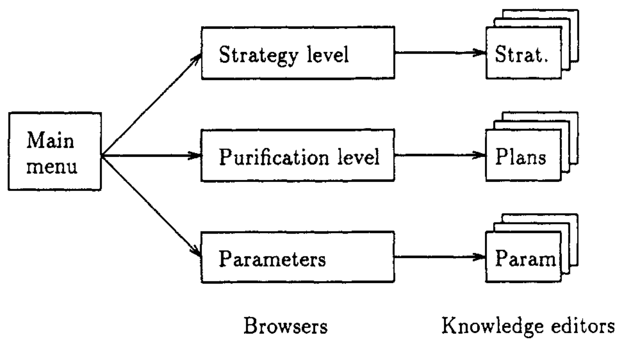

Figure 3. Dialogue structure in P10. The main menu (left) provides access to three categories of browsers (middle), which operate on sets of knowledge editors (right).

alternative routes depending on the results. In Figure 1b an example of such a partial plan for chromatography is shown. The actual path to take in the partial plan is determined at execution time by *decision points*. In this example, the intention is to apply gel filtration initially and then evaluate the outcome at the following decision point and determine the preferred branch in the partial plan (i.e., whether AC or IEx should be the next technique).

Currently, knowledge on how to select the appropriate path in the decision points is expressed in the form of decision tables. Figure 2 exemplifies such a decision table. (Naturally, other representations for decision points are possible, e.g., rules, but these tables seemed to be more cognitively valid and easy-to-use for the expert.)

Although P8 supports selection of running conditions for chromatography techniques, such knowledge is not captured by the model supported by P10. Instead, the selection of running conditions is hidden from the user of P10 and considered automatic at consultation time.

An important task for an expert system based on partial plans is to select which plan to follow given a purification problem. In the approach taken, each partial plan has a set of preconditions (formulated by the expert) which express in what situations it can be used. During consultation, the problem is to select among the applicable partial plans. A strategy mechanism is used to apply a certain strategy to distinguish between the applicable partial plans. The conceptual model of a strategy is a preference order for partial plans under some conditions. For instance, a different strategy (order of preference) is used when purity requirements are high than when safety is more important.

P10 was designed from this conceptual-domain model. It supports editing of preconditions, partial plans, decision points, and strategies. A graph editor was built for the partial plans, whereas various forms were designed to acquire preconditions, decision points, and strategies.

3.3. System Overview. When the P10 system is invoked, the *main menu* appears on screen. The main menu provides

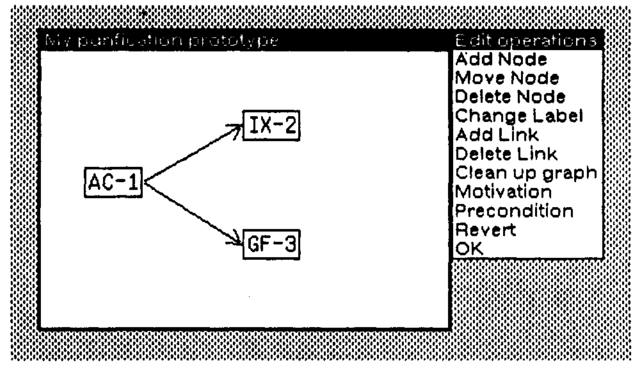

Figure 4. Partial plan editor in P10.

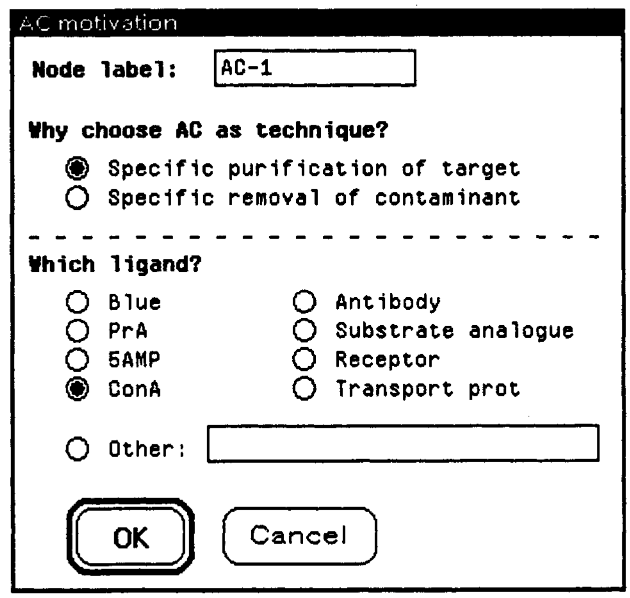

Figure 5. Sample motivation form for affinity chromatography.

access to major functions and browsers in the P10 system. From these browsers, it is possible to access a set of purification-specific knowledge-editing tools. These tools conform to the aforementioned conceptual-domain model. Figure 3 depicts the dialogue structure in P10.

Currently, P10 supports two levels of planning knowledge: *purification* and *strategy*. At the purification level, there is a browser which provides access to individual partial plans for procedural knowledge (see Figure 3). These partial plans can be entered and edited through the *partial plan editor*.

The partial plan editor supports a graphical language for partial plans. In the sample editor shown in Figure 4, the node labeled "AC-1" denotes the first purification technique to apply in this partial plan (affinity chromatography, AC). The graph constituting the partial plan can be modified using the edit operations at the right of the partial plan editor (see Figure 4). New purification steps are added by clicking on "Add Node". This brings up a pop-up menu for selection of technique. When the technique has been selected users are asked to justify their decision in a motivation form (see Figure 5 for a sample motivation form for affinity chromatography).

After the initial AC technique has been run, the outcome of this step is evaluated (at consultation time) in a decision point. Purification specialists can enter a set of conditions in P10 on how to proceed after the AC step. These conditions are edited by a *decision point editor* where different cases can be specified in terms of a set of parameters. As shown in Figure 4 affinity chromatography can, in this example, be followed either by ion exchange (IX-2) or by gel filtration (GF-3). Depending on the conditions specified in the decision point editor (which are later compiled into rules) and the actual

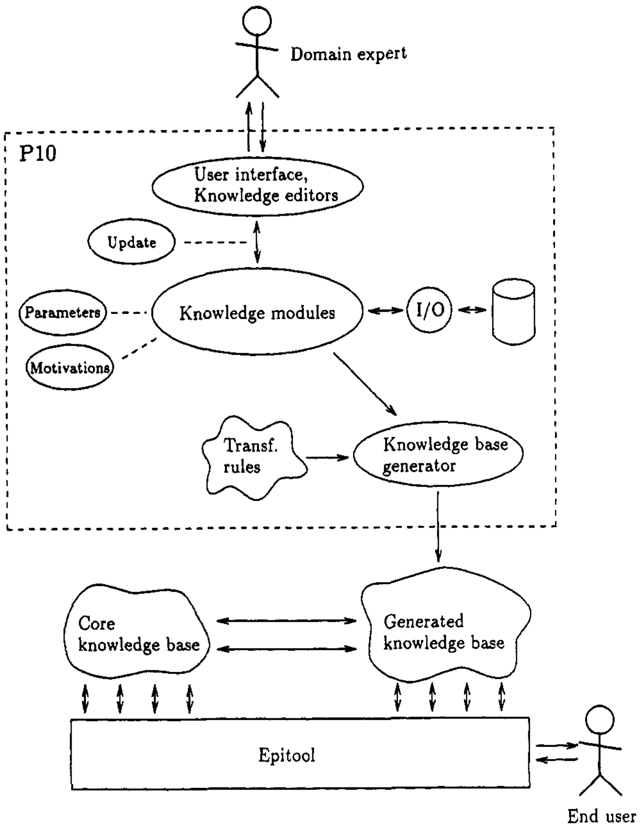

Figure 6. Overall architecture of the P10 system.

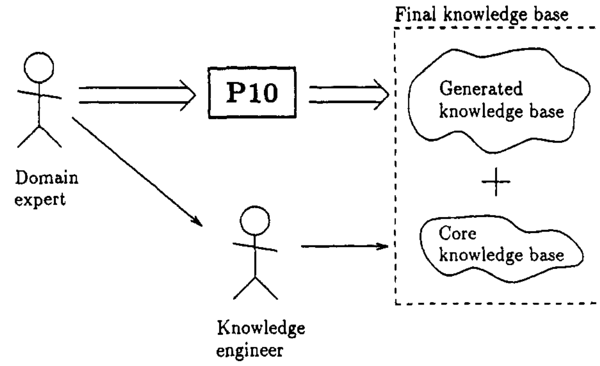

Figure 7. P10 acquires the bulk of the knowledge required directly from the domain expert and transforms it into a generated knowledge base. Especially difficult or rare cases are handled manually through a knowledge engineer, who develops a core knowledge base manually. The generated and the core knowledge base are merged into a final knowledge base.

(run-time) values provided to the expert system generated, the next technique recommended will be either ion exchange or gel filtration.

A *parameter browser* makes it possible to browse through and make changes to the (domain-specific) parameters used in different situations in the problem-solving process, e.g., in the decision points. A number of common parameters are predefined in P10 (e.g., sample volume, pH, and isoelectric point). If required, the user can define new ones.

Knowledge about how to select partial plans is input at the strategy level. From the *strategy browser* it is possible to access individual strategies, which can be entered/edited through a *strategy editor*. Strategies are essentially different orders of preference for partial plans, and it is possible to change these preferences through the strategy editor.

3.4. Architecture. The requirement that the KA tool should both provide a nice environment for experts and generate target knowledge bases from a domain-oriented specification means that such diverse issues as user interface, I/O, knowledge base generation, and knowledge base representation need to be treated by the same system. Therefore, the tool architecture needs special considerations. A critical constraint is the desire to provide the expert with means to run prototype versions of the target system. The way this problem is addressed will influence the architecture of the whole tool.

The P10 architecture comprises *knowledge editors* and *knowledge modules*. The motivations for the separation between editors and modules are (a) a desire to separate the surface interface from the deeper knowledge structures; (b) the fact that several editors can edit the same module, giving the user different "views" of the knowledge; and (c) the possibility to store entered knowledge on file in a convenient way.

The overall architecture of P10 is shown in Figure 6. A detailed description of the architecture is reported elsewhere.[13]

3.4.1. Knowledge Editors. The knowledge editors are a part of the user interface in P10 (see Figure 6). The usage ranges from simple tasks, such as handling of menus and forms, to more complex tasks, such as editing of partial plans in the form of graphs.

The editors are organized in a hierarchical tool box. The motivation for using this scheme is that new specializations to existing editors can be easily defined. Operations on editors can, henceforth, either be reused or further specialized.

An example of a knowledge editor is the *menu tool*, which is used to display menus and forms to the user. The layout is determined by a specification in each specialization of the tool. This tool is used in many different ways throughout the system, for instance the main menu is an instance of a menu tool and the motivation menu shown in Figure 5.

3.4.2. Knowledge Modules. Knowledge entered in editors is stored internally in P10, and this internal representation can be transformed into the format of an existing language/shell. A set of specializable *knowledge modules* is the basis for the internal representation in P10. P10 uses these knowledge modules for storage of "knowledge chunks". The motivation for this is to keep different kinds of knowledge (e.g., rules, decision tables, and procedural schemata) apart and to manipulate knowledge in a piecemeal fashion. The grain size of these knowledge modules are approximately that of a partial plan or a decision point.

When the expert has finished entering knowledge into the knowledge editor, the corresponding knowledge modules are updated accordingly by an *update mechanism* (see Figure 6). Note that the presentation in the editor and the actual representation in the module can be quite different. Like the editors, the modules are organized in an inheritance hierarchy. At the top, the *knowledge module class* can be found. At the next level of detail, we find modules that represent such structures as *decision tables* and *partial plans*.

The knowledge modules are used as a starting point for the generation of knowledge bases. The structures generated can be run together with manually written (core) parts of the knowledge base (see Figure 7). Although knowledge in the knowledge editors may be transferred to the corresponding knowledge modules and vice versa, the generation of knowledge bases from the knowledge modules is irreversible. In order to enable *re*generation of the knowledge base and enhance maintenance, it is possible to store the knowledge modules persistently on file (see Figure 6).

3.4.3. Generating Target Knowledge Bases. The production of knowledge bases is an important technical aspect of KA tools. The P10 system comprises a *knowledge base generator* for automatic generation of knowledge bases from the knowledge structures captured by P10. This knowledge base generator is generic in the sense that it is parameterized by a set of *transformation rules*. The rules essentially transform

```
(defrule decision-table-1
     ((col :in (A B C D)))

     (and premise strategy-seq)

     (transform decision-table-rule
          "create rule {rule-name} in backward is"
          "if {premise* and} then"
          "assert {scope-var}.SelectionStrategy to
                              new sequence {strategy-seq*, };"
          "end;"))
```

Figure 8. Simplified transformation rule (decision-table-1).

```
create rule Rule_61 in backward is
  if
    PartialPlan_5_Node_5.currentworld.NoOfSDSBand >= 2  and
    not PartialPlan_5_Node_5.currentworld.PureEnough
  then
    assert PartialPlan_5_Node_5.SelectionStrategy to
                     new sequence PartialPlan_5_Node_3;
end;
```

Figure 9. Rule (Rule_61) generated by P10 (Epitool). This rule has been generated through the transformation rule in Figure 8 (decision-table-1). (Other transformation rules have been used to generate the premise of Rule_61.)

knowledge modules into knowledge base constructs. Figure 8 shows a sample transformation rule (decision-table-1) from the P10 system. The rule decision-table-1 maps decision tables filled in by domain experts into appropriate Epitool[16] rules. A sample rule generated through decision-table-1 is shown in Figure 9.

The enabling condition for these transformation rules is adopted to the knowledge module representation, whereas the conclusion is typically a text template which represents constructs in the target knowledge base. (Note that the premise of decision-table-1 corresponds to columns in the decision table and that the conclusion is an Epitool rule template.) This rule format is primarily motivated by clarity; the transformation semantics are expressed in a well-known syntax. The transformation strategy adopted is exhaustive backward chaining for the transforming of knowledge modules into the constructs required. P10 comprises 40 transformation rules for knowledge base generation.

An alternative strategy for the generation of the target structures is reported by Musen.[17] This approach is based on database queries. A relational database is used as an intermediate representation between the user interface and the target knowledge base.

3.5. Remarks on P10. P10 is implemented on Common Lisp and runs in the Medley lisp environment, which is currently available on Xerox lisp machines, Sun workstations, and IBM RS/6000 workstations. Some basic facts about the P10 system are summarized in Table I. Note that the effort required to implement P10 was in the same order as the effort spent on the initial version of P8, which was approximately 2 person-months for the part of the knowledge base that P10 addresses[14] and 6 person-months for the whole P8 system.[6]

P10 has been evaluated by developing two expert systems. The first one was a *re*implementation of P8, and the second one was a system in the same class as P8 but with different purification knowledge. The reimplementation of P8 required 6 h, and the second system required 8 h. These figures should be compared to the original development effort for P8 (6 person-months for the complete system and 2 person-months for the parts generated by P10). The details of this evaluation are reported elsewhere.[14]

4. RELATED WORK

OPAL[10] is a KA tool which allows medical experts to enter and review cancer treatment plans. OPAL is essentially a domain-specific *knowledge editor* for a cancer-therapy expert system known as ONCOCIN.[18,19] The tool is based on a conceptual model of the domain. P10 and OPAL address different domains, but the overall task (planning) is similar. Both P10 and OPAL are based on a set of domain-dependent knowledge editors. Both P10 and OPAL use an intermediate knowledge representation (i.e., knowledge modules), but P10 differs from OPAL by using transformation rules for knowledge base generation rather than hard-coded functions.

Table I. Facts about P10

implementation environment	Common Lisp (Medley) and CLOS (PCL)
target language	Epitool
effort	6 person-months
expert time	60 h
code size	14 000 lines (Lisp) 6 000 lines (Core KB, partly from P8) 20 000 lines (total)
built-in rules	180
transformation rules	40

KAVE[20] is a KA tool that assists artificial ventilation experts in the formulation of rules for a decision support system in the area of respirator therapy. Experts encode and edit rules through rule-sheets specific to this domain. Several common parameters and rule types in the target system are supported. The KAVE system also includes a simulator where parts of the emerging knowledge base can be tested. Compared to domain-oriented tools such as P10 and OPAL; KAVE provides a conceptual-domain model at a slightly lower level since KAVE focuses on individual rules.

STUDENT[12] is a prototype tool which allows expert statisticians to enter and test statistical analysis strategies. STUDENT was initially based on a conceptual model induced from a statistical consultation system called REX.[21] In the STUDENT approach, it is possible to specify the knowledge acquisition *strategy* (e.g., active entry by the expert, querying the expert, and asking for sample solutions) for each piece of knowledge in the model.

KNACK[11] is a knowledge-acquisition tool that can be used by experts to develop systems for evaluation of different classes of design (i.e., different reporting tasks). KNACK can refine an initial domain model consisting of concept taxonomy, vocabulary, and procedures how to determine, compare, and propagate parameters. KNACK differs from P10 in that it tries to refine the domain model during interaction with the expert. KNACK uses sample reports and sample strategies to derive knowledge from rather than active knowledge entry.

5. SUMMARY AND CONCLUSIONS

Expert knowledge in protein purification planning can be modeled in such a way that a computer-based tool for knowledge acquisition can be developed. P10 is a KA tool that helps purification specialists to enter, edit, and review their planning knowledge as well as recipes, for instance in the form of partial plans and decision tables. The conceptual-domain model is sufficiently powerful for development of useful expert systems in the area, and yet it is sufficiently easy to use for purification specialists not trained in computer science, programming, or knowledge engineering.

Although it is possible to use commercially available shells for developing expert systems for protein purification (exemplified by the original implementation of P8), there are many advantages of using a KA tool. First, a user inferface for domain experts (i.e., nonprogrammers) is provided. Second, relatively low-level (symbol-level) structures, such as individual rules, can be hidden from the tool user and, thus, bring the knowledge expression language closer to the relevant domain. Third, KA tools help the developers to organize and structure the emerging knowledge base and aid the developers in coping with the complexity of large knowledge bases.

We are not prepared to completely remove the knowledge engineer from the development of expert systems (cf. Figure 7). Knowledge engineers are still important members of development teams; they can be seen as specialists in modeling and various forms of computer support (including expert system shells), rather than domain specialist. Thus, the experts' role in this setting is to enter domain knowledge into a KA tool, whereas the role of the knowledge engineer is knowledge modeling, KA tool development, and advising domain experts in knowledge structuring as well as various computer science issues.

Currently, P10 has been evaluated by developing two expert systems with promising results in terms of expert acceptance and productivity.[14] An obvious question is to what degree P8 facilitated the development of P10. Although actual program chunks could not be reused, the planning model on which P8 is based was to a large extent reused. The emerging P10 system highlighted several shortcomings in the model, and the development clarified many aspects of the model. Development of pilot expert systems has been suggested as a method of gaining a conceptual-domain model,[12] and one of the conclusions from P10 is that incremental development of KA tools can further improve the model.

One problem that P10 does not solve is the limited scope of the tool in terms of supported domains. For instance, P10 cannot be used to acquire knowledge about how to troubleshoot laboratory equipment. Hence, domain-oriented tools like P10 perform well, but only within the framework of the supported conceptual-domain model. Our current effort is directed toward finding means to specialize KA tools through meta-level tools for a broad variety of domains with quite different conceptual domain models.[22]

ACKNOWLEDGMENT

Thanks are due to Sture Hägglund, Kristian Sandahl, and John Egar for valuable discussions and comments on drafted versions of this article. John Brewer was the domain expert. This work has been supported by the National Swedish Board of Technical Development (STU) and by a grant from Pharmacia LKB Biotechnology AB.

REFERENCES AND NOTES

(1) Eriksson, Henrik; Sandahl, Kristian; Brewer, John; Österlund, Bengt. Reactive planning for chromatography. *Laboratory Information Management*; in press.
(2) Boose, John H. A knowledge acquisition program for expert systems based on personal construct psychology. *Int. J. Man-Mach. Stud.* **1985**, *23*, 495–525.
(3) Kahn, G.; Nowlan, S.; McDermott, J. MORE: An intelligent knowledge acquisition tool. In *Proceedings of the Ninth International Joint Conference on Artificial Intelligence*, IJCAI'85: Los Angeles, CA, 1985; pp 581–584.
(4) Eshelman, Larry; Ehret, Damien; McDermott, John; Tan, Ming. MOLE: A tenacious knowledge-acquisition tool. *Int. J. Man-Mach. Stud.* **1987**, *26* (1), 41–54.
(5) Marcus, Sandra; McDermott, John. SALT: a knowledge acquisition language for propose-and-revise systems. *Artif. Intell.* **1989**, *39*, 1–37.
(6) Eriksson, Henrik; Sandahl, Kristian. Knowledge-based planning of experiments in a biochemical domain–membrane protein purification. In *Proceedings of the Second Scandinavian Conference on Artificial Intelligence*, SCAI'89, Tampere, Finland, June 1989.
(7) Ananda, A. L.; Foo, S. M.; Gunasingham, Hari. Knowledge representation using an augmented planning network: Application to an expert system for planning HPLC separations. *J. Chem. Inf. Comput. Sci.* **1988**, *28* (2), 82–86.
(8) Newell, Allen. The knowledge level. *Artif. Intell.* **1982**, *18*, 87–127.
(9) Neale, Ian M. First generation expert systems: A review of knowledge acquisition methodologies. *Knowl. Eng. Rev.* **1988**, *3* (2), 105–145.
(10) Musen, Mark A.; Fagan, Lawrence M.; Combs, David M.; Shortliffe, Edward H. Use of a domain model to drive an interactive knowledge-editing tool. *Int. J. Man-Mach. Stud.* **1987**, *26* (1), 105–121.
(11) Klinker, Georg.; Bentolila, Joel; Genetet, Serge; Grimes, Michael; McDermott, John. KNACK—report-driven knowledge acquisition. *Int. J. Man-Mach. Stud.* **1987**, *26* (1), 65–79.
(12) Gale, William A. Knowledge-based knowledge acquisition for a statistical consulting system. *Int. J. Man-Mach. Stud.* **1987**, *26* (1), 55–64.
(13) Eriksson, Henrik. A Study in Domain-Oriented Tool Support for Knowledge Acquisition. Linköping Studies in Science and Technology, Licentiate Thesis 181, LiU-Tek-Lic 1989:21, Linköping University, 1989.
(14) Eriksson, Henrik; Specialized knowledge acquisition tool support compared to manual development—a case study. In *Proceedings of the Seventh IEEE Conference on Artificial Intelligence Applications*, Miami, FL, Feb 1991.
(15) Aikins, Janice S. Prototypical knowledge for expert systems. *Artif. Intell.* **1983**, *20*, 163–210.
(16) *Epitool Development Environment Reference Manual*; Epitec AB: Linköping, Sweden, 1989; Version 4.
(17) Musen, Mark A. *Automated Generation of Model-Based Knowledge-Acquisition Tools.* Morgan-Kaufmann Publishers, Inc.: San Mateo, CA, 1989.
(18) Shortliffe, Edward H.; Scott, A. Carlisle; Bischoff, Miriam B.; Campbell, A. Bruce; van Melle, William; Jacobs, Charlotte D. ONCOCIN: an expert system for oncology protocol management. In *Proceedings of the Seventh International Joint Conference on Artificial Intelligence*, IJCAI'81: Vancouver, Canada, 1981; pp 876–881.
(19) Hickam, David H.; Shortliffe, Edward H.; Bischoff, Miriam B.; Scott, A. Carlisle; Jacobs, Charlotte D. The treatment advice of a computer-based cancer chemotherapy protocol. *Ann. Intern. Med.* **1985**, *103* (6 pt 1), 928–936.
(20) Shahsavar, Nosrat; Gill, Hans; Wigertz, Ove; Frostell, Claes; Matell, Georg; Ludwigs, Ulf. KAVE: a tool for knowledge acquisition to support artificial ventilation. *Comput. Methods Programs Biomed.* **1991**, *34* (2/3), 115–123.
(21) Gale, William A. REX review. In *Artificial Intelligence and Statistics*; Gale, William A., Ed.; Addison-Wesley: Reading, MA, 1986; Chapter 8, pp 173–227.
(22) Henrik Eriksson. Meta-Tool Support for Knowledge Acquisition. PhD Thesis 244, Linköping University, 1991.