ASSESSING THE STRUCTURE OF KNOWLEDGE IN A PROCEDURAL DOMAIN

Michel C. Desmarais, Luc Giroux, Serge Larochelle, Serge Leclerc

Université de Montréal and
Canadian Workplace Automation Research Center

In most domains of knowledge, the process by which someone learns to become more expert is relatively constrained. People learn the basic concepts before they learn the more complex ones. They learn the simple, even though inefficient, methods for doing things before they learn the efficient but more complex methods.

These constraints in the process of learning reflect the structure of precedence, or of increasing complexity among knowledge items. This structure can be regarded in terms of implications: the knowledge of some complex concept implies the knowledge of some other, more simple concept. Similarly, the usage of an inefficient method implies that another, more efficient method is *not* known.

Obviously, the structure of implications among knowledge items is of extreme importance from a pedagogical point of view. It dictates the order in which to teach those items. It is also of great importance for knowledge assessment, since it is by such a structure that a tutor can infer what is or isn't known, and test just the right knowledge items such that the result will yield the most information about the individual's state of knowledge.

Structures that represent precedence, or increasing complexity of knowledge items, are well known in education (see Tatsuoka, 1985) and have, in fact, already been used in human-computer interaction to automatically infer user knowledge of a system (Zissos & Witten, 1986; Chin, 1986). Formal properties of implication structures have also been investigated by Doignon & Falmagne (1985). So far, the main approach has been to construct these structures intuitively, based on someone's experience of how the knowledge items are interrelated or on some evaluation of knowledge item complexity. Another approach has been to use text-books, course content, or documentation to verify the ordering in which knowledge items are introduced and to consider this as a basis for the knowledge structure. For instance, Pavel (1985) constructed a knowledge structure from the automatic analysis of UNIX™ on-line documentation. Our approach, similar to that of Pavel (1985), is based however on empirical data (see Desmarais & Pavel, 1987, for a previous study with the current approach). We will show how to construct such a structure from data on a number of individuals' knowledge of a domain. This approach has the advantage of not being biased by subjective judgment about the precedence or complexity of knowledge items. Moreover, given data on a number of individuals' knowledge states, the whole process of knowledge structure construction can be automatized.

This paper thus presents an empirical assessment of the methodology for constructing knowledge structures from data on individuals' knowledge state. We will demonstrate how such structures can be build and how efficient they are for inferring a single individual knowledge state from partial knowledge of that state.

THE KNOWLEDGE STRUCTURE

The knowledge structure is composed of knowledge items. No constraints is imposed on the definition of knowledge items. They can represent the comprehension of conceptual information as well as the ability to perform some task. All that matters is that the knowledge items define the knowledge domain in some meaningful and complete way.

The knowledge items are related to one another by two types of binary relations: the *logical implications* (or simply "implications") which states that A implies B, and the *negative implications*, which states that A implies *not* B. Those relations form the knowledge structure in question, which we will call the *implication network*.

Logical Implications

Probably the most important types of relation for knowledge assessment are that of precedence and of increased complexity among knowledge items. Those will be represented by **logical implications** in the implication network, in so far as they permit the inference of items that are known or not. In other words, precedence or increased complexity from A to B corresponds to a logical implication from B to A, to the extent that we infer that A is known if B is, and that B is not known if A isn't. This is not to say that precedence, increased complexity, and logical implication are all the same thing, but simply that they have the same properties and we will treat them as interchangeable here.

Figure 1 illustrates, by means of implication relations, the interdependencies that may exist amongst the abilities to solve problems in mathematics. Notice first that the structure forms a minimal partial order (it contains no cycles and no transitive relations). Notice also that the relations can be of different nature. In some cases, as in $2 \Rightarrow 1$, the implication stems from the fact that the type of problem in 1 is found in 2 as a sub-problem (i.e. we find a division problem in the algebra problem 2). Thus we have a clear precedence from 1 to 2. However, the implications from 5 to 4 and to 3 bear no such precedence, since problem 5 belongs to graph theory and does not require any knowledge of calculus nor matrices. It turns out that the implication is due to the extreme complexity of the solution for 5 (this problem was first proposed in 1852 and solved 125 years later after great efforts by many mathematicians) which suggests that if you solved that problem you must be knowledgeable enough to solve 3 and 4.

Negative Implications

The second type of relations is a *negative implication*. As an example, consider the fact that a student laboriously solves a complex system of linear equations algebraically, when the usage of matrices would have been much more efficient. From this example, we can conclude that the student does not master the matrix method, and that there is a negative implication from the algebraic to the matrix method of solving a complex linear equations system. Note that in a negative implication, one must discriminate between usage and knowledge, for it is the *usage* of some knowledge item that implies that another knowledge item is not known. Naturally, this type of relation is found in domains where knowledge items are manifest in some performance, that is, in *procedural domains*.

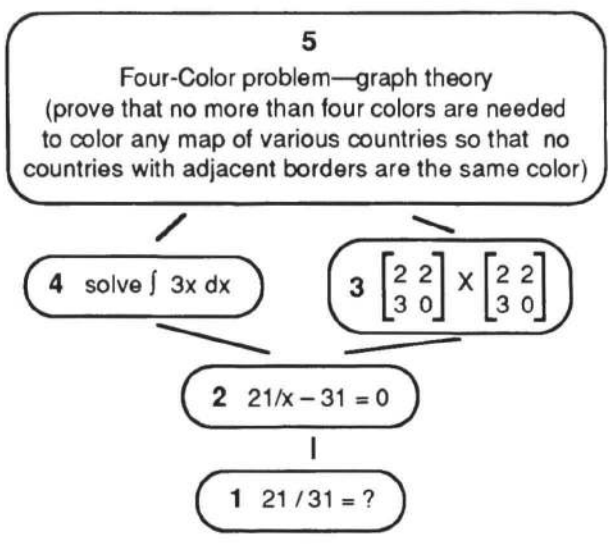

Figure 1. Implication network composed of logical implications among abilities to solve mathematical problems. Success for an item permits to infer that items implied are known, whereas failure permits to infer that items implying the failed item are not known.

Implication Network in a Procedural Domain

Although we have discussed the notion of implication structure in a general sense so far, we will now move the discussion in the context of *procedural knowledge domains*. Procedural domains can be characterized by *task structures* that define plans for completing specific tasks. Tasks also define **competences that can be represented by knowledge items**. Let us say a few words on the task domain in order to describe how the knowledge of that domain is represented and how the task structure relates to the implication network.

Text-editing

The domain in which we conducted our study is text-editing. This choice is determined by the fact that this study is part of a project for building an expert-system consultant for text editing (Desmarais, Larochelle, & Giroux, 1987). Thus, we are interested in the implication network both for pedagogical and for knowledge assessment purposes.

Text-editing is largely a procedural domain, in the sense that, in addition to concepts, the knowledge of this domain consists of actions, or procedures, for doing text-editing tasks. In fact, in the current study, **we will limit ourselves to the procedural**

dimension of text-editing, that is, to knowledge items that represent tasks, or *goals*, and to *primitive actions*, into which goals will ultimately be decomposed. (A primitive action is a task that cannot be further decomposed into sub-tasks and generally represents system functions, whereas a goal is a task that can be decomposed into sub-tasks, which can either be sub-goals if they are themselves further decomposed, or actions if they are not.) Thus, we will not have knowledge items that directly represent *concepts* like "buffers", or properties we can attach to characters (font, orientation, etc.) or to paragraphs (indentation, centering, etc), etc. Note, however, that such information could, indeed, be directly represented in the implication network and play an important role for knowledge inference.

Many tasks may be decomposed in a number of alternative ways which we will call *methods*. For instance, the task of replacing every occurrence of a word by another word can be achieved "by hand", replacing each occurrence one by one, or with some specialized system function designed especially for that task. The first method will thus be further decomposed into subgoals and actions for replacing text, whereas the second method will consists of a single primitive action (a system function).

Evidently, the second method will generally be much more efficient than the first one. In fact, given this information on method efficiency, we could presume that someone who uses the "find and replace each occurrence" method doesn't know the "search-replace" system function method. This leads us to a negative implication:

```
<find and replace each occurrence>
  ⇒¬ <search-replace function>
```

This relation is based on the postulate that if someone uses a sub-optimal method of doing some task, then the optimal method is not known. In a context where knowledge assessment is based solely on *known* knowledge items, as is the case of a coach who observes someone's performance, negative implications play a fundamental role for they are the only means of inferring what is *not* known from an implication network.

IMPLICATION NETWORK CONSTRUCTION

We described so far the nature of an implication network and its relation to the knowledge domain. In particular, we showed how some of the implications can be inferred from the task structure. We now turn to the problem of inferring *implications* and *negative implications* from **data on a number of individuals' knowledge state.**

Assessing Individuals' Knowledge State

The first step in the process of building an implication network is to gather information on a number of individuals' knowledge state. In order to do so, we elaborated a fairly exhaustive test to assess someone's knowledge of the editor WordPerfect™ (Leclerc, in preparation)[1]. The test contains 190 tasks. It is designed so that **the mastery of every major system function is tested individually by a task.**

There are three types of knowledge items that are associated with each task:

(1) **goal**: knowledge item that represents the task itself and which is considered mastered if the task is successfully completed, no matter what method is used;
(2) **methods**: knowledge item that corresponds to one or more primitive actions used *in the context of a goal.*
(3) **primitive action**: knowledge item that corresponds to a system function used; a primitive action is considered mastered if it is used successfully, no matter what the context is.

Hence, for each task, we find one knowledge item for the goal, one for each primitive action the task is decomposed into, and one for each method by which that task can be accomplished.

The distinction between a primitive action and a method enables us to discriminate between the usage of a system function in two different context. For instance, consider the two tasks of moving the cursor to the end of a word and to the end of the document. Although we will get a single knowledge item for the *primitive action* of moving the cursor one character to the right in both tasks, we will also get two different knowledge items for the *methods* which make use of that primitive action in each context. Indeed, moving the cursor to the right for going to the end of a word and for going to the end of the document legimately constitutes two different *competences.*

1 Leclerc, Serge (in preparation). Analyse de la structure de la connaissance des usagers d'un éditeur de texte, M.Sc. thesis in preparation, Département de psychologie, Université de Montréal.

Using the three types of knowledge items for the 190 tasks, we obtain the following distribution of knowledge items that compose the nodes of the implication network:

- 190 goals
- 195 primitive actions
- 286 methods

The total number of knowledge items is thus 671 (190 + 286 + 195).

The test was administered to 30 subjects. The performance varied from 56 to 146 successful tasks, which corresponds to 162 to 431 mastered knowledge items with an average of 307.

Compilation of Implications and Negative Implications

Once the data on individual knowledge states is gathered, the next step is to establish implications and negative implications among knowledge items. To determine if there is a relation between a pair of knowledge items, say A and B, we take the distribution of subjects along the following four situations:

(1) A and B are known
(2) A is known and B is unknown
(3) A is unknown and B is known
(4) A is unknown and B is unknown

then we state that there is an implication from A to B if we find people in situations 1, 3, and 4, but none in 2, as this situation would be impossible if, indeed, there were an implication. If there were a negative implication from A to B, then we would find people in all situations but 1. Establishing the implication network's relations then consists of analyzing the distribution of subjects over the four situations for each pair of knowledge items (the total number of pairs is 671 * 671 = 450,241).

Statistical parameters

If the world was black and white and there was an implication from A to B, we should never find a distribution like the following:

		B: *Known*	B: *Unknown*
A	*Known*	20	1
	Unknown	8	1

But because of noise, or simply because the implication does not reflect a clear precedence but something more like a strong surmise relation, we need some kind of statistical criterion to determine if there is or not a relation. We have used two statistical parameters to make this decision:

(1) the **minimal conditional probability** of B given A, paired with an **alpha error**. That is, given a measured conditional probability (20/21 = 0.95 in the distribution above) and a minimal conditional probability (we chose **$P(B|A)>0.85$**), we accept the implication from A to B if the lower bound of a [1 - alpha error] confidence interval around the measured conditional probability is greater than the minimal conditional probability (we chose an alpha error of **$p<.20$** which corresponds to an 80% confidence interval). In the distribution above, the lower bound of an 80% confidence interval around 0.95 is .86. Since it is greater than 0.85 we would accept the implication on this basis. However, even if we had a significant conditional probability, it is not necessarily different from the *initial* probability, meaning that the fact that A is known does not tell us anything more about B and that, consequently, there is no relation between A and B. For that reason, a second statistical parameter is needed.

(2) **the minimal probability of interaction, chi-square**: the chi-square value of the distribution determines if, indeed, there is a relation between two items. In the current study, we chose a chi-square corresponding **$p<0.15$**. For the distribution above, the chi-square is 0.41 and not significant at $p<0.15$. Thus we would reject the implication on the basis of this parameter.

Both parameters must be over the chosen criteria in order to set a relation, and both criteria apply to implication as well as negative implication relations (except that instead of 0.85 we will take 0.15 for the minimal conditional probability and look at the upper bound of the confidence interval).

Pruning and grouping

Having set the relations, it is often the case that we find transitive ($A \Rightarrow B$, $B \Rightarrow C$, $A \Rightarrow C$) and symmetric implications ($A \Rightarrow B$ and $B \Rightarrow A$). Because transitive relations are redundant in the knowledge inference process, they are removed from the structure. As for knowledge items involved in a symmetric implications, they are grouped together to

Table 1
Distribution of the number of relations, nodes, and knowledge items in the implication network

	relations	nodes	K.I.
implications	2804	145	355
non transitive	*393*		
-incoming		77	203
-outgoing		137	289
-incoming & outgoing		69	137
negative implications	3247	346	555
-outgoing		145	354
-incoming		201	201
-incoming & outgoing		0	0

form a single node in the structure. All incoming and outgoing relations of each node in the group are redirected to that node. The resulting structure corresponds to a minimal digraph.

Composition of the Implication Network Derived

Of the 671 knowledge items to start with, only 555 are involved in the implication network and, consequently, 116 knowledge items (671 – 555) are not related to any other items (35 of those 116 are not related because they are known by every subject and thus are rejected by the *minimal probability of interaction* criterion: indeed, it is impossible to establish any interaction with another item in this case). Moreover, because of grouping, the 555 knowledge items only form 346 nodes in the structure. There are 97 groups involving between 2 and 6 knowledge items, and one involving 47 knowledge items (grouping often occurs because a goal is always achieved by a single method, in which case the goal and the method will mutually imply each other); 248 nodes are composed of a single knowledge item. The distribution of implications and negative implications is given in table 1.

The structure is composed of a total of 2804 implications. 137 nodes have outgoing implications, 77 have incoming and 69 have both incoming and outgoing implications. Taking into account nodes that comprise multiple knowledge items, those figures become 289, 203, and 137 knowledge items respectively. Of the 2804 implications, only 1220 are non redundant (i.e. are not different paths between the same nodes) and only 393 are non transitive (i.e. form the minimal digraph). The distribution of the number of intermediate nodes in the transitive implications is compiled in table 2.

Table 2
Distribution of intermediate nodes in transitive implications

Intermediate nodes	Frequency
1	718
2	790
3	579
4	239
5	69
6	16

As for the negative implications, they are much more numerous with 3247 relations, but there is no transitivity with this kind of relation. All the 346 nodes in the implication network are involved in negative implications. 145 nodes have outgoing negative implications (355 knowledge items) and 201 have incoming negative implications (all of which are single knowledge item nodes).

VALIDATION OF THE IMPLICATION NETWORK

How valid are the implications derived? To answer this question, we conducted a simulation of knowledge inference for each of the 30 subjects with the following procedure: we sampled randomly a portion of the **successfully** completed tasks of a subject and fed this information to the knowledge inference module, which inferred the known and unknown knowledge items according to the implication and negative implication relations, and from the grouping of knowledge items. (Because of the context of our research, we wish to simulate the situation of a coach which is restricted to infer a pupil's knowledge from the *observation* of competence—a coach does not have direct information on what isn't known). We then compared the inferred known and unknown knowledge items with the actual knowledge state of the subject as measured directly from the test.

The results of the simulation are summarized in figure 2. A compilation of the correct ("hits") and incorrect ("false alarms") inferences is plotted for both known and unknown knowledge items, as a function of the proportion of the subjects' successful tasks that was sampled and fed to the knowledge inference module. The graphs show the values averaged for the 30 subjects. The white area indicates the size of the sample; the grey area indicates the information added by the inference process (hits); and, finally, the black area indicates the incorrect inferences (false alarms).

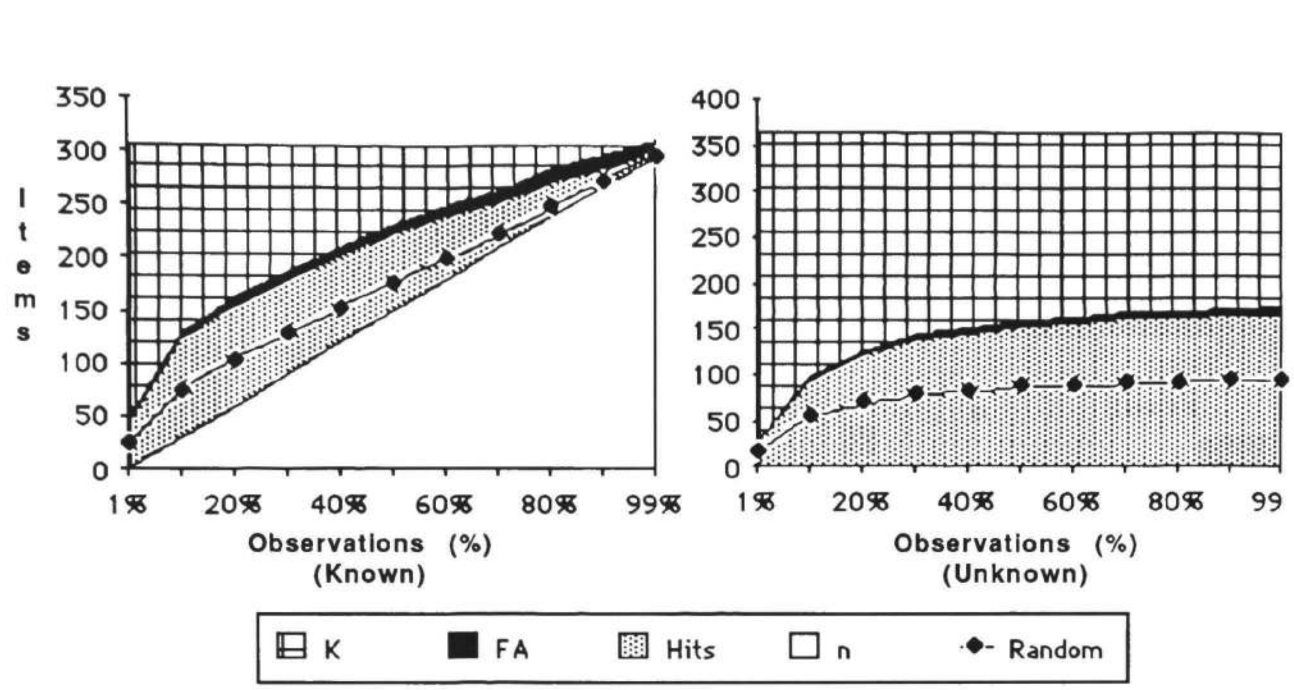

Figure 2 Analysis of the inference of known and unknown knowledge items as a function of the proportion of known items given to the knowledge inference module. The area labeled 'K' represents the actual known and unknown knowledge items. The dark area represents incorrect inferences (False Alarms) whereas the grey area represents the correct inferences, or 'hits'. 'n' is the number of knowledge items given. Hence the grey area represents the correctly inferred information for knowledge assessment. We included a "random inference" curve as a comparison.

The proportion of false alarms is relatively low in both graphs (below 10%), but increases for the known items to the point where all inferences are false alarms when the inference module is fed with *all* of the successful tasks, as can be expected. In fact, the greatest proportion of "added information" is between 10% and 40%, where the unknown inferred knowledge items are close to their maximum and where the proportion of false alarms over the hits is still relatively low for the known knowledge items.

Initial vs. added information

It must be noted that, for the purpose of knowledge assessment, the knowledge items inferred on the basis of the implication network constitute "added information" to what we ought to call the "initial information". That is, if we were to make a knowledge assessment from a sample of a subject's knowledge state, we would start with the initial information, namely, the initial probability of knowledge, and add to it the knowledge inferred from the implication network. For instance, in our case, we would start with at least 35 items known since their initial probability is 1 (they were known by every subjects).

The more severe the *minimal probability of interaction* is, the more the added information will differ from the initial information. In other words, the minimal probability of interaction assures us that the inferences made constitute information we didn't start with.

CONCLUSION

We have demonstrated that we can establish implications and negative implications among knowledge items, as well as grouping, by means of an empirical method and that we can characterize the structure constructed by some statistical parameters. We have also demonstrated that this structure can be used to infer a portion of known and unknown knowledge items from the observation of a portion of the known items. Moreover, we have all reasons to believe that with a sufficient number of individual knowledge states we can capture the structure of implication among knowledge items and that this structure constitutes a fundamental dimension of knowledge.

On the other hand, a number of questions remains unanswered. Although the simulation showed that the structure has a relevant power for knowledge inference, it is not clear how much more efficient it is compared to other, more simple schemes, like a linear structure where we simply order knowledge items as a function of the number of people who know them. If, in fact, the structure of the knowledge domain was linear, the performance of the current model (in terms of knowledge inference) would turn out to be similar to that of model based on a linear structure, as, indeed, the structure derived would itself be linear.

It would also be interesting to know precisely how many implications we missed with the small number of subjects we had and how many would be required to miss only a few (hopefully there might be a computable answer to this question from the statistical parameters of the network).

We are currently in the process of comparing the present knowledge inference scheme to simpler ones. A qualitative analysis of the structure obtained is also under way (Leclerc, in preparation). Finally, we have plans to repeat this experiment with other knowledge domains.

REFERENCES

CHIN, David N. (1986). User Modeling in UC, the UNIX Consultant, *Human Factors In Computing Systems, CHI'86 Conference Proceedings*, 24-28, Association for Computing Machinery, New York: NY.

DESMARAIS, Michel C., LAROCHELLE, Serge, & GIROUX, L. (1987). The Diagnosis of User Strategies, *Human-Computer Interaction—INTERACT'87*, H.-J. Bullinger and B. Shackel (Eds.), Elsevier Science Publishers B. V. (North Holland), 185-189.

DESMARAIS, Michel C. & PAVEL, Michael (1987). User knowledge assessment: An experiment with UNIX, *Human-Computer Interaction—INTERACT'87*, H.-J. Bullinger and B. Shackel (Eds.), Elsevier Science Publishers B. V. (North Holland), 151-156.

DOIGNON, Jean-Paul & FALMAGNE, Jean-Claude (1985). Spaces for the assessment of knowledge, *International Journal of Man-Machine Studies*, **23**, 175-196.

PAVEL, Michael (1985). UNIX knowledge assessment, Psychology Department, Stanford University.

ZISSOS, Adrian Y. & WITTEN, Ian H. (1985) User modelling for a computer coach: a case study, *International Journal of Man-Machine Studies*, **23**, 729-250.

TATSUOKA, Maurice M. (1986). Graph theory and its applications in educational research: A review and integration, *Review of Educational Research*, **56**, 291-329.