6th Banff Knowledge Acquisition for Knowledge Based Systems Workshop, 1991

CREATIVE KNOWLEDGE ACQUISITION: AN ANALYSIS

Timothy C. Lethbridge

Department of Computer Science
University of Ottawa
Ottawa, Ontario, Canada K1N 6N5
(613) 564-8155 tcl@csi.uottawa.ca

ABSTRACT

We discuss the idea of 'creative' knowledge acquisition (KA) by which we mean generation of primarily new knowledge using a knowledge-based system. We present a conceptual analysis outlining the properties of creative KA tasks. Then we develop a process model of creative KA based on a knowledge enterer's concept-elucidation objectives. We use the conceptual analysis and the process model to justify some recommendations relating to the knowledge representation and user interface of a creative KA system. The recommendations involve helping users generate, record and effectively sequence their thought processes. We are implementing many of the recommendations in version 4 of our CODE system.

INTRODUCTION

When surveying use of our lab's knowledge acquisition (KA) system, CODE (Skuce et al, 1989, Skuce 1991a), over the past few years, one major point in particular strikes us: CODE's use as a *creative assistant* has often been more valuable than the knowledge bases we have built/\..

For example, during a major industrial application of CODE to software requirements analysis, it took us many months to establish a substantial knowledge base and to convince project team members to use it as the central conceptual repository and documentation source (Skuce 1991b). Almost from the beginning of our involvement however, team members were able to benefit from the process of building the knowledge base: By the manipulation of conceptual structures through an effective user interface, complex issues with which they had previously struggled were resolved.

We have taken to creating very small knowledge bases with CODE to develop ideas in our research (including for this paper). The knowledge bases frequently become obsolete as ideas rapidly evolve; but that does not matter: the value has come from use of a KA system as an analysis tool to get us over the creativity 'hump'. In one domain, the design of our core knowledge representation, we have created knowledge bases from scratch three separate times in order to make use of the creative potential of the process.

The focus of this paper is the problem of how to optimize the creative potential of KA systems, whether the resultant knowledge base is to be maintained or not. We start by analysing the creative aspects of KA so we can understand what needs optimizing. Then we develop a hypothetical model to describe the process a user goes through when using a KA system creatively. Finally we use an example user session to make some suggestions about creative KA. These suggestions affect 1) knowledge representation, 2) ontology and 3) user interface.

1. CREATIVE KNOWLEDGE ACQUISITION

The general meaning we give to 'knowledge acquisition' (KA) is any process leading to the addition or refinement of knowledge (i.e. concepts) in a knowledge base. We prefer this broader meaning to that restricted primarily to 'expert systems' or 'expertise' (Gaines 1989); we do not however consider a task as common as writing a document to be KA, unless it is part of a process leading to knowledge base building.

By *creative* knowledge acquisition we mean KA of primarily *new* knowledge, due to direct manipulation of and feedback from conceptual structures evolving in parallel in minds and in a knowledge base. It is a largely individual process, comparable to creative writing, but with much more potential for tool support than possible with natural language composition.

By *new* knowledge, we mean knowledge not present in the mind of the domain expert or in the knowledge base before the KA session (although less refined knowledge may have been present). New knowledge is originated (i.e. created) in mind or machine using processes (deductive, inductive, intuitive etc.) that detect distinctions or patterns in pre-existing knowledge. Processes that are not considered to be originating knowledge include observing, translating, loading, decompressing, inheriting and combining (as long as these processes lead to nothing distinctive being observed about the knowledge).

We consider knowledge to be composed of *concepts* which are mental or mechanical representations of anything that can be 'thought' about. Two important types of concepts are independent concepts and the relations between them that we call *property occurrences*.

The early stages of creative KA can typically be described as 'individual brainstorming into a knowledge base', although brainstorming has a connotation of haphazardness that should not be applied to all creative KA.

Two particular aspects of KA-related literature are relevant to creative KA. One is the study of serendipity, the discovery of knowledge one is not looking for (Roberts, 1989). Creative KA involves more than just enhancing chances for serendipity: One *often is* looking for specific knowledge one does not possess. The other useful aspect of the literature is the use of KA in design (Reich, 1990; Mark, 1990); designers have much to benefit from the enhancement of creative potential.

2. PROPERTIES OF KA TASKS: COMPARING GENERAL AND 'CREATIVE' KA

Before we describe our process model for creative KA, we present a conceptual analysis comparing creative KA to general KA. The analysis is based on classifying KA tasks with the objective of identifying where we might target technique improvements. Other KA classification literature tends to be organized by subtask, technique, tool, application area, assessment criterion etc. (e.g. Wetter and Woodward 1990)

Our analysis is in the following ten subsections, each describing a semi-independent property on whose scale we can place any KA task. Following each property we indicate in what way creative KA tends to be specialized. Our model, described later, is developed with these specializations in mind, but may be applicable in more general KA situations.

The properties are listed in order, such that those that most distinguish creative KA are placed first. The properties are phrased as questions to be posed about each task; under each question is a list of points outlining a rough scale in which the answer should fall. The points listed for each property are not meant to be the only possibilities, nor are they meant to be mutually exclusive or necessarily form a logical progression. Within a given KA task, a property typically involves a range of points on its scale, which may apply at once or at

different times; this implies that some aspects of a task may be considered creative and some may not.

Property 1. At what level of clarity does the knowledge exist in the minds of the domain experts, prior to knowledge acquisition?

a) The knowledge is considered concrete and factual: The expert(s) know (have conscious access to) what they know. The KA process merely transfers the knowledge between human and machine.

b) The knowledge is is fuzzy, unsystemized etc: The experts may know what they know, but generally not precisely, and they generally cannot clearly characterize the knowledge which may be undeveloped 'ideas'. Typically, mental heuristic knowledge is at this level of clarity. The KA process crystalizes the knowledge.

c) The knowledge is latent, experiential etc: The expert(s) do not know what they know. The KA process draws out the knowledge.

d) The knowledge does not exist. The KA process generates new ideas by providing a medium for the organization of thoughts.

Answer d most strongly characterizes creative KA, although answer b also frequently applies. The knowledge is already 'created' in form a, and form c requires emphasis on elicitation techniques that detract from spontaneous creativity (see property 4 below).

Property 2. How is the knowledge originated?

a) Most of the knowledge originates as the result of computational processes (e.g. it is extracted from raw data using machine learning techniques).

b) Most of the knowledge originates as the result of human thought processes.

Creative KA falls at the latter end of this spectrum; creation of the knowledge is done non-algorithmically by human thought processes (frequently intuitive, and generally poorly understood by today's science).

Property 3. What formulation[1] of the knowledge is performed prior to its entry into the KB?

a) The knowledge is originally formulated as natural language documents, typically without the intent to create a KB.

b) The knowledge is gathered from the expert(s) in one form (natural language notes, diagrams, results of elicitation tools etc) and is reformulated when entered into the KB. The gathering from experts is done with the intent of adding knowledge to a KB.

c) None, the knowledge is elicited, formulated and stored as an indivisible interactive process.

Creative KA follows the pattern of point c. The idea is that feedback from the interactive process stimulates creativity.

---

1 By *formulating* knowledge, we mean representing it in some medium.

Property 4. What are the primary elicitation techniques used?

a) None. The knowledge is entered into the KB, with reports of errors, ambiguity and inconsistency as the only feedback. We do not consider such feedback alone to be an elicitation technique, although it can be made to trigger concept structuring (d below).

b) Discussion techniques (interviews, case studies etc.) with little machine involvement.

c) Concept structuring techniques (e.g. repertory grid, interpretive structural modelling) performed without machine assistance.

d) Exemplar-based techniques, usually performed with machine assistance.

e) Concept structuring techniques performed with machine assistance. The machine guides much of the KA process.

f) Agenda maintenance techniques: The machine attempts to help the user decide what to tackle next.

g) Alternate perspectives feedback: The user guides the KA process, receiving elicitatory stimulus by looking at alternate perspectives on knowledge already entered. The user's natural ability to make inferences in ways that could not be forseen by the machine is reinforced.

Although any machine assisted technique can assist creative KA, technique g, where the user is most in control, best characterizes our vision of creative KA.

There is extensive discussion of elicitation techniques in the literature. A decomposition comparable to the above may be found in (Gaines, 1990, p 9-19). A good survey of those applicable to expert systems is (Neale, 1989). Interpretive structural modelling is discussed in (Warfield, 1976).

Property 5. Who performs the formulation of the knowledge into the machine knowledge representation?

a) The machine itself, largely automatically from data, natural language etc.

b) The knowledge engineer acting as an intermediary who works with domain expert(s) away from the machine and then goes to the machine to enter knowledge.

c) The knowledge engineer acting as a facilitator who works with domain expert(s) in front of the machine.

d) The knowledge engineer, after having developed domain expertise: The true domain expert(s) act as critics.

e) The domain expert(s), with the knowledge engineer acting as active critic. The KE is present, guiding the process, but the domain expert(s) have control over the machine interaction. Semi-intelligent software may substitute for the KE.

f) The domain expert(s) on their own, with a knowledge engineer (or software) performing occasional critiquing.

g) The domain expert(s) acting as their own knowledge engineers.

Creative KA leans towards methods f and g, although all of c through g are possible. In one project we had experiences with methods b through f, and our system demonstrated its creativity-stimulating potential to the greatest extent using method d. We believe, however that this was an artifact of our having much greater tool expertise than the domain experts

did when they used methods e and f. A major thrust of our research is to increase the usability of creative KA systems in order to make methods f and g the norm.

Property 6. What type of expert collaboration is involved?

a) The process is characterized by group brainstorming.

b) The process involves parallel formulation of detailed alternate perspectives on the 'same' knowledge, followed by 'merging' and conflict resolution. This differs from brainstorming in that detail is generated prior to resolution.

c) The process alternates individual formulation of knowledge in areas where each expert has greatest expertise, followed by negotiation to resolve conflicts and disagreements.

d) The process involves individual responsibility for areas of the knowledge base with minimal overlap.

e) The knowledge source is primarily one individual.

We have used all methods except b, and although method a tends to stimulate the most creativity, little refinement of the knowledge is possible. Our concept of creative knowledge acquisition involves only the individual-work phases of methods b to e.

Property 7. What is the primary purpose of performing the knowledge acquisition?

a) To provide input to a program for use in its reasoning processes. This is the classical expert systems purpose.

b) To convey the knowledge to other humans: E.g. for a tutorial, documentation or retrieval system that performs little reasoning.

c) To define terms. E.g. when developing a domain's terminology.

d) To analyse, or better understand a concept; to explore thoughts. The knowledge base may be only temporary, or kept as a record of thinking (the secondary purpose is often b or e).

e) To design something: The knowledge forms requirements and/or specifications for the thing. It later becomes documentation about the thing and/or is actually used in processes that create the thing.

Creative KA can have any of these purposes, but those later in the list are more typical: Creative knowledge is gathered more for *immediate* problem-solving than with the delayed objective of solving problems *with* the knowledge. The process of representing knowledge tends to be more important than the knowledge. The addition to knowledge in people's minds tends to be more important than the addition of knowledge to the KB.

Property 8. What is the scale of the knowledge base?

Typically creative KA is performed on small discrete 'problems' (property 7, above) and created by individuals (property 6) hence knowledge base size is naturally limited. KB sizes have been limited by technology, but many people believe in a 'bigger (within limits) is better' philosophy, with long-term projects that attempt to capture much of the detail in a domain. While we laud such efforts, we believe in the need to develop special technologies for 'personal KB's. For a somewhat imperfect analogue, consider the differences between typical use of a large database system and typical use of a spreadsheet program.

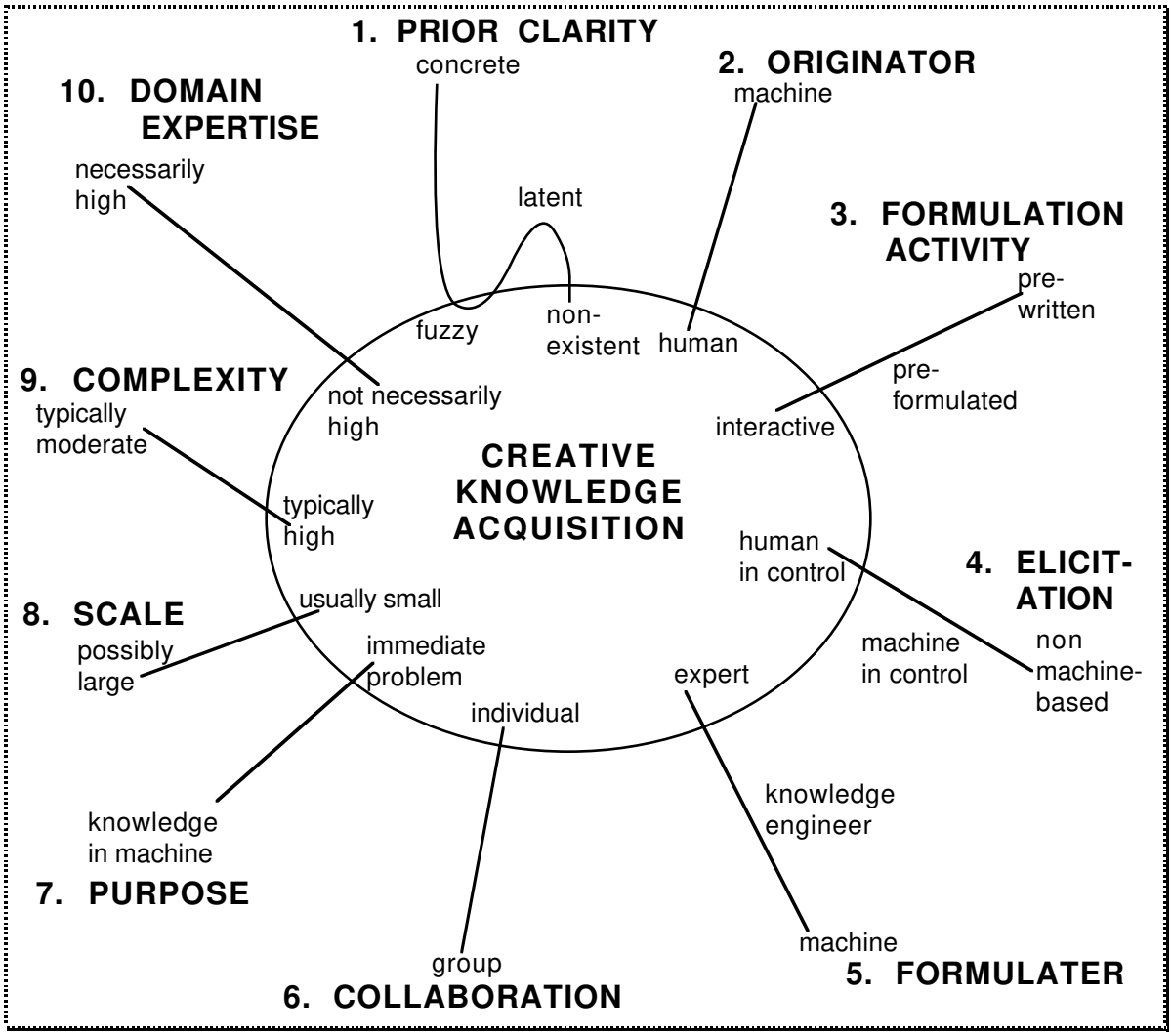

Figure 1: Semi-independent properties of knowledge acquisition tasks, showing how we characterize 'creative' KA. Each property is represented by a scale; properties of creative KA tasks tend to be specialized with values near the ends of the scales that are in the central ellipse.

Property 9. How complex is the knowledge acquired?

Complexity measures such things as interconnectedness, non-linearity, dimensionality, degree of meta-representation, degree of projection etc.

While smaller in scope, we see creative KA as tending to deal with greater knowledge complexity than mainstream KA. Complex problems are what people need creative KA for. Only in small KA efforts can time be afforded to expand concepts to extreme levels of detail.

Property 10. What is the level of expertise of the domain 'experts'?

While other types of serious KA require someone who truly knows the concepts (or has recorded them in documents), creative KA may involve non-expert knowledge source persons. Such persons may have peripheral knowledge and be using KA to expand their expertise (e.g. developing theories). Alternatively, such persons might be using KA to formulate 'naive' suggestions to real experts (in the same manner that someone might design their own house).

3. A PROCESS MODEL OF CREATIVE KA

In this section we describe our hypothetical model of the creative KA process. The model has two related components: 1) a model of knowledge-base editing objectives that may shed some light on how to assist a user in sequencing a creative KA task, and 2) a model of informal knowledge that can help explain how to prevent the user from getting bogged down too early in details.

We have developed the model in order to explain observed phenomena and to provide a framework for ideas about the improvement of creative KA. We make no claims about the cognitive correctness of our model; others may wish to persue such research. For us the model is primarily an aid to thought that we have found useful. In section 4 we will present several recommendations, several of which arise from the model.

Our model is not a 'user model' in the standard sense of the term. There is a literature on 'user models' and 'information source models' (e.g. Thost, 1989). We primarily differ in that we are not dealing with 'machine conceptualizations of particular users'; we do not emphasize automatically adapting interactions to such models. Instead we have a single general model that we use 1) to structure our own analysis, and 2) to suggest possibilities for facilities the user could call on, under his or her control, to aid memory and planning of a creative KA task.

KA as achieving mental 'concept elucidation' objectives

We model the mental state of a creative KA system's user as a loose collection of *concept-elucidation objectives*. Concept elucidation can be thought of as clarifying a concept, expanding the level of knowledge associated with it, building a conceptual substructure centred around it, or 'formalizing' it (see next subsection). Experience leads us to believe that creative productivity can be enhanced if the user arranges the objectives in a hierarchy.

We consider an objective to be a mental entity; the process of its achievement is performed simultaneously in mind and KA system. On the KA system side, the user builds machine conceptual structures; in an ideal system these correspond to the mental conceptual structures being built at the same time. We envisage the machine representation process assisting the parallel mental process — maintaining the network of complex details of which the mind would otherwise lose track. In other words, we see a sort of mind-machine symbiosis.

The hypothesized collection of objectives develops in steps as follows: The user mentally searches for, and seizes on, an idea for a conceptual structure that, if represented, would help achieve a particular elucidation objective. Representing the idea becomes a sub-objective. The sub-objective, and possibly parent objectives, are removed if the representation of the idea's central concept satisfies them, i.e. if the user feels that no further elucidation is necessary, that the conceptual structure has been completely built.

Steps of the preceding form take place iteratively until there are no more objectives to achieve. The objective collection grows and shrinks, and the conceptual structures in mind and machine become elucidated.

At each step, just one concept is actively changed in mind and machine (although passive consequences caused by effects such as inheritance usually also occur). It is important not to confuse the objectives with the conceptual structure being built, even though in our model each step involves one objective and one central concept.

To summarize the above, at each step:

- An objective is added in the user's mind. This may be a sub-objective of any other objective. We call it the*current objective*; the sequence of current objectives over time corresponds to the user's stream of consciousness.
- A user interaction may be made to commence satisfaction of that objective on the machine side. This may be nothing more than recording an empty independent concept. Other possibilities include adding a property occurrence, deleting a concept etc.
- The objective and a chain of its parents may be removed. Removal of an objective implies it has been achieved by the user interaction. When the last objective is removed, the task is complete.

We do not intend by our idea of a loose 'objective hierarchy' to suggest that the minds of creative KA system users work in a Prolog-like manner. To avoid this perception, it has been suggested that a blackboard architecture might be a superior model. We agree that such an approach could be useful, but it ignores the creative advantage we ascribe to structuring creative thinking hierarchically. Further research may shed some light on whether our view is justified.

KA as the process of increasing concept formality

We describe the process of elucidating a concept in the machine as making it more *formal* (Lethbridge, 1991a). We use the word 'formal' with its general meaning of 'having shape or form'. A concept becomes more formal if more links are made from it to other concepts and/or if other concepts it is linked to become more formal. Two concepts are *linked* if a property occurrence of one refers to the other (directly or indirectly).

In our model, the user's decision that an objective is achieved is based on his or her perception that the objective's central concept has reached a desired level of formality. The desired level of formality should presumably be higher for concepts of a higher-level objective; in fact if a user elucidates peripheral concepts to an excessive level of formality, he or she can be said to be *digressing* (helping to prevent this is a separate research problem).

When many concepts are first represented in the machine, they are *informal* since they lack links; they typically contain only descriptive text. We envisage the early phases of creative KA as involving the addition of large numbers of informal concepts very quickly (one step every few seconds). In later phases we see greater emphasis on increasing formality as objectives become more nearly achieved. In many applications, we see KA eventually moving out of a creative phase into one of validation and debugging — at such a time we envisage use of a somewhat different set of KA tool facilities from those used to enhance creativity. We also envisage a KA task moving from creative mode to validation mode and back again, several times.

How can we use the model?

There are three primary aspects of the creative KA process where our model gives rise to ideas for machine KA assistance: 1) Helping the user generate the initial idea that forms an

objective, 2) Allowing the user to record his or her stream of consciousness without impedance, 3) Helping the user to organize his or her work:

- Generating the idea: We can provide initial knowledge, feedback and ways of viewing the knowledge.
- Recording the stream of consciousness: We must allow the user a) to follow his or her preferred path without distraction, and b) to represent anything that comes to mind as quickly and elegantly as possible.
- Organizing work: We can help the user to backtrack and to remember what her or his objectives are.

In section 4 we will outline some recommendations that arise from these points, but first we present a slightly more detailed look at the third point.

Organizing the creative KA task

We can attempt to keep a machine model of the user's elucidation objectives by maintaining a history of the corresponding concepts the user edits. By comparing the relationships of concepts in this history we can arrange it in a hierarchy ordered by sub-objective. Various aspects of the user interface can manipulate this hierarchy to assist the user:

- Sequencing assistance: We belive that helping the user to progress in a logical sequence may enhance creative potential. A logical sequence would consist of always progressing by creating a sub-objective of the current objective or by backtracking in the objective hierarchy when the desired level of formality is reached. A visual display of the objective hierarchy could help keep the user on track.
- Undoing and exploration assistance: The backtracking mentioned in the previous point was non-destructive; after completing work, the user steps back to the most recent incomplete objective. An alternate use of the objective hierarchy would be destructive backtracking, i.e. an extension of the standard 'undo' UI technique.
- Incompleteness detection: After extensive knowledge base editing, it can be useful to review leaves of the objective hierarchy, to see if there are any possible incompletenesses.

Although we have not yet applied these techniques to knowledge base editing, preliminary experience with some of these ideas in a CASE environment (an enhanced Smalltalk browser) has convinced us of their usefulness.

Generality of the model

Although we described the above model using the perspective of a knowledge source person working in intimate comunion with a creative KA system, we do not discount other possibilities. For example there could be a knowledge engineer involved (property 5), or there could be several knowledge sources (property 6). Such possibilities do not invalidate our model, they merely complicate it. For example, the objective hierarchy would be the abstract 'consensus' objective hierarchy (the idea of an objective hierarchy is abstract anyway, so making it more so is not of great concern!)

4. AN EXAMPLE USING THE PROCESS MODEL

In this section we step through an example creative KA session, making suggestions about how to prevent various creativity problems. Figure 2 (which is split into several parts) shows the session's progress.

We assume the presence of a basic knowledge representation (KR) system with a frame-like knowledge structure and multiple inheritance. Our suggestions describe facilities needed above and beyond this. Some of our suggestions are well-known, others less so.

The suggestions relate to three areas of KA system requirements: 1) knowledge representation, 2) ontology and 3) user interface. Although these tend to be researched separately, we believe it is important to consider them together for the purpose of solving problems such as creativity enhancement.

The list of recommendations is *not* intended to be comprehensive. In particular, it is not at all concerned with the following problems we believe are orthogonal to enhancing creativity.

1) Getting the knowledge right (making it consistent, correct, unambiguous or complete).
2) Ensuring the knowledge gathered is useful, i.e. preventing digression or excessive detail.
3) Following standards of knowledge representation style, e.g. to help keep knowledge bases consistent and comparable.

Some of our recommendations *may* help solve these problems, but some might in fact hinder such solution. For example, we discourage the user from stopping to worry about the consequences of entering a piece of knowledge in the early stages of KA. Such stopping may help get the knowledge right and ensure it follows standards, but it can hinder creativity.

We have made the observations described in this section repeatedly in a number of small KA projects. The particular example is synthetic and extremely over-simplified, but the domain is that of a real project.

Our research work is ongoing, so we expect to further refine the ideas in the future. Some of the recommendations use the terminology and conceptual framework of CODE, but we believe they can readily be transferred to the context of any other system. For greater detail on some issues see (Lethbridge, 1991b). For most of the recommendations, there exist compliant KA systems; we believe however, that no system available today complies with all recommendations.

Setting the scene

Imagine the following: A designer has in mind a new type of telephone service. He or she mentally develops the idea, but then realizes it has many complexities that are hard to organize in natural language, diagrams or other conventional media. He or she also wants to be able to explore options and receive automatic feedback about the idea. For these and other reasons it is decided to use a 'creative KA' tool.

Step a: Creating a blank slate

The user starts with a top-level objective

> Suggestion a1: Multiple interaction paradigms should be available, including graphical and textual; but where possible the paradigms should share homologous appearance, operations, feedback modes etc.
>
> *In figure 2a, the user is using a graphical view (see Joseph, 1989) that operates like a drawing program.*

Textual views organized alphabetically or hierarchically should also be provided. These might operate like an outline processor The idea is to give the user maximum flexibility to record thoughts.

Suggestion a2: It should be possible to construct various types of 'maps' of parts of the knowledge base.

In figure 2a, the user is operating on an empty subset of the KB where any independent concept or property (relation) may be explicitly displayed.

In order to cater to the user's thought patterns, the user should be able to easily request display of any pattern of concepts and relations. We call such patterns 'knowledge maps' (Lethbridge, 1991b). Techniques for specifying knowledge maps include selection through browser or 'hyper' links, and arbitrary logical expressions to select or hide knowledge. Types of knowledge map include standard isa hierachies, property hierachies (suggestion h1), hierarchies of arbitrary transitive relations (e.g. 'part-of'), semantic networks, and many others.

Suggestion a3: The user should not be forced to name a concept, in fact the user should not be forced to provide any information he or she is not ready to provide.

In figure 2a, the name has not yet been provided for the new concept.

Naming is particularly distractive, requiring thought processes entirely different from those of concept creation. The user has to either choose the correct term from those that exist or create a neologism; either process can be difficult and the result may be unnecessary naming of an abstract subtlety. Hurriedly concocted names can 'stick' and cause confusion later. Some systems unfortunately require concept naming for 'internal' identification purposes. Position in a window, position in conceptual context and/or an automatically generated label or icon should however allow the user to distinguish the concept from others.

Suggestion a4: Maintain separate concepts for different levels of representation.

In figure 2a, a separate metaconcept *might be created to represent the description of the new empty concept.*

Such a concept would maintain meta-knowledge properties such as 'who did it', 'orientation on graph', etc. This separation of concerns can help keep the conceptual structure clear in the user's mind, but more important from a creativity standpoint is encouraging the proper application of *metarepresentation* (keeping the prepresentation of a representation of x separate from the representation of x). We have found poor understanding of this idea to be one of the greatest sources of hidden difficulty when creating a knowledge base.

This is one of a series of suggestions we will make regarding the importance to creativity of allowing the user to handle subtleties of representation. In general, the user should be able to readily distinguish between subtlely distinct things so as to be able to 'say different things' about them.

We believe there are too many subtle representation issues still to be researched for knowledge representation 'standards' to be developed yet.

Suggestion a5: Allow any level of formality of concept representation.

In figure 2a, the concept is completely informal except for invisible links to a metaconcept (suggestion a4).

See section 3 for a discussion of informality.

Step b: Naming a concept

The user has the lower-level objective to name the concept (despite suggestion a3 above, naming in this case is easy). Once naming is done, the naming objective is removed from the objective hierarchy. Naming is but one of a vast number of 'potential objectives' opened up by the top level objective, naming is merely the first that occurred to the user.

Suggestion b1: The interaction should be non-modal.

In figure 2b, the user should have been able to specify the name without requesting a prompt.

All basic interactions should be one-step processes so the user can change his or her mind, suspend an operation etc.

Suggestion b2: Knowledge manipulation, where possible, should be direct rather than performed using some command-based interface.

In figure 2b, the user should have been able to simply type over 'NNCF telephone'.

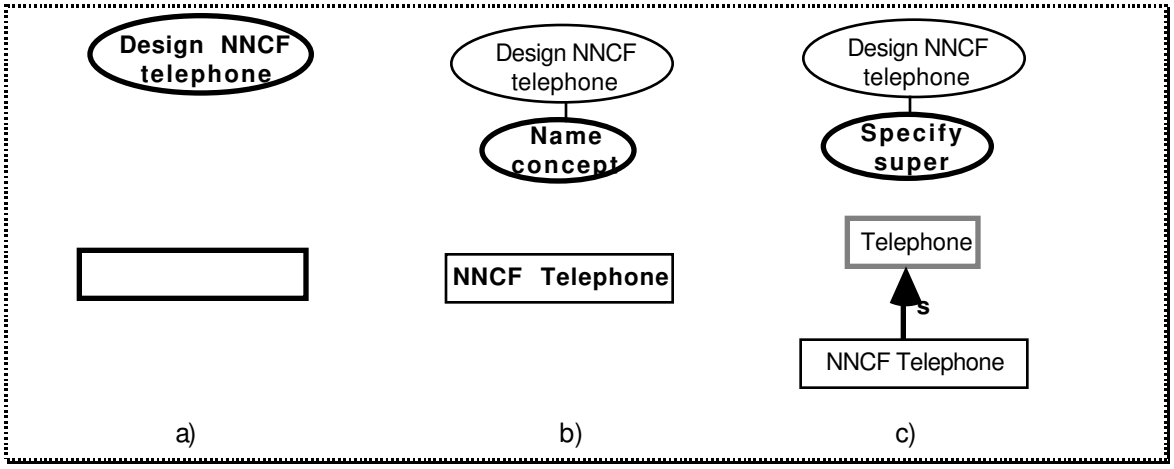

Figure 2 a,b,c (part 1 of 5): Each lettered section represents a user's step towards elucidating his or her idea for a telephone design. The ellipses represent a hierarchy of typically subconscious user objectives, the bold objective being current (the system may attempt to track this hierarchy, but it is not actually seen on the screen). Below the ellipses are conceptual graphs of the growing knowledge base as drawn by the user. Knowledge added to satisfy each objective is in bold (entirely new) or halftone (pre-existing knowledge merely added to this graph).

Step c: Specifying a superconcept

The user wants to say what the 'NNCF telephone' *is*. A superconcept link is established to the concept Telephone, which already exists in the knowledge base.

Suggestion c1: An ontology must be provided as a starting point for the KA task.

In figure 2c, the concept 'telephone' already existed in the ontology.

In the general case there must be sufficient concepts in the ontology to prevent the user from being forced to digress excessively (Skuce, 1990).

Suggestion c2: Effective ways to make references must be provided.

In figure 2c, the user should be able to locate 'Telephone' without excessive delay.

Making references can be done by a) locating concepts (via pointing, menus, wildcard patterns etc.) and/or b) composing a concept reference from other references (e.g. 'lawnmower of suburbanite'). It can be facilitated textually or graphically. Being able to refer to a recent history of references can also speed reference-making.

Step d: Preparing for a related concept

It has occurred to the user that there is a microprocessor involved somehow. The first step is to record a new concept.

Suggestion d1: The user should be able to open as many views into the knowledge as desired, in different windows. Furthermore, the user should be able to tie windows together to form browsers.

In figure 2d, the user may want to separate the microprocessor specification from that of the telephone itself. The user may want to explore existing knowledge about microprocessors before proceeding.

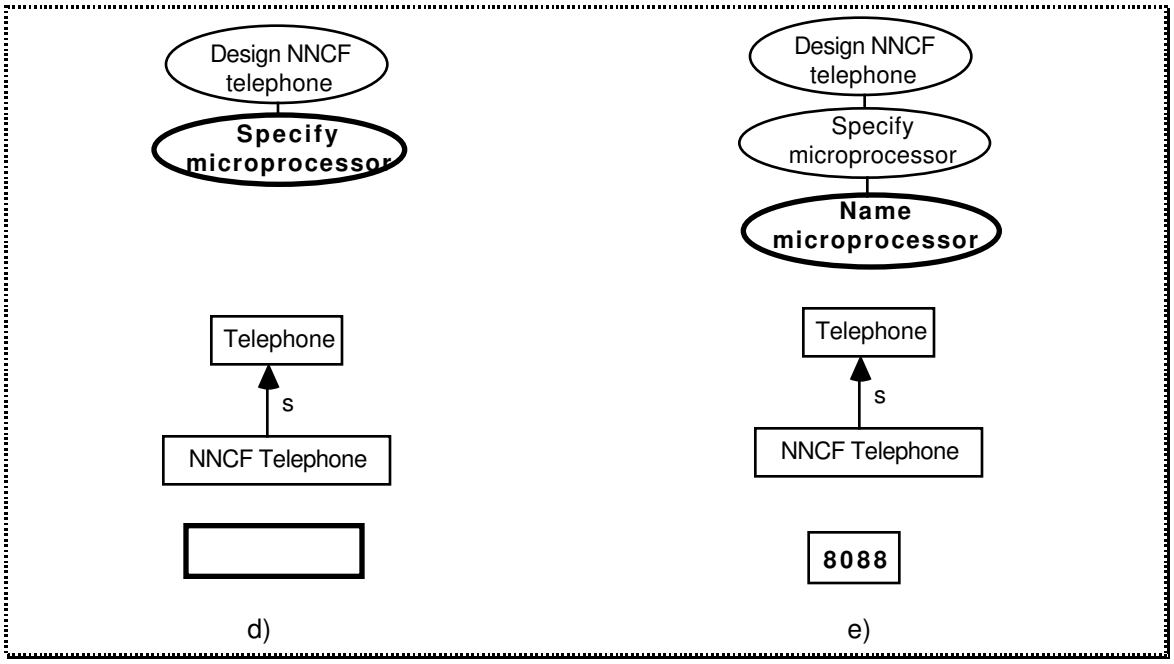

Figure 2 d & e

Step e: Naming the related concept

The user names the concept '8088'.

Suggestion e1: Concepts with the same name should be allowed.

In figure 2e, the user should not have been slowed down if another concept named '8088' had already existed.

Step f: Preparing for a property

The user has not finished specifying knowledge about the microprocessor, but his or her stream of consciousness is diverted into recording the fact that the telephone is to have a program. This is a sub-objective of the main objective.

Suggestion f1: The user should be able to specify links separately from their related concept.

In figure 2f, the user has specified a property occurrence rather than an independent concept to record his or her idea about 'has a program'.

The user should be able to work on a representation idea from either the relation or the independent concept perspective, whichever occurs to him or her.

Suggestion f2: Allow levels of formality when specifying properties.

In figure 2f, the new property has been specified very informally, with nothing more than a place-holder icon.

See also suggestion a5.

Suggestion f3. Provide assistance with tracking the objective hierarchy.

In figure 2f, the user has (for the first time) diverged from a strict stack of objectives, so objective tracking assistance might prove useful.

See section 3 for discussion.

Suggestion f4: Support should be provided for inverse properties.

In figure 2f, many properties of microprocessors and programs (e.g. memory size/requirement constraint, runs/runs-on) are mutual inverses.

The system should maintain such inverse relationships automatically. The main justification from a creativity point of view is prevention of distractive duplication of work.

Suggestion f5: Support should be provided for parallel properties (i.e. those whose refinement is similar in corresponding subtrees of an isa hierarchy).

In figure 2f, the 'purpose' of the microprocessor might be that of the program.

This is not a manifestation of inheritance, it is a form of concept-dependent constraint entered by the user. Often have we wasted time copying-and-pasting parallel properties. In general we do not follow Lenat's philosophy (Lenat, 1990) that such is a good way to enter knowledge: It reduces knowledge maintainability, and although it may cater to very short term creativity (recording stream of consciousness) in the medium term it impedes it by doubling work and increasing fear of inconsistency.

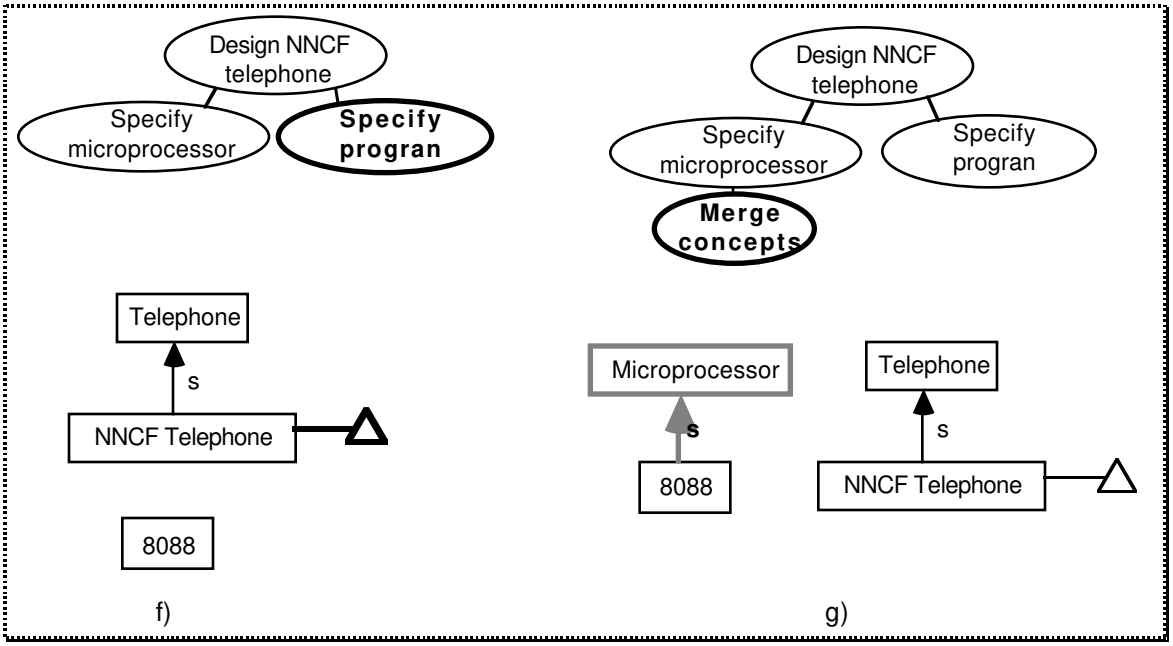

Figure 2 f & g. The triangle represents a 'property occurrence' (a conceptual relation)

Step g: Merging a concept

The user reverts to the 'Specify microprocessor' objective since the system has indicated that '8088' was already in this particular knowledge base. The operation performed is to merge concepts (unifying both them and references to them in the knowledge base; the result has the union of properties).

Suggestion g1: Provide sufficiently powerful editing primitives.

In figure 2g, the user performs a concept merge, one of many less frequent primitive knowledge-editing operations.

Aside from standard adds, deletes, renames etc., a number of less frequent primitive editing operations are needed to prevent side effects of operation composition. For example, it is necessary to ensure inheritance links are not temporarily broken (with the resulting loss of subtle knowledge (Lethbridge, 1991b)). An example of a less frequent editing primitive is 'concept reparenting'. All editing primitives need effective reference-making aids (suggestion c2)

Observaton g2: Help the user remember unfinished work.

In figure 2g, the user has not followed a logical sequence through the hierarchy, so she or he may later forget that about the objective of specifying the program.

Providing, on demand, a list of concepts visited but not recorded as 'satisfied' might be useful.

Step h: Adding a named property

As a further step towards specifying the microprocessor, the user indicates that NNCF telephones have electronic parts.

Suggestion h1: Organize properties hierarchically.

In figure 2h, the relation 'has electronic parts' is clearly a refinement of the relation 'has parts'. Whereas 'has parts' had previously inherited from some superconcept like 'device', the operation performed here makes 'has electronic parts' inherit as well (passively adding a superconcept 'device with electronic parts' to 'NNCF telephone' — not shown here)

We have found that having hierarchical properties (as opposed to the usual 1-level slot) substantially improves the power and compactness of the representation, and helps guide the user to the appropriate knowledge.

Suggestion h2: Provide a full ontology of properties.

In figure 2h, the user finds the 'electronic parts' relation to be already present, so creativity is speeded.

Not only do we emphasize that properties should be attached to the ontology of independent concepts, but the hierarchy of properties should be independently examined with reuse in mind.

Suggestion h3: Make property occurrences distinct from properties (i.e. ensure occurrences of conceptual relations and abstract conceptual relations are distinct concepts)

In figure 2b, while the property 'electronic parts' had previously existed, its occurrence in 'NNCF telephone' had not.

This is a another subtle representational distinction that we found essential to make (see suggestion a4). The set of things one can say about 'has parts' as a general relation is very different from what can be said about 'NNCF Telephones have electronic parts'.

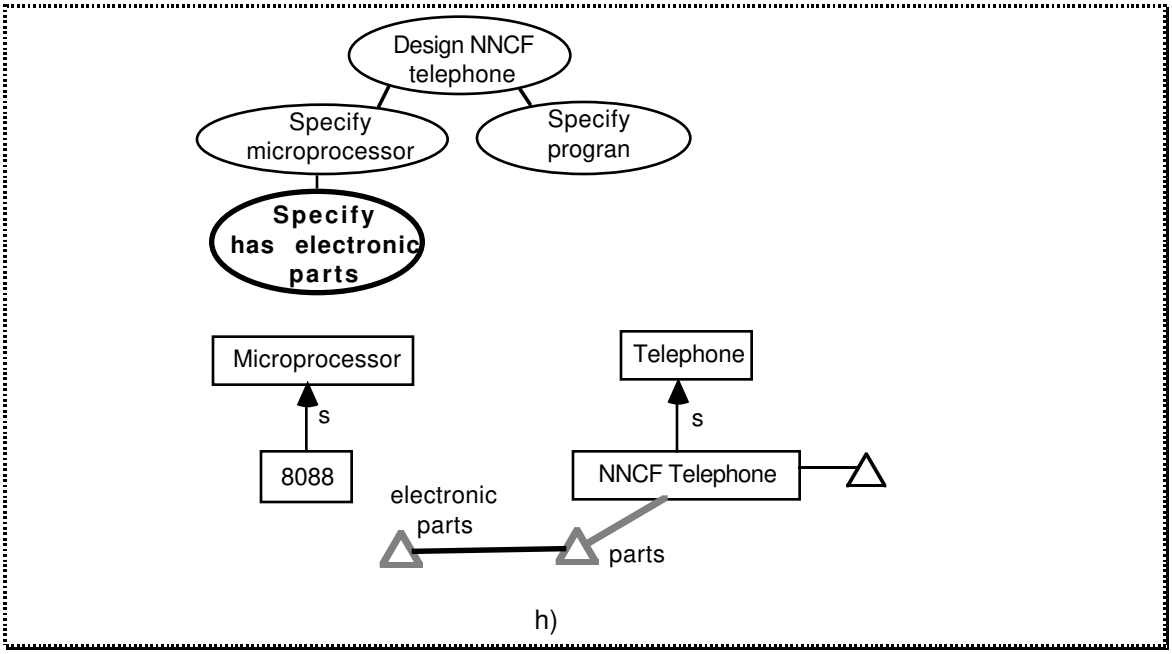

Figure 2 h. Properties such as 'has parts' are normally abbreviated (i.e. to 'parts')

Step i: Specifying an associated concept

The user indicates that the '[some] electronic parts of NNCF telephones are 8088s'. This many seem odd or incorrect; in fact it is only incomplete.

Suggestion i1: Keep associated concepts distinct from their superconcepts by default.

In figure 2i, the concepts '8088' and '8088 electronic part of NNCF telephone' should be distinct, because different properties apply to them. The latter, for example has the property 'contained in NNCF telephone'. This distinction is not shown visually in the figure.

This is another subtle distinction (like suggestion a4) that tends to be overlooked in most KR languages.

Suggestion i2: Feedback of suggestions made or problems detected should occur as soon as possible but not impede the user.

Figure 2i is the first step in the example where logical deduction performed by the KA system might detect problems. For example, '8088' might not be a valid refinement for the property 'electronic parts' as inherited.

Some kind of feedback should be provided immediately (a window listing problems, a colour change) but the user should have the option of responding to the feedback or not. As was pointed out in property 7, quite often the user may be less concerned with correctness and more with working out ideas.

Suggestion i3: Incompleteness should be tolerated.

In figure 2i, the user has not yet specified any other parts.

Unless asked, the system should not make deductions based on the assumption that the knowledge specified is complete, even for the purpose of passive feedback. To do so would be more distractive than worthwhile. (Inconsistency should also be toleraated, see suggestion s1).

Suggestion i4: Associated concepts should be specifiable in a dynamically interpretable way.

In figure 2i, we can assume the user typed 'an 8088' as the 'value' of 'electronic parts of NNCF Telephone' and that the system interpreted this as a request to create an associated subconcept of '8088'.

Another option the user may choose is typing some arbitrary natural language that the system cannot parse: In such a case, the user is recording knowledge 'informally' (see discussion in section 3). If the user had entered 'a widget', and the system had no concept of widgets, then the entry would be recorded as informal.

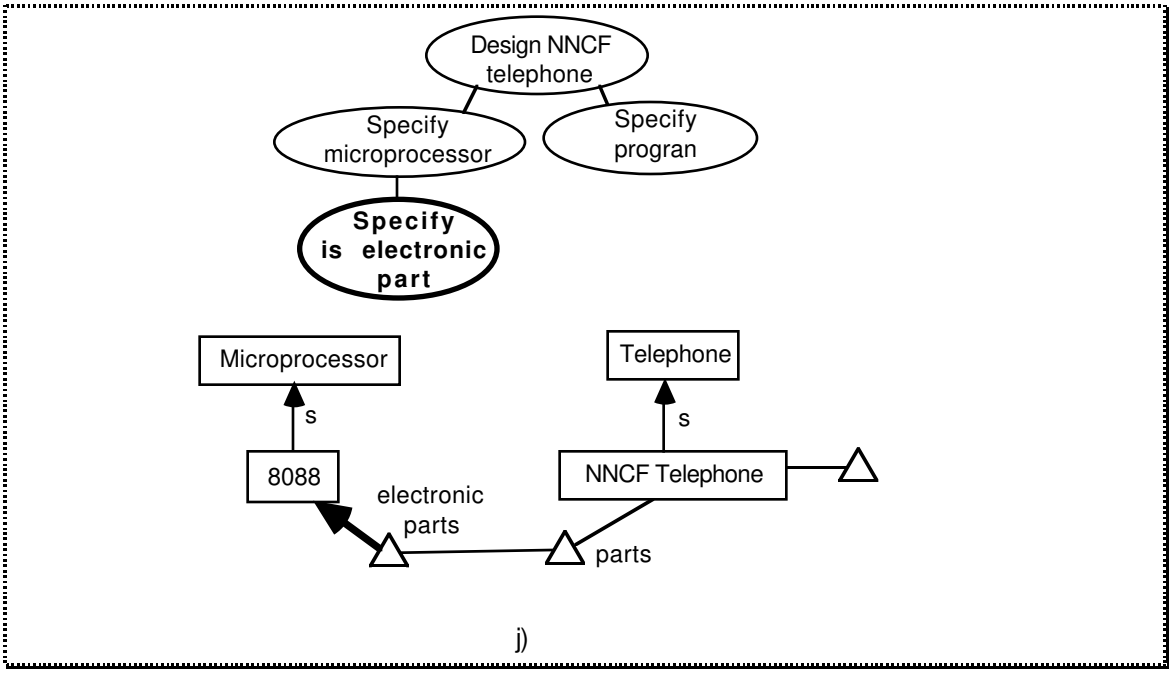

Figure 2 i

Subsequent steps

The nine steps described in this example should ideally take about a minute to enter (in a good system with an experienced user, we anticipate five seconds per step not counting thinking time). The user may continue to rapidly expand the knowledge base, eventually reaching a level of satisfaction that a sufficient level of detail has been recorded.

Suggestion s1: Commitment to the KB should be a separate step.

The user has to be confident that he or she is not 'damaging' existing knowledge, or entering 'unrepairable' knowledge. Three techniques can assist this: a simple undo capability, flagging changes as 'pending' until explicit commitment and providing a more complicated 'back-out' of committed changes. We believe that any kind of logical inconsistency should be tolerated up to the point of explicit committment (see suggestion i2). Commitment to the KB would imply that updated knowledge becomes visible in all views.

Suggestion s2: Powerful knowledge base restructuring techniques should be available so the user can correct errors and apply standards even after committing changes.

The justification for this is the same as in suggestion s1: Without confidence that problems can be fixed later, the user will slow down to get things right and may avoid entering concepts in which he or she lacks confidence.

Suggestion s3: Ensure that the system can be operated at a basic level with little or no training.

This may seem to be an elementary observation, when there are large numbers of primitive operations available (suggestion g1) it is important even to experts that basic operations can be performed without having to remember how. If a user has to look up how to perform an operation, his or her stream of consciousness will be interrupted.

5. CONCLUSIONS

Knowledge acquisition systems can be used in a largely creative mode where the actual knowledge represented in the machine may be of secondary importance to the inspirations created in the mind of the system's user. By analysing what contributes to creativity when performing KA, we have identified several key recommendations to enhance the creative potential of a KA system.

Our production KA system, CODE Version 2.4, follows some of the recommendations, and gave us many insights leading to the development of the other recommendations. At the time of writing we are implementing the latest version of CODE, version 4, which we anticipate will eventually fulfil all the recommendations.

A general recommendation for a creative KA system can be summarized as follows: "Facilitate the user's thinking - avoid inhibiting it". To be more precise, we present a list of recommendations below. Following each recommendation we indicate:

a) the suggestions from section 4 that the recommendation summarizes,

b) whether the recommendation applies to user interface (UI), ontology (ONT) or knowledge representation (KR), and

c) whether the recommendation is to help generate ideas (GEN), to help record the stream of consciousness (REC), or to organize the KA task (ORG).

List of recommendations

1. Give the user a sense of control over and immersion in the knowledge and system (suggestions a1, a2, b1, b2, f1, g1, i2) (UI, KR) (GEN, REC)
2. Don't force the user to follow particular thought paths (suggestions a3, a5, b1, e1, f2, i2, i3) (UI, KR) (REC)
3. Make allowances for subtleties of representation so that anything wanted can be represented in an intuitive way (suggestions a4, f4, f5, h1, h3, i1) (KR) (REC)
4. Provide the user with as much starting knowledge as possible (suggestions c1, h2) (ONT) (GEN)
5. Provide capabilities that prevent duplication of work (suggestions c1, c2, f4, f5, h1) (UI, ONT, KR) (REC)
6. Maximize the speed with which the user can make connections among concepts (suggestions c1, c2, h1) (UI, ONT, KR) (REC)
7. Make the tool adaptable to the user's preferred patterns of thought and work (suggestions a1, a2, d1, s3) (UI, KR) (GEN)
8. Prevent the formation of psychological barriers to the user's progress (suggestions s1, s2) (UI, KR) (REC)
9. Assist the user to track the task, forwards and backwards (suggestion f3) (UI, KR) (ORG)
10. Help the user remember unfinished work (suggestion g2) (UI, KR) (ORG)

ACKNOWLEDGEMENTS Support for this research is or has-been provided by Bell-Northern Research and the Natural Sciences and Engineering Research Council of Canada. The author thanks Douglas Skuce, Ingrid Meyer, Yves Beauvillé and Bruce Porter for their ideas and inspiration.

REFERENCES Gaines, B.R. (1989). Design Requirements for Knowledge Support Systems, *4th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 12-1.

Gaines, B.R. and Shaw, M.L.G. (1990). Cognitive and Logical Foundations of Knowledge Acquisition, *5th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 9-1.

Joseph, R.L. (1989) Graphical Knowledge Acquisition, *4th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 18-1.

Lenat, D. and Guha, R.V. (1990). *Building Large Knowledge-based Systems*, Addison-Wesley.

Lethbridge, T.C. (1991a). A Model for Informality in Knowledge Representation and Acquisition,*Workshop on Informal Computing*, May 29-31, Santa Cruz CA.

Lethbridge, T.C. (1991b). *CODE Version 4 Reference Manual*, in preparation.

Mark, W. and Schlossberg, J. (1990). Interactive Acquisition of Design Decisions, *5th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p 20-1.

Neale, I.M. (1989). First Generation Expert Systems: a Review of Knowledge Acquisition Methodologies, *The Knowledge Engineering Review*, pp. 105-145.

Reich, Y. (1990). Design Knowledge Acquisition: Task Analysis and a Partial Implementation, *5th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 26-1.

Roberts, R.M. (1989). *Serendipity: Accidental Discoveries in Science*. New York, Wiley.

Skuce, D., Wang, S., and Beauvillé, Y. (1989). A Generic Knowledge Acquisition Environment for Conceptual and Ontological Analysis, *Proc. 4th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 31-1.

Skuce, D., and Monarch, I. (1990). Ontological Issues in Knowledge Base Design: Some Problems and Suggestions, *Proc. 5th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 32-1.

Skuce, D. (1991a). A Wide Spectrum Knowledge Management System, Submitted to: *Knowledge Acquisition.*

Skuce, D. (1991b). Knowledge Management: A Tool and Its Use in Software Engineering Design, Submitted to: Eighth IEEE Conf. on AI for Applications.

Thost, M. (1989). Generating Facts From Opinions With Information Source Models, Proc. IJCAI, Detroit, pp. 531-536.

Warfield, J.N. (1976). *Societal Systems: Planning, Policy and Complexity*, New York, Wiley.

Wetter, T. and Woodward, B. (1990). Towards a theoretical framework for knowledge acquisition, *5th Knowledge Acquisition for Knowledge-Based Systems Workshop,* Banff (Canada), p. 35-1.