Chapter 3 A Computational Approach for Teaching a Graphical Learning Strategy

3.1 Introduction

The approach to this research has been influenced by the fact that graphical mapping is difficult to learn. To assist the learner in this endeavour, an approach for teaching graphical mapping amenable to computer implementation has been conceived. The proposed approach has three stages:

1. **Pre-Teaching**. The purpose of this stage is basically to introduce the task the learner is expected to accomplish as well as to gauge the learner's prior knowledge.

2. **Practice and Tutoring**. This book is mostly dedicated to this stage. Here, the learner practices mapping under the supervision of a computer tutor.

3. **Post-task Feedback**. This stage is concerned with analysing the learner's final map and then providing feedback and advice about how to improve it.

The lack of an expert mapper makes difficult the use of protocol analysis in the current research. Thus, this research has striven for presenting a training approach which would require a minimum of mapping expertise in its development. The approach sketched above is described in **Section 3.2**. A system which implements the second stage of this approach, named MapTutor, has been built. **Section 3.2.4** presents an overview of the components which make up this system. This section also shows how these components interact to provide individualised tutoring. The final part of this chapter is devoted to the knowledge component of MapTutor.

3.2 A Computational Learning Approach

As can be apprehended, the approach described here is similar in spirit to the modelling approaches described in **Section 2.6.2**. The main difference between that approach and the one described here is that the former is not intended for one-to-one tutoring, and therefore does not provide immediate feedback, whereas the approach presented here strives for individual, interactive tutoring wit h instantaneous feedback. Also, the approach discussed here requires a minimum of expertise from an instructional designer both in mapping and in the subject domain.

3.2.1 Pre-Teaching

In this stage, the learner is introduced to the graphical technique, especially the semantics utilised to represent the text in graphical form. The latter includes presenting the set of available links and providing a brief explanation of their meanings.

The learner is then told what she is expected to do (i.e., what the learning objectives are). Also important in this stage, is to ascertain the student 's background knowledge to see whether it is compatible to the minimum prior knowledge the program assumes she brings to the task. If the learner lacks adequate prior knowledge to achieve the learning goals, the program should supply her with it, or at the very least tell her that she does not have background knowledge sufficient for understanding the material. Finally, providing simple exercises by using a domain well known to the learner may be helpful here. This stage has been implemented by means of a tutorial program described in **Chapter 5**.

3.2.2 Practice and Tutoring

As its general strategy, MapTutor asks the learner to perform certain operation on the text (e.g., select a concept) or on her map (e.g., link two concepts on a partial graphical map), evaluates the result of that operation, and provides feedback about the efficacy of the operation. What follows is a brief account of how MapTutor carries out its duties.

The concepts of interest in the text are partitioned into two lists: (1) major-concepts, containing the learning objectives, i.e.,what must be learned from the text; and (2) minor-concepts, containing the auxiliary, explanatory concepts (other than those in the first list). Ideally, the learner should, at the end of instruction, know all of the concepts of interest in the text, but including on her map all of the concepts in the second list is not essential. Membership of these lists is determined via a relevance score rated in advance by the designer with the help of a domain expert, and is independent of the learner (see **Section 3.4**). The highest level of relevance is attached precisely to those concepts in the major-concepts list, whereas the lowest level corresponds to examples used to elaborate the text.

At the outset, MapTutor constructs another list, called known-concepts, containing those concepts the system assumes the learner knows beforehand. This assumption is based on the expected prior knowledge of the target students, who are expected to be mature students. For example, in the sample-text presented in **Table 3–1**, the student is assumed to understand the concept of tree, whereas the same does not hap pen with concept biotic factor. Although a particular learner (e.g., a Biology student) might also know the meaning of biotic factor, it is quite unlikely that anyone (in the target-student population) will not know the meaning of tree. Thus, no major concept, even if an assumed part of the learner's background knowledge, is added to this list at the beginning of instruction. In other words, the MapTutor requires that the learner demonstrates she really knows each major concept by representing them appropriately on her graphical map. As instruction progresses, MapTutor keeps list known-concepts by adding or retracting elements (i.e., concepts) according to its beliefs as to whether the learner has mastered those elements or not. The informal learnability criterion adopted by MapTutor can be stated as follows:

> The learner knows concept C if she has linked C with an adequate number of the most relevant concepts related to it.

This is definition is clearly vague, so that it needs to be made precise. A formal definition of this learnability criterion is presented in **Section 3.5**. By now, the important point is that MapTutor determines that the learner has successfully completed the mapping session when all major concepts are known by the learner according to the learnability criterion established by the program. More formally, the learner will have completed the tutorial session successfully when:

$$\boxed{\text{major-concepts} \subseteq \text{known-concepts}}$$

This mastering criterion will henceforth be known as successful task criterion.

What follows is the general, top-level strategy (algorithm) employed by MapTutor.

While there are concepts to be linked left in the major-concepts list, or the learner gives up, do:

1. Ask, if necessary, the learner to select two concepts from the text and link them together using one of the link names in the canonical set provided.

2. If the learner gets stuck (i.e., if she does not know which concepts to choose and link), while there are to-be-linked concepts, suggest two relevant concepts for her to consider linking. MapTutor determines that the learner has got stuck when either she has asked for help or her stretch of idle time (i.e., no-input period) exceeds a predetermined limit. The strategy used by MapTutor for suggesting to-be-linked concepts when the learner gets stuck is discussed later in **Chapter 4**.

3. The learner has made a link between two concepts (otherwise, she would not have gone out of the loop in **Step 2**). Determine whether this link is acceptable by verifying whet her the corresponding relationship between the two concepts can be mapped onto the canonical link the learner has just drawn[1].

   If the student's link is considered correct, it will not be further analysed. Otherwise, the program raises an exception, called triggering-condition, which is set to one of following values according to the status of the wrong link[2]:

   ◊ t-inexistent-link — when in effect there is no relationship between the given concepts.

   ◊ t-mismatching-link — when there is a relationship between the given concepts, but this relationship cannot be represented by the canonical link drawn by the learner.

   ◊ t-inverted-link-matching-name — is a special case of t-mismatching-link in which the learner may have inadvertently drawn the link in the opposite direction. When the learner systematically draws links in the direction contrary to that MapTutor expects her to draw, it could be the case that the learner prefers to interpret the link this manner (see **Section 3.7.3**). It is acknowledged that MapTutor does not address this problem appropriately. That is, all links drawn this way are currently treated by MapTutor as slips (but see **Section 7.2**).

   These triggering conditions are used to guide the diagnostic process which tries to identify the cause of a wrong link.

4. If **Step 3** has determined that the student's link is a correct one, just let the learner know in non-obtrusive manner so that she can follow her progress, and go to **Step 8**. MapTutor intervenes with parsimony, because it is believed that the more a method intervenes in the learning process, the more a learner may be distracted from the task at hand.

[1] From now on, the expression a relationship between two concepts will be taken with the meaning a relationship between two concepts as it occurs (either implicitly or explicitly) in the text. The expression a link between two concepts will stand for a canonical link drawn between two boxes representing concepts on a graphical map.

[2] Constants and procedure names used throughout this book should not be read as English words. Instead, they are symbols which, incidentally, have some resemblance with English names.

5. If **Step 3** arises a triggering condition other than t-INVERTED-LINK-MATCHING-NAME

   call the top-level diagnostic procedure to try to find out why the learner is wrong. If this procedure is successful at determining the nature of the wrong link, it returns SUCCESS along with the relevant nature; otherwise, it returns FAILURE. The diagnostic process is discussed thoroughly throughout **Chapter 4**, especially in **Section 4.3**.

6. If **Step 5** is successful, tell the learner she's wrong and provide instructional corrective feedback according to the nature of error pointed out by the diagnostic procedure (see **Section 4.9**).

7. If **Step 5** is not successful, do nothing, i.e., let the learner go on. In this case, MapTutor has run out of steam, so that little can be done, except perhaps to record the situation as accurately as possible in order to try to overcome this weakness in the future.

8. Update performance model (see **Section 4.3**).

9. Go back to **Step 1**.

Note that MapTutor does not impose any right sequence of linking concepts. Thus, the case in which the learner has not added a link where and when MapTutor would is postponed to the end of the session. This decision is based upon the fact that it is impossible to know whether the learner will add the link later during the session. Also, by proceeding so, the program somewhat lets the student learn at her own pace and style. Using the tutoring strategy described above, MapTutor analyses the student's input step-by-step, providing interactive correction while she is engaged in building a graphic map. This strategy consists essentially in monitoring opportunistically the learner and providing immediate feedback when the program determines it is necessary to do so.

3.2.3 Post-task Feedback

Right after the learner has finished her map, the program constructs an instructional plan to provide post-task correction and suggestions about how to improve future maps. This stage consists in analysing the learner's final map and then providing feedback and advice about how to improve it in a way much like the expert mapper in **Section 2.6.2** does. This idea has been implemented and is further explored in **Chapter 4**.

3.2.4 MapTutor's Architecture

MapTutor, the system which implements the second stage of the approach described above, can be divided into four main pieces:

A knowledge Base containing information about the to-be-mapped text, including all concepts and relationships of interest found in it. The knowledge base also contains descriptive information about each canonical link utilised by the system.

A performance model which not only keeps track of the learner's moves while constructing her map, but also and most importantly, tries to evaluate them.

A teaching unit responsible for elaborating feedback whenever the performance model determines the system should do so.

An interface containing the to-be-mapped text, a set of canonical links, a pane where the learner is expected to draw her map, and a pane which provides her with the necessary feedback.

MapTutor's general architecture is seen as a diagram showing the communication which occurs among the various parts of the system in form of message passing, as presented in **Figure 3–1**.

Notice that the **Control Centre** is in fact a virtual unit whose only function is to pass each message on to the unit which is actually able to handle it. The **Authoring Interface** at the top is through where the knowledge is introduced into the system, but it is not yet implemented, so that currently a programmer's interface has been used instead (see **Chapter 5**). MapTutor's knowledge base will be presented throughout the rest of this chapter.

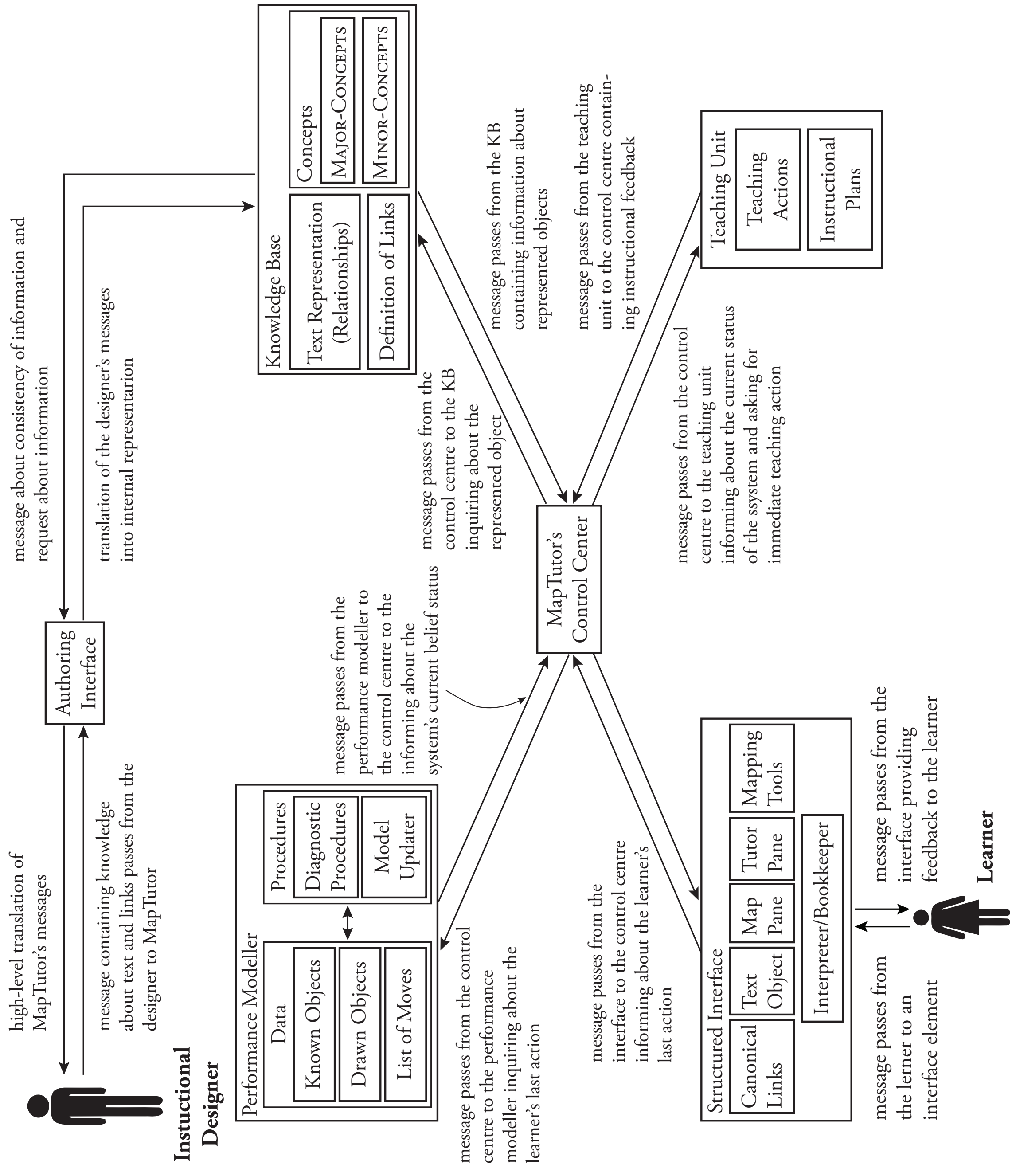

Figure 3–1: MapTutor's Architecture

3.3 Text Analysis

The first step towards representing a text into MapTutor's knowledge base is to analyse the text into smaller units until the propositional level (see, e.g., Kintsch, 1974) has been reached. The sample-text presently represented into MapTutor is a three-paragraph expository text on Ecology, extracted from Rowland's (1992) Biology textbook. The primary purpose (learning objective) of this text is to define and explain the concept of habitat. Accordingly, the text also contains descriptions of other auxiliary concepts (e.g., microhabitat, biotic and abiotic factors) as well as explanatory material (instances and examples) which help to both clarify and elaborate the main concept. The sample-text was chosen based on the following criteria:

- fair information density, which is supposed to be adequate for introductory graphical mapping;
- very little amount of background knowledge needed for understanding it, based on the subjects' expected prior knowledge;
- a fairly familiar subject in order to try to alleviate the burden of map construction (see McCagg & Dansereau, 1991).
- it was not created ad hoc to teach graphical mapping, because, as Holley & Dansereau (1984b) observe, the training material should not be so different from actual material as to cause lack of motivation to the learner.

Moreover, the text does not use either adjunct aids (e.g., questions, headings, titles, summaries, etc.) or text embellishment with selective function (e.g., highlights) which would provide hints to the learner as to where she should pay more attention. **Table 3–1** presents the unabridged version of the sample-text.

The sample-text was divided into 8 idea units. Each idea unit, which corresponds to one or more sentences in the text, was further divided into clauses. This latter subdivision included paraphrases and substitution of referents wherever necessary to make the clauses as clear as possible (cf. Sharples et al., 1994). For instance, the idea unit corresponding to the sentence:

An organism's habitat is the place where it lives, its address.

was divided into three clauses:

(a) a habitat is a place

(b) organisms live in habitats

(c) a habitat is like an address

The sample-text was analysed within this framework so as to make it easy for the learner to achieve the particular goals of grasping the learning objective and learning how to map the text. The content structure (main ideas) of the text is outlined in **Appendix A**.

An organism's habitat is the place where it leaves, its address. Our woodland is the habitat for a whole host of organism. Many organisms will only occupy a small part of the total habitat, for example, the snail in our woodland. This small part of the total habitat is called microhabitat. Each habitat will have certain distinct features which affect the organisms living in it. On the other hand, there are psycho-chemical or abiotic factors: climate, soil, type of water (marine, fresh, running, still) and so on. On the other hand there are biotic factors, which are determined by the organisms which share the habitat. For example, organisms which eat each other compete with each other for food or provide shelter. Biotic and abiotic factors are not independent of each other. For example, the trees in a woodland affect the humidity, temperature and amount of sunlight there. So, trees, a biotic factor, influence the psycho-chemical features of the habitat which, in turn, will affect the other organisms living in the woodland. Understanding the complexity of interaction between organisms and their habitat is one of the challenges faced by ecologists. (Rowland, 1992, pp. 667–8)

TABLE 3–1: SAMPLE-TEXT ON HABITAT

3.4 Analysis of Concepts

Subsequent to the text analysis described above, the concepts identified in the sample-text were classed into major concepts — i.e., the most important ones to the learning objectives — and minor concepts — i.e., concepts used as auxiliary, explanatory material. Each concept was then assigned a rank on a five-point scale which indicates its relative importance for the full understanding of the text. These concepts and their respective rank scores are summarised in **Table 3–2** and **Table 3–3**[3].

Also included in the following in **Table 3–2** and **Table 3–3** is whether each concept will have in its representation a pre-requisite or a definition slot. A concept is prerequisite to another if the former is necessarily part of the definition of the latter. In the text at hand, only concept MICROHABITAT seems to satisfy this criterion. Typically, minor concepts will not need to have a definition because they

[3] I was helped in this concept ranking by Dr Libby John at BIOS, University of Sussex. She was also responsible for pointing out some ambiguities in the sample-text and for clarifying some pieces of it. I am very grateful for her assistance.

are used as explanatory devices (e.g., in metaphors and analogies) and therefore should be assumed to be well known by the learner target.

In fact, using familiar concepts to explain another unfamiliar one constitutes a good educational practice (see, e.g., Good & Brophy, 1990; Howard, 1987).

Concept	Rank	Prerequisite	Definition
HABITAT	5	none	yes
WOODLAND	4	none	n/a
ORGANISM	4	none	n/a
PHYSICO-CHEMICAL FACTOR	4	none	yes
ABIOTIC FACTOR	4	none	yes
BIOTIC FACTOR	4	none	yes

TABLE 3–2: MAJOR CONCEPTS IN THE SAMPLE TEXT

Concept	Rank	Prerequisite	Definition
MICROHABITAT	3	HABITAT	yes
PLACE	2	none	n/a
ADDRESS	1	none	n/a
SNAIL	2	none	n/a
CLIMATE	3	none	n/a
SOIL	2	none	n/a
TYPE OF WATER	3	none	n/a
MARINE	1	none	n/a
FRESH	1	none	n/a
RUNNING	1	none	n/a
STILL	1	none	n/a
ORGANISMS EATING EACH OTHER	2	none	n/a
ORGANISMS COMPETING FOR FOOD	2	none	n/a
ORGANISMS PROVIDING SHELTER	2	none	n/a
TREE	2	none	n/a
HUMIDITY	2	none	n/a
TEMPERATURE	2	none	n/a
AMOUNT OF SUNLIGHT	2	none	n/a

TABLE 3–3: MINOR CONCEPTS IN THE SAMPLE TEXT

3.5 Concept Learnability

MapTutor's measures of the student's performance consist of two lists of beliefs about the learner:

1. believed-concepts is a list containing those concepts which MapTutor believes the learner somewhat knows how to map. In other words, believed-concepts contains all of the concepts connected thus far to at least one other concept on the student's map.

2. known-concepts is a sub-list of believed-concepts containing those concepts MapTutor believes the learner knows how to use correctly. Membership of this list of elaborated as follows.

Associated with each concept in the list believed-concepts, there is a belief-degree which quantifies the extent to which MapTutor believes the learner knows how to use (i.e., map) that concept correctly. The belief-degree of a given concept *C* is elaborated as follows. Suppose the learner connects *C* to a number of concepts. The extent to which the learner knows how to map *C* is given by the extent to which she knows how to link correctly *C* to each of the important concepts related to it. Then, assuming that we have an assessment function which, given a link, returns the degree of correctness of the link, we can say that. the contribution of a link between *C* and c_i to the knowing of *C* is given by:

(assessment of link between C and c_i) × (relevance of c_i)

Remember from **Section 3.4** that the relevance of a given concept is given by its pre defined rank. Thus, the extent to which the learner knows how to map *C* will be given by the sum over all links between *C* and c_i's the learner has made, which can be written as:

$$\sum_{1 \le i \le n} ass(C, c_i) \times rank(c_i) \qquad \textbf{(3.1)}$$

where *n* is the number of concepts in the knowledge base representing the text at hand, and $ass(C, c_i)$ is a function of assessment of the link made between concepts *C* and c_i. Notice that the only constraint in **Equation 3.1** is that any c_i be different from *C*. This means that the learner is free to link *C* to any concept in the knowledge base, except to *C* itself.

Now, we will know with absolute certainty that the learner knows how to map *C* when she connects correctly *C* to all concepts which are actually related to

C[4]. If the value returned by our assessment function when the link is correct is k, this certainty corresponds to:

$$\sum k \times rank(c_i) = k \times \sum rank(c_i) \tag{3.2}$$

with c_i ranging over all concept which are actually related to C.

Equation 3.2 is defined as the link-value (LV) of concept C, and to make it simpler, constant k is made equal to 1[5]. Therefore, LV can be written as:

$$LV(C) = \sum rank(c_i) \tag{3.3}$$

Now, if the assessment function returns a greater value for a correct link than for a wrong link (as seems to be intuitive), LV will correspond precisely to the maximum value that **Equation 3.1** could get. Another important characteristic of LV is that, given a concept C, its LV depends only on the ranks of the concepts related to it and the value returned by the assessment function for correct links, which are both known at the outset.

We can finally· define the belief-degree of a concept as the proportion of the score the learner obtained by **Equation 3.1** to the maximum score she could possibly get, which is given by **Equation 3.3**. Defined in this way, belief-degree has the important property that,

> By establishing (experimentally or otherwise) a knowledge threshold we can allow the learner to prove to know how to map a given concept without necessarily requiring her to link this concept to all other concepts related to it. This also means that a large number of correct maps can be allowed by the system.

The ideas above are further formalised as follows. Belief-Degree (henceforth BD) is calculated for each concept c_i in list BELIEVED-CONCEPTS as:

[4] Actually related to C means those concepts whose relationships to C have been input into the system. That is, those relationships involving C the system knows the existence of.

[5] This can safely be done because absolute values of the evaluation function do not make sense. That is, what matters is that such an evaluation function should return values relative to the degree of correctness of the given link. For instance, by just knowing that an assessment value is, say, 1, one will not be able to tell whether the link was considered correct, incorrect, or some fuzzy value in between. Moreover, in the discussion which follows, it is not necessary that a link be considered right or wrong. It could also be represented by a continuous degree of correctness, ranging from, say, 0 — representing a wrong link — to 1 — representing a correct link. A tree-state scale has been used by MapTutor, but a larger discrete scale or ·even a continuous one could also be used without making many changes in the program.

$$BD(c_i) = \frac{\sum_{1 \le j \le n, j \ne i} ass(c_i, c_j) \times rank(c_j)}{LV(c_i)} \quad \textbf{(3.4)}$$

where:

- n is the number of concepts represented into the program's knowledge base;
- c_j ranges over all of the concepts the learner has linked to c_i;
- $ass(c_i, c_j)$ is a function of assessment of the link made between concepts c_i and c_j, which is defined as:

$$ass(c_i, c_j) = \begin{cases} 1, & \text{if } c_i \text{ and } c_j \text{ have been correctly linked} \\ 0, & \text{if } c_i \text{ and } c_j \text{ have been incorrectly linked, but } c_i \text{ and } c_j \\ & \text{are in fact related do each other} \\ -1, & \text{if } c_i \text{ and } c_j \text{ have been incorrectly linked, and there is no} \\ & \text{relationship between } c_i \text{ and } c_j \end{cases}$$

These values quantify how good or bad a given link is. Thus, they are to be interpreted as relative values; they are senseless when taken in isolation.

- $rank(c_n)$ is the rank of concept c_n described in **Section 3.4**;
- LV(c_i), the link-value of concept c_i, is defined as follows:

$$LV(c_i) = \sum_{m \ne i} rank(c_m)$$

with c_m ranging over all concepts actually related to c_i[6].

Given the definitions above, a major concept is a member of list KNOWN-CONCEPTS if its BD is equal to or greater than a given threshold (Knowing-Threshold — henceforth KT) which is an arbitrarily pre-set parameter of the program. Note that the concept of membership of list KNOWN-CONCEPTS is dynamic, so that if the BD of a major concept which is a member of this list falls below the threshold, it is retracted from the list. Currently, KT is set at *0.6*, which means that the learner will be assumed to know how to map a given concept when her score (BD) obtained for the links she made connecting this concept is 60% of its link-value (LV).

[6] Notice that, as LV depends only on the pre-set ranks, it is calculated for each concept just once, when the given domain is represented into the program. That is, LV is a knowledge representation parameter. By contrast, BD is a performance parameter, and therefore, it is calculated incrementally during a tutoring session.

An Example of Belief-Degree Calculation. At this point, an example illustrating the ideas above may be helpful. Suppose, for instance, that a learner has just made a link between concepts c_{10} and c_8, as shown in **Figure 3–2**.

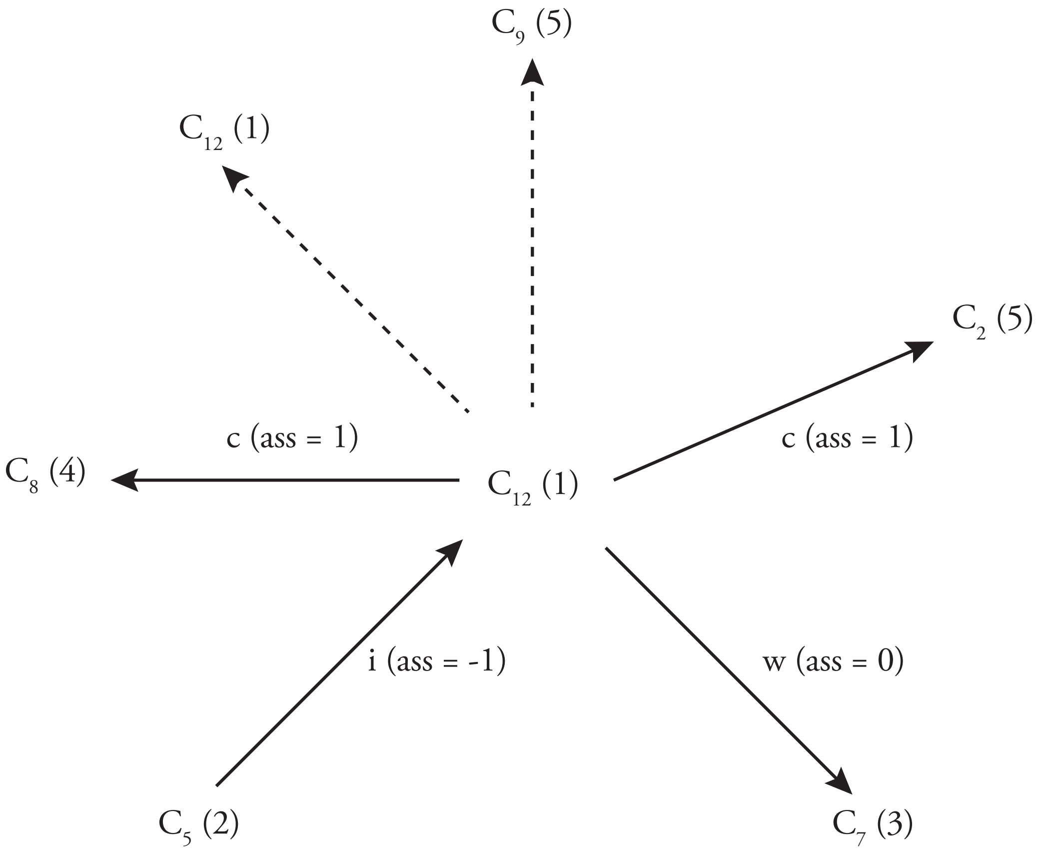

FIGURE 3–2: EXAMPLE OF BELIEF-DEGREE CALCULATION

Figure 3–2 should be interpreted as follows. c_n's are arbitrary concepts; the numbers enclosed between parentheses are the ranks of the respective concepts; a full line represents a link actually made by the learner; *c* means a correct link; *i* means an inexistent relationship; *w* means a wrong link, where there is in fact a relationship; and $ass = n$ means that the assessment function assumes value n. Broken lines represent existing relationships which the learner has not considered yet.

In this example, we have:

$$\begin{aligned}\textsc{lv}(c_{10}) &= rank(c_9) + rank(c_2) + rank(c_7) + rank(c_5) + rank(c_8) + rank(c_{12}) \\ &= 5 + 5 + 3 + 2 + 4 + 1 \\ &= 20\end{aligned}$$

$$\begin{aligned}\textsc{bd}(c_{10}) &= [ass(c_{10}, c_2) \times rank(c_2) + ass(c_{10}, c_7) \times rank(c_7) \\ &\quad + ass(c_{10}, c_5) \times rank(c_5) + ass(c_{10}, c_8) \times rank(c_8)]/\textsc{lv}(c_{10}) \\ &= [1 \times 5 + 0 \times 3 + (-1) \times 2 + 1 \times 4)/20 \\ &= 0.35\end{aligned}$$

Since BD$(c_{10}) = 0.35 <$ KT, c_{10} would not be a member of list KNOWN-CONCEPTS. It is interesting to observe that, if in a situation subsequent to that described in this example, the learner had correctly linked c_{10} to c_{12} ($rank = 1$), then BD$(c_{10}) = 0.40$, and c_{10} would yet not be in KNOWN-CONCEPTS. However, if she had instead correctly linked c_{10} to c_9 ($rank = 5$), then BD$(c_{10}) = 0.60$, and now c_{10} would then be included in KNOWN-CONCEPTS. This makes sense because, as c_9 has $rank = 5$, it is bound to be part of the gist of the text, whereas c_{12}, with $rank = 1$, is probably merely a detail or example used to elaborate or explain the main point of the text. Therefore, it is much more likely that the learner will know c_{10}, given that she knows the relationship between it and c_9 — an important concept — than when she knows the relationship between c_{10} and c_{12} — a not-so-important concept.

Equivalence of Concepts: A Special Case. A special case, which the learnability criterion defined above must take into consideration, occurs when there are two concepts which are equivalent to each other (e.g., one concept is the definition of the other, or they are represented by synonymous word-concepts). It does not seem fair to require the learner to connect (explicitly) all concepts related to such a pair of concepts to both concepts in the map. For example, in the sample-text, ABIOTIC FACTOR and PHYSICO-CHEMICAL FACTOR represent the same entity. Hence, if, for instance, the learner connects ABIOTIC FACTOR to TEMPERATURE, she may still connect this latter concept to PHYSICO-CHEMICAL FACTOR, but she will not be required to do so. This appears to be reasonable because the second link does not add very much to the learner's knowledge about those concepts. But, if a learner decides to link the related concepts to only one concept of the equivalence pair (say, ABIOTIC FACTOR in our example), how will the other concept (PHYSICO-CHEMICAL FACTOR in our example) have its BD increased so that this concept will be considered as a known concept? The solution for this particular problem takes three steps:

1. Only one concept[7] of a pair of equivalent concepts is taken into consideration when calculating the link value of other concepts connected to the pair.
2. When the learner links a concept c_1 to two other concepts, c_2 and c_3, and these latter concepts are equivalent, only the first link will be taken into consideration for the BD calculation of c_1[8].

[7] Of course, since the concepts are equivalent, they will have the same ranks and link-values. Thus, the choice of which concept will be chosen is completely irrelevant.

[8] This solution is provisional and will be consistent only when both links are either right or wrong (i.e., when the links themselves are consistent). But, what if the learner's first link is correct

3. The learnability criterion will be generalised so that it is applicable to this special situation. The new learnability criterion is then defined as follows.

The learner will know concept c_1 when she knows c1 by applying the old criterion defined above, or she knows c_2, c_2 is equivalent to c_1, and she has proven to know this fact by connecting c_1 and c_2 by using the appropriate link (e.g., link EQUIVALENT).

3.6 MapTutor Links

The links currently used by MapTutor are among those most commonly used in graphical strategies research (see **Section 2.4**). These links are presented in **Table 3–4** and their meanings have already been extensively discussed earlier.

MapTutor does not purport to present a general set of canonical links which could be used with all kinds of texts. Nevertheless, even without encompassing all possible relationships which could occur in an arbitrary expository text, the links chosen for the sample-set used in this research appear to be sufficient for encoding example texts, and thus are quite suitable for experimental purposes.

Link
PART OF
IS A
LEADS TO
EQUIVALENT
HAS PROPERTY

TABLE 3–4: SAMPLE-SET OF LINKS EMPLOYED BY MAPTUTOR

The choice of the sample-set of links for MapTutor does have empirical support (see **Section 2.4**; see also Holley & Dansereau, 1984b). It should also be emphasised that the proposed system allows for the use of different sets of links as well as different texts. MapTutor achieves this by having its diagnostic and teaching modules independent from any particular set of links. That means that

and the second one is wrong (or vice-versa). Should we always consider the correct link or the wrong one in the BD calculation of the concept? Should we always consider the first link (be it correct or not)? Why not consider always the second link? Perhaps the learner's misunderstanding is that she does not know that the two equivalent concepts are actually equivalent. No easy solution has been found to this question, but in any case, feedback will be provided for both links. This problem stem from the fact that this research has not concentrated on reasoning over links and how they fit together in an overall map.

MapTutor's design allows, without any further modifications, the set of links to be easily changed if they eventually prove to be inadequate (see **Chapter 5**).

3.7 Knowledge Representation

3.7.1 Representation of Concepts

MapTutor keeps a list containing definitions and some additional information about the concepts found in the to-be-mapped text. These concepts are represented into MapTutor's knowledge base according to the concept **Prototype** presented in **Table 3–5**.

According to the concept **Prototype**, Name is the word-concept (Quillian, 1968) which represents the given concept; **Definition** is the concept's definition, if applicable (see above); **Rank** is the concept's rank described above; **Prerequisite** is the concept, if any, that should be learned before the one being described; Link value lv is the concept's link value described earlier in **Section 3.5**. Slot **Equivalent to** contains a concept which is equivalent to the one being described. This slot is necessary for implementing that special case of concept learnability discussed in **Section 3.5**. The last two slots — **Default location** and **Where in text** — are not conceptually important for this book. Default location corresponds to where the concept will initially be drawn on the map pane when the learner selects a concept for the first time in the text, and *Where in text* is a list containing the positions in the text where the concept can be found; the program uses such lists to decide whether the learner has select or not a concept after clicking the mouse on the text. Therefore, these are more specific implementation details and interface features employed by the program described here. As an example, concept microhabitat has the representation shown in **Table 3–6**.

The representation of all concepts found in the sample-text used by MapTutor can be found in **Appendix B**.

Slot	Definition
Name	the concept's name
Definition	the concept's definition (if any, see **Section 4.3** and **Section 4.9**)
Rank	the concept's rank
Prerequisite	the concept (if any) that should be learned before the one being described
Link value	the concept's link value
Equivalent to	the concept's equivalent

Slot	Definition
Default location	where the concept will initially be drawn on the map pane
Where in text	list containing the positions in the text where the concept can be found

Table 3–5: Concept Prototype

Slot	Value
Name	MICROHABITAT
Definition	Microhabitat is a very small, specialised habitat, such as a clump of grass or an open space between rocks. [Adapted from The American Heritage Dictionary of the English Language, Third Edition, Copyright ©1992 by Houghton Mifflin Company, Microsoft-BookShelf 1994.]
Rank	4
Prerequisite	HABITAT
Link value	10
Equivalent to	none
Default location	(11, 163) [Top, left of the concept's enclosing rectangle in the map pane's coordinate system.]
Where in text	[(277, 289) [This concept ·appears only once in the given text.]

Table 3–6: Representation of Concept microhabitat

3.7.2 Representation of Links

MapTutor keeps a list containing definitions and other useful information about each link in the set of canonical links provided by the program. Each link in MapTutor's knowledge base is represented according to the LINK **Prototype** shown in **Table 3–7**.

The first three slots — **Full name**, **Short name** and **Abbreviation** — are all used to identify the given link and their distinction is not conceptually important. Full name is used by the program when it provides feedback; Short name is used by the program to label the buttons representing the links in the interface; and *Abbreviation* is the label used in the links drawn on the map[9]. The distinction

[9] **Short name** and **Abbreviation** are used in order to avoid cluttering the interface and the student's map, respectively.

made between **Definition** and **Meaning** in **Table 3–7** is that a link's definition stands for itself, whereas its meaning represents how the link should be interpreted once it connects two (boxes representing) concepts on the graphical map. Slot **Inherently ambiguous** provides some clue to the program by anticipating a potential source of difficulty the learner could face in interpreting the meaning of the link being described. For example, although the semantics of link IS A seems to be clearly well defined, the same does not always happen with link LEADS TO.

Slot Coexists with contains precisely the link (if any) which can be used together with the one being described without being either redundant or duplicated. If there are two links which can coexist with each other, this coexistence must mean links in opposite directions, otherwise the link system will at best be redundant (if not ambiguous). In the current canonical link system employed by MapTutor, only link leads to can coexist with itself[10]. For example, biotic and abiotic factors affect each other can be represented by the following canonical propositions: *biotic factors leads to abiotic factors* and *abiotic factors leads to biotic factors*. Finally, as seen in **Chapter 2**, canonical links should not be taken literally by the word used to label them. Instead, they have meanings broader than the corresponding English words. Slot **Keywords** represents a list containing the keywords depicting actual relationships which the link is supposed to cover. Link part of, for example, has the representation shown in **Table 3–8**.

Slot	Definition
Full name	the link's full name
Short name	the link's short name; used in the link's pane
Abbreviation	the link's abbreviation
Definition	the link's definition
Meaning	the link's meaning
Inherently ambiguous	whether or not the link's is inherently ambiguous
Coexists with	the link which can be used together with this one
Keywords	the set of keywords which indicates the link's usage

TABLE 3–7: LINK PROTOTYPE

[10] Strictly speaking, IS A can coexist with itself. But, if we have both c_1 IS A c_2 and c_2 IS A c_1 simultaneously, then the most appropriate link will be EQUIVALENT. Teaching the most appropriate link is what MAPTUTOR is all about.

Slot	Value
Full name	IS PART OF
Short name	PART OF
Abbreviation	PART
Definition	A concept is IS PART OF another concept when the first is a piece, portion or integral constituent of the second (the whole)
Meaning	The concept represented by the origin node IS PART OF the concept represented by the terminal node.
Inherently ambiguous	no
Coexists with	none
Keywords	[PART, SEGMENT, PORTION]

TABLE 3–8: REPRESENTATION OF LINK PART OF

The representation of all links used by MAPTUTOR during its experimental stage can be found in **Appendix C**.

3.7.3 Representation of Relationships

MAPTUTOR keeps information not only about the relationships which hold among concepts in the text, but also about how to map each relationship onto one of the canonical links provided. Each relationship of interest found in the to-be-mapped text is represented into MAPTUTOR's knowledge base according to the **RELATIONSHIP Prototype** depicted in **Table 3–9**.

Slot	Definition
Link	the canonical link the relationship should be mapped onto
Original concept	one of the concepts connected by the relation
Terminal concept	the other concept connected by the relation
Is explicit	whether or not the relation is explicit in the text
Ambiguous	whether or not the relation is ambiguous in the text
Reasoning needed	estimated amount of reasoning needed to uncover the actual relationship in the text
Justifications	the justification for the choice of the expected link
Where in text	where the relation can be found in the text

TABLE 3–9: RELATIONSHIP PROTOTYPE

Link stands for an appropriate canonical link, from the set of links currently provided, onto which the relationship can be mapped. Notice that each relationship is represented as many times as the number of links it can be mapped onto. Origin concept and Terminal concept are the (boxes representing) concepts the link drawn on the map originate and terminate at, respectively. This distinction is made on the map by means of arrows which indicate the direction of the link being depicted. The direction of a link specifies how the link should be read. This choice is· arbitrary, but once made, one must stick to it in order to avoid confusion. For example, as seen in **Section 2.3.1**, Holley & Dansereau (1984b) prefer looking at link PART OF from an owner-concept to an owned-concept (see **Figure 2–1**), so that the link has (say) the intended meaning has part. On the other hand, I have found it more natural to represent the same link in the opposite direction, with the intended meaning of (say) is part of. Future expansion of the system described here should allow the learner choose the direction that best suits her interpretation and style of looking at relationships. Slot Is explicit indicates whether the relation is explicit in the text, or whether it is implicit and requires some inferences in order to uncover it. Slot Is ambiguous tells the program whether or not the relation appears ambiguous in the text, and therefore may be hard for the learner to interpret. Because the program is designed to use real-world texts, it is important to have fed into the system information about ambiguity, which occurs very frequently in many pieces of text. For example, a close look at the following excerpt taken from the sample-text[11],

> Many organisms will only occupy a small part of the total habitat, for example, the snail in our woodland. This small part of the total habitat is called microhabitat.

reveals that it is not clear whether the author's intended meaning is the snail's habitat (which is a small part of the woodland — a habitat) is a microhabitat or the snail occupies our (whole) woodland, which is a microhabitat (of a larger habitat around it).

Reasoning needed is the amount of reasoning necessary to uncover the actual relationship in the text, as estimated by the designer. The motivation for including this information is not only the same as that of information about ambiguous pieces of text, but also this slot estimates the mental effort the learner will spend to map the relation ship onto one of the links provided by the system, after she has correctly identified it in the text. Currently the system uses a three-point reasoning scale: EASY — the designer believes the learner will have no problem in understand the relationship; DIFFICULT — when the designer anticipates the

[11] It is not really fair talking about a piece of text taken out of context. Lest you are not convinced, take a look at the text itself.

learner will have trouble when trying to understand and then map the given relationship; FAIR — if the degree of difficulty falls somewhere in between. The Justifications slot is a pair of justifications the program uses in order to provide feedback which justifies the choice of the expected link. This slot uses a prespecified format which will be further explored later in **Chapter 4**.

Finally, slot Where in text is not conceptually interesting; it simply tells the program where the relation can be found in the text, so that it can provide the learner with some visual feedback. As an example, the relationship between concepts ORGANISMS and HABITAT found in the text has the representation presented in **Table 3–10**[12].

Slot	Value
Link	PART OF
Original concept	ORGANISMS
Terminal concept	HABITAT
Is explicit	no
Ambiguous	no
Reasoning needed	fair
Justifications	[RJ2: organisms live in habitat, RJ4: if someone lives in some place, they are PART OF it]
Where in text	(0, 49)

TABLE 3–10: REPRESENTATION OF THE RELATIONSHIP BETWEEN ORGANISMS AND HABITAT

The representation of all relationships of interest found in the sample-text used by MAPTUTOR can be found in **Appendix A**.

3.8 The Importance and the Role of the Expert's Knowledge

In order to produce the ranking of concepts described above, the expert was asked to rate the list of concepts[13] found in the text. In the ranking of the sample-text, a five-point scale was used, and the expert was asked to rank the concepts according to this scale, with the most important concepts having the highest scores. After this numerical ranking was completed, the concepts were divided into minor ones (i.e., those concept with rankings four and five) and major ones (i.e., those with ranking from one to three).

[12] The indexes preceding the justifications slot in this example will become evident latter.

[13] The designer should avoid including in this list those concepts which are too general (e.g., features in the sample-text).

It should be emphasised that the ranking of concepts depends on the subjective judgement of the expert (or instructional designer) about what the most important concepts in the text are. Major concepts correspond to what the instructional designer believes to be the concepts essential for the learner to understand the text at hand, whereas minor concepts are not essential for this purpose[14]. A major concept is required to be adequately mapped according to MapTutor's learnability criterion discussed in **Section 3.5**, whereas the learner is allowed to choose among the minor concepts the ones she prefers to include in her map. To reduce the dependency on the judgement of a single expert, this classification of concepts should ideally be carried out by, say, three experts instead of only one. The resulting rankings would then be taken to be the averages of the rankings from the three experts. This approach has not been adopted in the current research simply because experts were in very short supply.

Usually, an expository text contains too many ideas (concepts) and relationships among them, so that the learner is not able to memorise everything (or even most things) she comes across when reading the text. Thus, the learner must focus attention on the important points of the text and engage in activities that help her to encode the main ideas, even when this means not learning other less important information (Anderson & Armbruster, 1982). The learner's knowledge about the task requirements, that is, her knowledge about what is to be learned or remembered from the text, may range from complete knowledge (e.g., when the teacher specifies in detail what the learner should learn) to almost no knowledge (e.g., when the teacher simply asks the student to study a book's chapter) (Anderson & Armbruster, 1982). It has been shown that the learner achieves better results (i.e., learns more) when the requirements are specific and explicitly provided than when they are too general and vague, or not provided at all. Learning goals are closely related to (but different from) task requirements. According to Rothkopf (1982), goals are used by the learner to identify text elements which are particularly important for achieving the task requirements. He also suggests that knowledge about the goals is a necessary ingredient for learning from text, since this knowledge,is responsible for guiding the learner's selective processes (i.e., for determining what is important and what is not important for the particular learning task).

What follows from the discussion above is that when the learner's goals conflict with the task requirements, specified in the current situation by the instructional

[14] For instance, a concept which could be substituted for another (different) one without jeopardising the text's general meaning should not be considered as essential for its understanding. Thus, this concept would not be expected to be a major one.

designer by means of the concept rankings described above, she may have trouble in constructing her map under the supervision of MapTutor. For example, if the learner believes that a given concept is not so important as to be included in her map, and in fact this concept has been classified as a major one, the successful task criterion (see **Section 3.2.2**) adopted by MapTutor will not be fulfilled. The difficulty faced by the learner in this example would be similar to that which would happen in a conventional, analogous situation in which the task requirements (e.g., summarising the main points of a chapter) specified by a teacher conflicted with the learner's own goals. To avoid this sort of problems, the learner ought to be told what the learning objectives (i.e., the essential to-be-mapped concepts) are before starting a training session with the program.

3.9 The Influence of the Learner's Prior Knowledge

The knowledge the student brings to a reading task can be divided into (Dole, Duffy, Roehler, & Pearson, 1991): (1) specific knowledge (i.e., knowledge about the topic); (2) general knowledge (e.g., knowledge about how to draw inferences); (3) knowledge about the organisation of the text and (4) knowledge about reading strategies (including metacognition). The learner uses prior knowledge to rate the importance of segments of the text (Dole et al., 1991); to draw inferences (McKoon & Ratcliff, 1992); to elaborate the text (Reder, 1980); to monitor comprehension (Brown, Armbruster, & Baker, 1986); and ultimately to construct a model of comprehension (van Dijk & Kintsch, 1983). Linguistic knowledge is also a key factor for learning from text (see, e.g., Breuker, 1984)[15].

It seems clear that the processes of extraction and encoding of meaning of a text depend as much on the learner's prior knowledge as on the characteristics of the text itself. Applying a given graphical strategy to a piece of text consists essentially in selecting important concepts and relationships, and then depicting them in a diagram following certain conventions. Most graphical strategies require the learner to make explicit the relationships using a constrained set of links or special symbols. Relationships between concepts are not always explicit in the text, and thus, often need to be inferred by the learner. This happens because no text contains explicitly all the information it conveys.

[15] Other factors such as affective and conative processes (see, e.g., Snow & Farr, 1987); styles of learning (see, e.g., Entwistle, 1981; Messick, 1987); intelligence (see, e.g., Sternberg, 1988); and individual differences (see, e.g., Snow, 1989) can also play a crucial role in learning.

It follows that inferences[16] have to be made in order to understand the content of any text. In order to draw inferences, the reader uses prior knowledge and knowledge about the parts of the text already read (Just & Carpenter, 1987).

The necessity of inferences and its dependency on the learner's prior knowledge illustrate the fact that applying the mapping strategy does depend upon this latter factor. Inferences are essential for uncovering hidden relationships, which in turn is essential for mapping. The data collected from the exploratory study described in **Chapter 6** and **Appendix D** seem to indicate that the influence of prior knowledge can be beneficial[17] or detrimental[18] for learning the mapping strategy using MapTutor.

MapTutor's target-users should somehow be familiar with the domain represented into the program, but they are not required to have any knowledge about the mapping strategy. Also, one could further anticipate the influence of the learner's prior knowledge along two dimensions: (1) familiarity with the domain; and (2) familiarity with the mapping strategy. **Table 3–11** summarises the four extreme cases (i.e., familiar/not-familiar) regarding these dimensions of prior knowledge. These cases are further explained below:

Case I: The learner is familiar with both the domain and the mapping strategy. In this case, she is expected to have little gain in learning since she already knows how to use the strategy and will learn little (if anything) about the domain.

Case II: The learner is familiar with the mapping strategy, but not with the domain. In this case, she will benefit from learning the domain as well as by practising more the strategy.

Case Ill: The learner is familiar with the domain, but not with the mapping strategy. In this case, she is expected to benefit most because she will spend most of her cognitive effort learning the strategy itself (as opposed to trying to make sense out of an unfamiliar domain).

[16] Inference, according to McKoon & Ratcliff (1992), is 'any piece of information that is not explicitly stated in a text' (p. 440).

[17] For example, when the learner has already some previous knowledge about the strategy — as in the case of subject $\mathbf{S}_5$ in **Appendix D**.

[18] For example, when the learner insists on representing her own naive knowledge or MapTutor does not possess the necessary background knowledge — as in the case of subject $\mathbf{S}_4$ in **Appendix D**.

Case IV: The learner is neither familiar with the domain nor with the map ping strategy. In this case, she might still benefit from learning the strategy, but as she might have difficulties in understanding the domain, she will probably be overburdened.

	Familiar with domain	Not familiar with domain
Familiar with mapping	**Case I**: Learner will benefit least	**Case II**: Learner might benefit
Not familiar with mapping	**Case III**: Learner will benefit most	**Case IV**: Learner might be overburdened

Table 3–11: Expected Influence of Prior Knowledge in Learning How to Map

3.10 Conclusion

This chapter outlined the general strategy followed by MapTutor - a computer pro gram which has been proposed as a solution for teaching graphical mapping. It has also presented the program's architecture as well as its knowledge component. It could be argued that with the way this knowledge is defined it explores very few ideas of artificial intelligence, as Feifer (1989), for example, did. Nevertheless, a more detailed (i.e., more fine-grained) level of representation is unnecessary to the use MapTutor makes of its knowledge base. The knowledge structure of MapTutor presents two great advantages in relation to Sherlock: (1) it is much easier to both specify and implement than the semantic network employed by the latter program; and (2) more important, despite its simplicity, it is sufficient to identify causes of error at the right level and thus provide appropriate individualised feedback.

MapTutor's goal is to assist a student in drawing a map representing her under standing of a given piece of text. Conceptually speaking, the cognitive tasks involved in the use of a graphical strategy consist in breaking the text down into parts, identifying the relationships among these parts, and then representing them using the syntax adopted by the graphical strategy (Weinstein & Mayer, 1986). Thus, it seems clear that the application of a graphical learning strategy involves the entanglement of understanding the text at hand as well as the technique itself. Therefore, in addition to the teaching of a graphical learning strategy, the program is also concerned with the acquisition of declarative knowledge from educational text. As such, the program diagnosis and feedback are based upon two types of knowledge: (1) about the text being read; and (2) about the map being constructed. Nonetheless, despite the inherent entanglement between understanding the text and learning how to map, it should be emphasised that the current research has been primarily concerned with how to facilitate the training of a graphical learning strategy. In other words, the learning of the text should be seen as a by-product of the program.

A learning strategy can be classified according to its generality. A content-dependent strategy is one which is tailored to particular types of text. Content-independent strategies, on the other hand, are intended to be general and in principle can be applied to any educational text, although students may have difficulty in adapting the strategy to particular types of text they come across (Dansereau, 1985). All the graphical learning strategies mentioned in this book are to large extent content-independent. Accordingly, the expected generality of MapTutor is that the knowledge about the mapping strategy the learner acquires ought to transfer (ideally) to most other mapping situations (with/without using computers) involving educational texts. However, MapTutor does not purport to teach graphical learning strategies in general, since, as seen in **Chapter 2**, there are a wide variety of such strategies, and the syntax/semantics employed by some of them may be rather different from the mapping strategy. For example, the knowledge acquired by using the program is not even expected to transfer to concept mapping (see **Section 2.3.5**; see also Novak & Gowin, 1984), despite the apparent similarity between the two strategies. The main reason for this, in this particular example, is that the concept mapping strategy (as defined by Novak & Gowin, 1984) emphasises the ideas of subsumption (Ausuel et al., 1978) and hierarchy (see **Section 2.3.5** for further details), whereas the mapping strategy implemented by MapTutor does not. Thus, in summary, the users of the program are expected to learn the mapping strategy by means of a systematic analysis of understanding possible relationships between concepts, and depicting them using the syntax of nodes and links employed by this particular strategy. This includes mapping the actual relationships found in the text at hand onto the set of canonical links provided by the system.

Finally, an important point of this chapter are the definitions of the criterion of concept learnability and the successful task criterion adopted by MapTutor. The importance of these definitions is that they allows us: (1) to quantify how much of the learning objectives the learner has already accomplished in any instant of an interactive session with the program, and thus to be able to decide when the she has or has not already done enough; (2) to have some hint as to where look for potential source of misunderstanding. The latter will be explored next chapter.