Chapter 7 Conclusions and Future Directions

7.1 Contributions to Knowledge

The main contribution of the current research is the development and full implementation of a computational method for teaching a graphical learning strategy. Moreover, an important implication of this research seems to be that perhaps mapping is not as complex as previously imagined provided a fair text and a good learning environment containing multiple forms of feedback are provided. Furthermore, precise diagnosis and immediate feedback may not always be necessary when those conditions are satisfied.

In order to contrast and show advances achieved, the current research will be compared with Sherlock research along three dimensions: inherent features, research features, and experimental work. These comparisons, summed up in **Table 7–1** to **Table 7–3**, are explored below.

Inherent Features. Those features which are necessarily part of the underlying theoretical method and program which implements it are referred to as inherent features. **Table 7–1** summarises the comparisons made between these features inherent to each method and program.

- **Interface**. Sherlock's interface is far simpler than MapTutor's, but the former is also a far cry from reality. In Sherlock, concepts are already part of the initial map, and the learner does not have to select them in the text — as required by any learning strategy; the program does not offer any actual facility for mapping itself; the learner is not even allowed to draw the links by herself: the program does this on her behalf. By contrast, MapTutor's interface (see **Chapter 5**) presents the learner with a full-fledged environment for mapping: from text — in which the learner can select the concepts to include in her map — to mapping tools providing short-cuts which makes the mapping task easier for the learner.
- **Knowledge Representation**. MapTutor uses a relatively simple knowledge representation schema (see **Section 3.7**; see also **Appendices A**, **B** and **C**), but it provides all functionality the program needs. Neither MapTutor nor Sherlock will work when the knowledge base is incomplete, but this problem is inherent to any knowledge-based system. Sherlock has some

deductive capability which allows it to reason over links. This issue has not been addressed by the current research.

- **Learner Model**. MapTutor's successful task (see **Section 3.2.2**) and concept learnability (see **Section 3.5**) criteria allow the program to know when the learner has successfully demonstrated a solution to the mapping task, and thus various map configurations are admitted by the program. Link usage statistics provide a clue which guides part of MapTutor's diagnostic process. Sherlock's learner model corresponds to the learner's possible interpretation of icons representing concepts on the screen.
- **Mapping Obstacles**. According to the current research (see **Chapters 4** and **6**), finding concepts in the text, slips and misconceptions are the main obstacles the learner faces when drawing a map. MapTutor tackles all these obstacles, whereas Sherlock's research takes only misconceptions into consideration.
- **Misconceptions and Diagnosis**. The current research proposes that the causes of misconceptions (i.e., wrong links) lie in misunderstandings of concepts, relationships between the concepts, and meaning of link types (see **Chapter 4**). On the other hand, according to Feifer (1989), these misconceptions are caused by misunderstanding of (concepts represented by) icons, plans, and missing or wrong facts about the domain. A closer look at both point-of-views shows that the main source of disagreement is Feifer (1989)'s argument that learners use plans (like the one presented in **Table 6–1 on page 149**) in order to decide which link to use. The current research did not attempt to invalidate his argument, but even if this were the case, the form of feedback derived from such plans has proved to be ineffective (see Feifer, 1989, **Chapter 8**).

  MapTutor establishes whether a link is right or wrong by deterministic look-up, whereas Sherlock does the same by means of spreading activation search. It is interesting to comment upon the contentious argument Feifer uses against deterministic search. Feifer (1989, p. 141) suggests that it would be virtually impossible, due to the large number of possibilities, to anticipate the possible correct links. Nonetheless, MapTutor was able to spot right/wrongs links most of the time while using only 52 instances of the relationship prototype (see **Appendix A**). Moreover, MapTutor failed at spotting three possibly correct links simply because it did not have the necessary background knowledge (see below), which was not taken into consideration in the experimental studies.

With regard to determination of causes of misconceptions, MapTutor uses causes ordination to guide the search, whereas Sherlock always asks as guidance. MapTutor could have been more efficient if it always asked the learner before starting the diagnostic process, but in practical terms, this would be very annoying to the learner.

- **Feedback and Help Support**. MapTutor presents multiple forms of feedback, and this appears to be effective despite the fact that diagnosis is not so accurate. MapTutor offers three basic forms of feedback: opportunistic (see **Section 4.9**), user-demanded (see **Section 5.7**) and post-task (see **Section 4.9**). These forms of feedback are mostly concerned with the teaching of concepts, meanings of links and relationships. MapTutor also provides help for the learner by means, of its associated help and tutorial programs (see **Section 5.5**), as well as by suggesting concepts for her to consider when she is stuck (see **Section 4.9**). By contrast, Sherlock uses only opportunistic feedback about plans (containing uninstantiated variables) and facts about the domain.
- **Research Facilities and Generality**. The current method appears to hold more promise for further development, because of the research facilities it provides. MapTutor does not rely upon other methods (e.g., semantic network classification) or special links (e.g., not) in order to work. On the other hand, Sherlock appears to be specially tuned (i.e., ad-hoc programmed) for the particular sample-text and set of links employed. It is not known to what extent Sherlock needs those ingredients to work. Also, development time of new instructional material in MapTutor is likely to be shorter than Sherlock's, because the latter needs a great deal of reprogramming.

  MapTutor's end-of-session reports (see **Section 5.8**) has proved to be a very useful research tool during the experimental studies described in **Chapter 6**. Furthermore, the program has other research facilities which were not tried out during the experimental work described in this book. These facilities include multiple diagnostic strategies (see **Section 4.5**) and on-line evaluation (see **Section 6.2.1**).

  There are yet no experiments which provide enough evidence to claim that MapTutor is domain-independent. However, it seems fair to suggest so, because the method and program which implements it use both the sample-text and the current set of links with the sole purpose of providing examples. Perhaps it will not work well with all types of text, but there is no restriction against qualitatively different domains or link types.

Research Features. Those features which are not part of either the methods or the programs themselves, but have been chosen by the authors in order to demonstrate them, are considered as research features. The comparisons between the research features of each method are summarised in **Table 7–2**.

- **Link Types**. The research literature on graphical learning strategies provides some empirical support to all links used by MapTutor. On the other hand, Sherlock used a very odd link — namely, link NOT, which has no support in the literature. According to Feifer (1989), the learner would be expected to draw a NOT link between concepts X and Y whenever the antecedents of the following rule (plan) were satisfied (pp. 66–7):

  IF X is not a Y

  AND Y is not a X

  THEN Make a NOT link from X to Y

  Thus, NOT would be expected to be drawn whenever the link between two concepts was neither IS A nor EQUIV. According to Feifer's definition, link NOT also applies whenever there is no relationship between the concepts. MapTutor has currently 24 concepts represented in its knowledge base, and there are 21 expected IS A links plus one expected EQUIV link. Therefore, according to Feifer's 1989 theory, there should be *530* NOT 1inks[1]. If we are fair enough to require that only one (instead of two) NOT link be drawn between two concepts, we will still need to represent *530/2 = 265* NOT links. Take into consideration that the best mapper who took part in the experimental sessions with MapTutor spent about 23 minutes to draw 23 links (i.e., about one link/minute), she would have to spend about 4.30 hours only to draw NOT links. But even worse are the cognitive implications of link NOT. For example, it seems reasonable that knowing that finger is part of the human body is knowing much more than knowing that finger is not an aeroplane, finger is not a snail, etc.

- **Text**. MapTutor used a three-paragraph, 191-word long text extracted from the Biology domain. The analysis of this text resulted in 24 concepts considered important for its understanding. Sherlock used a two-paragraph, 116-word long text in Business Law. MapTutor used an original excerpt, whereas Feifer (1989) is not clear about what he meant by 'edited

[1] There are 24 choices for the first concept; once this has been chosen, there are 23 concepts left to be chosen as the second one; from this total 22 links cannot be NOT. Therefore, there are 24 x 23– 22 = 530 NOT's.

version' (p. 6, footnote 6). This 'edited version' of text yielded 26 concepts (see more about text considerations below).

- **Background Knowledge**. Sherlock used a good deal of background knowledge gathered during pilot studies, whereas MapTutor did not represent any sort of background knowledge explicitly. Nonetheless, there is no impediment to do so (see **Section 7.2**).
- **Text Representation**. MapTutor needed 24 instances of concept prototype (see **Appendix B**) plus 52 instances of relationship prototype (see **Appendix A**) to represent the sample-text. On the other hand, Sherlock used 150 nodes and 330 links for representing the text (including background knowledge) in its semantic network.
- **Links Representation**. MapTutor needed only 5 instances of link prototype to represent its 5 link types (see **Appendix C**), whereas Sherlock used 115 nodes and 216 links in its semantic network plus 16 production rules to represent six link types.

Experimental Work. The series of experiments carried out to provide empirical support for each method. This is the most difficult term of comparison between the two bodies of research, because the data collected, as well as the procedures and analyses carried out in each work were rather different. This means that the results simply do not appear to be comparable. Even so, **Table 7–3** attempts to compare MapTutor and Sherlock in terms of experimental work.

The trouble with comparing the experimental work is not only regarding quantitative results, but also (and mostly) concerning quality of results. Feifer (1989) claims that one major the source of misunderstanding in mapping is that of interpretation of icons (concepts represented by boxes). According to Feifer (1989, p. 38), 'The interface would be too cluttered if each concept were defined completely on the icon. The learner must, then, decide what conceptual entity an icon refers to.' The results of the experiments described in this book seem to indicate that the problem of icon interpretation may not be as serious threat to mapping as Feifer imagined. Very few subjects who took part in the experiments with MapTutor stated they could have misunderstood icons representing concepts. Moreover, only one subject (**S**$_7$) committed a serious mistake which could be attributed to interpretation of icon in Feifer's classification, and even so, this happened in exceptional circumstances (see **Appendix D**). Could these discrepant results be ascribed to differences in the text? The answer is: In part, it is very likely. Feifer (1989) seemed to have used a much more difficult text. The *Consideration* text used by Feifer used 116 words to describe 26 concepts (i.e., 0.22 concept/word); MapTutor's sample-text on Habitat used 191 words encompassing 24 concepts (i.e., 0.13 concept/word). This means that Feifer's text was much more dense than the one used in the current research. Moreover, Feifer's domain was completely unknown for his subjects (whose profiles were comparable to the subjects' used in the current research), whereas most subjects who used MapTutor had at least heard about most of the concepts in the text. Should MapTutor have used a more difficult text? The answer is categorically, No. Remember that the main desideratum of both MapTutor and Sherlock is to be able to teach how to use the mapping strategy, and difficult texts are not suitable for teaching a learning strategy because they may overburden the learner and make learning much more difficult to take place (see, e.g., Kozma, 1992; Norman & Bobrow, 1975; Royer, Cisero, & Carlo, 1993). The choice of the sample-text (see **Section 3.3**) used in the experiments described in the current research tried to take these pitfalls into consideration.

Feature	MapTutor	Sherlock
Interface	Close to real-world mapping Full-fledged mapping environment: text, links and mapping tools	Too simplified Concepts already the map No actual facilities for mapping
Knowledge Representation	High-level prototypes of concepts, links and relationships	Low-level primitives Semantic network Production system
Learner Model	Concept learnability criterion Links usage Successful task criterion	Interpretation of icons
Mapping Obstacles	Finding concepts in text, slips and misconceptions	Misconceptions
Misconceptions	Misunderstanding of concepts, links, and relationships	Misinterpretation of icons Flawed plans Missing/wrong facts
Diagnosis: Right/Wrong Link	Deterministic	Spreading activation search

Feature	MapTutor	Sherlock
Diagnosis: Causes of Errors	Hardly asks Ordination of causes	Always asks Flawed plans (buggy rules)
Feedback	Opportunistic, user-demanded, and post-task Concepts, links, and relationships Text reference On-line definition of links	Opportunistic Facts Plans with variables
Help Support	Help and tutorial programs Suggestion of concepts	None
Research Facilities	Multiple diagnostic strategies On-line evaluation High-level reports Domain independence	Limited transcript (trace)
Generality	Domain-independent text and link types	Ad-hoc programmed

Table 7–1: MapTutor vs. Sherlock I: Inherent Feature

Feature	MapTutor	Sherlock
Link Types	Literature support	Link NOT has no support in the literature
Text	Three-paragraph, 191-word long Domain: Biology Original excerpt 24 concepts	Two-paragraph, 116-word long Domain: Business Law Edited version 26 concepts
Background Knowledge	No	Yes
Text Representation	24 instances of Concept Prototype 52 instances of Relation Prototype	150 nodes and 330 links in semantic network

Feature	MapTutor	Sherlock
Links Representation	5 instances of Link Prototype	115 nodes and 216 links in semantic network 16 production rules

Table 7–2: MapTutor vs. Sherlock II: Research Features

Feature	MapTutor	Sherlock
Subjects	7 PhD students	4 mature subjects
Session Duration	23 min to 60 min	90 min
Drawn Links	Maximum: 60 Minimum: 19 Total: 181	Maximum: 19 Minimum: 10 Total: 70
Selected Concepts	Maximum: 23 Minimum: 9 Total: 120	None
Diagnostic Accuracy	Wrong links: 61 out of 64 Correct links: 117 out of 117	Wrong links: 25 out of 26 Correct links: 32 out of 33
Feedback	Appears effective	Ineffective

Table 7–3: MapTutor vs. Sherlock III: Experimental Work

7.2 Limitations of the Research and Future Directions

The results from the preliminary evaluation have suggested various re-evaluations of both the method and the program which implements it. What follows are some revisions which should be implemented before a new series of experiments take place.

7.2.1 Supporting Organisation

Most subjects who took part in the experiments liked to organise their maps in their own way even before starting linking the concepts. However, they did not like to come back to the text in order to select a concept before being allowed to drag it. This seems to suggest that including a **Dragging Tool** in the **Tools Pane** would be a good idea. The trouble with implementing this idea in MapTutor in a straightforward manner is that the idea of concept selection is reminiscent of an old design of the program which pervades a big deal of it. MapTutor's tools grid was only developed when the author began to plan the experiments described in **Chapter 6** and realised that the program was not as easy to use as initially imagined. The tools grid has proven to be a major improvement in MapTutor's interface which required only minor modifications of the program. On the other hand, including a Dragging Tool does require some major changes. Furthermore, it cannot be predicted (before implementing it) how this new tool would fit the current semantics of the interface.

7.2.2 Turning MapTutor Less Obtrusive and More Friendly

Currently, MapTutor appears to be obtrusive and the feedback messages are a bit too long. The program ought to take advantage of the fact that it already provides a great deal of feedback in various formats, and reduce the amount of opportunistic, explicit feedback. Thus, a promising idea would be to have the program relinquish its assistance (i.e., automatic feedback) and let the learner assume more and more responsibility as the session progresses until a point is reached that all messages would be user-demanded ones only[2]. Also important would be having a Feedback Tool responsible for providing feedback and advice about the currently selected link.

Other non-obtrusive (e.g., visual) forms of feedback would also play significant role. For example, the use of green and red drawn links corresponding to right and wrong ones, respectively, may prove to be very helpful for the learner to follow her performance.

Having a **Definition Tool** which would provide an alternative definition of a selected concept appears to be another option, because this makes the diagnosis and automatic concept feedback unnecessary. The latter appeared to be ineffective anyway.

The overwhelming number of slips perpetrated by subject $\mathbf{S}_4$ (see **Table D–1**) suggests that the program should take some precautions against distracted learners. A way of reducing slips would be to have always a selected link name only upon the learner's own request. For example, a single mouse click would select a link name only for immediate use, whereas a double click would work as a

[2] This is called *teaching scaffolding* and is suggested by Dole et al. (1991).

single click currently does: keep the selected link name as the current one until the learner chooses another one.

The experimental studies served also to point out some feedback features missing in MapTutor. For example, as seen in **Chapter 6**, subject $\mathbf{S}_4$ proved to have understood concept HABITAT, but instead of concentrating on other major concepts afterwards, she preferred to turn her attention back to this concept. As she could not find other relation ships between HABITAT and other concepts, she wanted to represent her own knowledge of HABITAT, which the program did not accept. Her frustration could have been avoided by MapTutor if she were told that she had already proven to know HABITAT, so that she could concentrate on other concepts instead. For example, the use of, say, green concepts to indicate when the learner already knows a concept seems to be very promising. This would suggest the learner to map other concepts she has not proved to know yet.

Diagnosis can also benefit from this new less obtrusive interface because, it will be possible to ask the learner before diagnosing without making the program (as a whole) too much obtrusive. Asking more and guessing less, will certainly make the diagnostic process more reliable.

7.2.3 Improving MapTutor Reasoning Capabilities

According to Ausubel (1963), meaningful learning will occur only when the new information can be related to what the learner already knows. Therefore, prior knowledge is critical for learning from text, and the learner should be allowed to elaborate its graphical representation of a text with information from her prior knowledge. The closed-world assumption in **Section 4.5** constrains the learner too much in mapping her own knowledge. Perhaps, the most obvious solution would be to include a great deal of background knowledge, but even so, it will hardly be complete (i.e., there will always be something left out), but there is some room for improvement here.

The expectation is that when MapTutor cannot determine the cause of the error (i.e., in the case where there is no relationship between the concepts in the text), it should not tell the learner that the link is wrong. Instead, it should say that there is no such relationship in the text, but if the learner thinks it is fine, then it should tell her to keep going. Thus, a simple rewording of the feedback message seems sufficient for making the program more credible. A more sophisticated solution, however, would be to endow MapTutor with the capability of learning from the learner's prior knowledge. In this case, the instructional designer responsible for the program would be in charge of validating this new knowledge, before putting it to use. MapTutor already does part of this task, since it records all links made by the learner and presents this information, in a high-level form, to the researcher, who can then make use of it. What is lacking is the capability of translating and storing this new information in, say, a temporary knowledge base while waiting for the design to validate it or not.

7.2.4 Definition of Rules of Interaction among Links

As seen, MapTutor is not able to reason about relationships among links. This depends on deeper investigation in the links themselves, which is missing in the literature of graphical learning strategies. In order to keep the intended domain-independence, as well as to be able to investigate other (perhaps more important) aspects of mapping, this has been left out of the current research.

An example which suggests the necessity of investigating the interaction among relationships comes from subject $\mathbf{S}_1$ (see **Appendix D**) when she wanted to link FRESH to ABIOTIC FACTOR by means of IS A, representing the proposition fresh is an abiotic factor. Should MapTutor tell the learner she was wrong when she did so? After all, FRESH is a TYPE OF WATER, which in turn is an ABIOTIC FACTOR. Thus, by inheritance, FRESH is an ABIOTIC FACTOR. But, by the same token, FRESH is a THING[3], and just knowing that fresh is a thing is, from an educational standpoint, tantamount to knowing nothing at all about concept FRESH. Perhaps, the best solution in cases like this would be to recognise this fact and to tell the learner her link was not so bad, but she would rather link FRESH to ABIOTIC FACTOR by using TYPE OF WATER as an intermediary concept.

Another problem partially investigated was that concerning the equivalent relation. The equivalence rules described in **Section 3.5** do not jeopardise domain independence, because many expository texts contain definitions, synonyms, etc., where those rules apply. Link EQUIVALENT is not even required to be present; instead, it is allowed to have a better treatment than otherwise. Yet, this treatment can still be improved by having MapTutor identify and provide more effective feedback than currently it does. For example, if the learner draws a link meaning that SOIL is an ABIOTIC FACTOR, and afterwards draws a another link meaning that SOIL is part of PHYSICO-CHEMICAL FACTOR, the program will tell her that her latter link is wrong but not that it is inconsistent with the former, since ABIOTIC FACTOR and PHYSICO-CHEMICAL FACTOR are equivalent concepts.

[3] This would be the highest level concept in a global ontology (see, e.g., Lenat & Guha, 1990).

7.2.5 A Better Inverted Link Treatment

It seems reasonable to expect the learner to draw links in a consistent manner, but she should not be required to draw a given link type in the direction preferred by the instructional designer. No subjects complained about the semantics chosen according to the author's own intuition during the experiments, but this may not always be the case. Thus, a treatment for consistency regarding link direction better than the one currently in use by MapTutor would be to identify the learner's preferred direction for her links. A simple way of implementing this would be to ask her preferences for link semantics at the outset. These preferences would include not only directions, but also link labels.

7.2.6 Authoring Interface

An authoring interface for MapTutor (see **Figure 3–1**) would probably improve its research potentiality, since it would allow the instructional designer to represent various link types and text much easier than currently it is. The starting point of designing an authoring interface for MapTutor is to take advantage of the current layout of the program's interface itself[4].

When entering a new lesson (i.e., a new knowledge base), the designer would be prompted to enter the text in the first place. Then, the authoring program would prompt the instructional designer to select portions of the text to be used as concepts. Once selected, these concepts would be drawn in the map pane in a way similar to what MapTutor's interface itself does. Also, each concept would have the slots corresponding to its concept prototype (see **Section 3.7.1**) filled in with default values. The designer would then be responsible for editing the instances of the concept prototype to suit her needs, but the program would also provide some facilities for this purpose. For example, by letting the designer position the concepts in the desired default location, the program would be able to record the coordinates of the chosen position.

The creation of instances of the relationship prototype (see **Section 3.7.3**) referring to the various relationships among the concepts would be facilitated by allowing the designer to draw links between the concepts. These links could then be edited taking as a model a template based on the relationship prototype. Editing link types appear to be the easiest part of the implementation. It would simply

[4] Indeed, part of the knowledge currently represented in MapTutor's knowledge base was acquired by using an old modified version of the program's interface. This version was able to provide the positions of selected pieces of text as well as of the coordinates of concepts in the map pane.

require the instructional designer to edit a template based on the link prototype presented in **Section 3.7.2**.

Finally, establishing an intercommunication protocol between the authoring interface and the main program would allow the designer to try out the newly created knowledge-base. This would also allow the designer to go from one program to another whenever she wanted.

The current difficulties in implementing this roughly specified authoring interface simply lie in: (1) specifying and implementing the details, and (2) modifying MapTutor's current manipulation of knowledge bases.

7.2.7 Full-Scale, Long-Term Experimental Work

The experiments carried out with MapTutor addressed questions which could be answered within the time allowed by the current research project. They were also useful to validate some decisions anticipated in the design of MapTutor, as well as to understand and explain the program. However, the long-term expectation is that the extended use of MapTutor will influence the learner's characteristics by making available to her the mapping strategy which, in turn, will enhance her performance in learning from text. Thus, a systematic evaluation of the effects of MapTutor as a supplemental learning program is necessary. The best way of evaluating MapTutor is to conduct experiments by using the hypothetico-deductive paradigm with pre-/post-tests, so as to estimate whether MapTutor can really achieve its objective of teaching mapping (see **Section 6.3**).

7.3 Concluding Remarks

Some years ago, the Venezuelan government embarked upon a seemingly Herculean task: to increase the intelligence of its whole population (Dominguez, 1985; Nickerson, Perkins, & Smith, 1985; de Bono, 1985). This book is far from this ambition, but it reflects a current trend amongst researchers who worry about the teaching of higher order thinking.

This research has asked and attempted to answer whether it would be possible to teach students a graphical learning strategy by means of a computer tutor, and what techniques should be used in designing such a tutor. Preliminary evaluation of MapTutor indicates that the proposed solution is at least potentially more promising in educational terms than previous research. MapTutor addresses mapping as a whole, whereas Sherlock only addresses the issue of diagnosis, and perhaps this is not so important for teaching mapping. MapTutor seems to work because it is an integrated environment. Moreover, MapTutor is easy for the intended users to understand. The learner can easily understand its external behaviour, and the instructional designer can easily understand its internal behaviour by examining the self-reports generated the program. Last, those experimental studies also suggested interesting directions for future research which could lead to definitive answers to the questions raised in the current research.