Chapter 1 Introduction

1.1 Introduction to Graphical Learning Strategies

Reading is perhaps the most important way of conventional learning. The ability to read well is even more important nowadays than earlier, taking into consideration the amount of information in circulation and readily available through computer networks. Learning from text requires the learner to attend to incoming information, access her prior knowledge, make inference, resolve conflicts so as to accommodate the new knowledge, and finally, encode the new information into memory so that she will able to recall it when needed. Purposeful reading, thus, is a rather complex activity.

Learning strategies exist that have been shown to enhance understanding of written material. A **learning strategy** is a set of activities in which a learner engages in order to facilitate the acquisition of knowledge (Derry & Murphy, 1986; Weinstein & Mayer, 1986). Common examples of learning strategies are highlighting, note-taking, summarising, among other. A **graphical** or **spacial** learning strategy is a means of constructing an alternative representation of a text which presents a diagrammatic layout of its contents. The major interest of application of graphical strategies in education is in the area of reading comprehension (see, e.g., Holley & Dansereau, 1984c; Lambiotte, Dansereau, Cross & Reynolds, 1989). Graphical Representation of text typically takes the form of networks of nodes and links — representing respectively concepts and relationships among concepts and domains. These representations have been given names such as **concept maps** (Novak & Gowin, 1984); networks (Holley & Dansereau, 1984b); **schematisation** (Breuker, 1984); **semantic network** (Fisher, 1990); **mind maps** or **pattern notes** (Buzan, 1989; Jonassen, 1987); and **knowledge maps** (McCagg & Dansereau, 1991). Although many researchers claim that their graphical strategies are grounded on principles such as schema theory and propositional network, Lambiotte et al. (1989) believe that intuition is the main *rationale* behind those strategies.

Research literature points out various hypothetical advantages of the graphical learning strategies. For example, by making explicit the organisation of the content of the text, a graphical representation may improve learning (Just & Carpenter, 1987). Breuker (1984) add that such a strategy may provide a spatial arrangement and an external memory which helps one to reduce the cognitive demands associated with the processing of a piece of text. Moreover, graphical strategies are potentially good candidates for and ideal strategy since they appear to entail most of the processing, such as depth of processing and elaboration, that has shown to improve meaningful learning of written material (Goetz, 1984).

On the negative side, most researchers (e.g., Holley & Dansereau, 1984b) agree that it is not only difficult to employ graphical strategies, but also to train students in their use. McKeachie (1984) points out another problem associated with graphical strategies: '[they] may be cumbersome and time-consuming when students must master larger blocks of material' (p. 303).

1.2 Problems with Graphical Learning Strategies

Most learning strategies (be it conventional or graphical ones) follow an overall plan which runs as follows:

1. *Read* the text.
2. *Select* the material which is relevant to the learning objectives.
3. *Represent* the selected material according to the notation required by the learning strategy.

It follows that, when trying to execute a given learning strategy, the learner may have trouble in running any of these steps. For example, she may not know how to infer a given relationship between two concepts (i.e., she has trouble in **Step 1**); she may not know precisely what the learning objectives are, so that she selects parts of the text at hand which are irrelevant to those objectives (**Step 2**); finally, she may not understand the semantics of the new representation very well, so that it does not seem to make sense for her (**Step 3**).

Problems with **Step 3** appear to be overwhelming in the specific case of graphical learning strategies, and this may explain why those strategies are so difficult to learn and apply. The same does not appear to apply to conventional learning strategies — even those most demanding ones (e.g., summarising) — because, in most of such cases, the learner is very familiar whit the semantics utilised in the new representation, since it is either very close to or the very same as the written representation she has been taught since the early grades. The case of graphical learning strategies is another story, because they typically have a new notation with its own structure and semantics, which the learner must learn and apply in practice. Most graphical strategies require the learner to make explicit relationships between concepts using a constrained set of links or special symbols, thus making the task more difficult. Hence, the particular hard problem associated with graphical strategies is, once the relationship have been identified, *how they should be represented* (**Step 3** above).

Direct teaching of learning strategies has received little attention by researchers involved in the use of computers in education, despite the current trend amongst educational psychologists. The research described in this book is concerned with the problem of how to provide appropriate tutoring on graphical learning strategies by means of a computer-based tutor. This problem has not properly been taken into consideration previously. Sherlock — an intelligent tutoring system developed by Feifer (1989) — constitutes, to some extent, an exception to this state of affairs, because it is aimed at training students in graphical strategies. However, the practical educational objectives of this early program were not met, because it concentrated on student modelling with emphasis on bug diagnosis, rather than tutoring itself[1].

1.3 MapTutor: The Proposed Solution

Using the domain of graphical mapping with a constrained set of links, the learning strategy plan sketched above can be instantiated as follows:

1. *Read* the text.
2. *Select* the concepts of interest.
3. *Draw* the selected concepts on the map, choose links appropriate to each situation among the set of links provided, and connect the concepts together so as to represent the respective relationships between them as presented in the text.

It follows that, in order to correctly execute this instantiated **Step 3**, one must:

1. Understand each proposition being represented, including both concepts under consideration; and
2. Understand the meaning of the canonical links provided, so that the appropriate link can be chosen to represent each proposition.

This book identifies three forms of misunderstandings which may hamper this process:

- *Misunderstanding of the concepts under consideration*;
- *Misunderstanding of meanings of canonical links*; and
- *Misunderstanding of the piece of text in question*.

[1] Sherlock will be fully discussed in **Chapters 2** and **7**.

Thus, procedures have been proposed which are able to diagnose each of these forms of misunderstanding. Moreover, each of these procedures has an associated feedback procedure tailored to particular situation. The ideas above have been implemented in a computer-based tutor called MapTutor.

MapTutor is primarily aimed at facilitating the training of a graphical learning strategy — namely, graphical mapping using a constrained set of links. However, in contrast with Sherlock, it tackles the problem as a whole. **Figure 1–1** presents a finished graphical map resulting from a real interactive session with MapTutor[2].

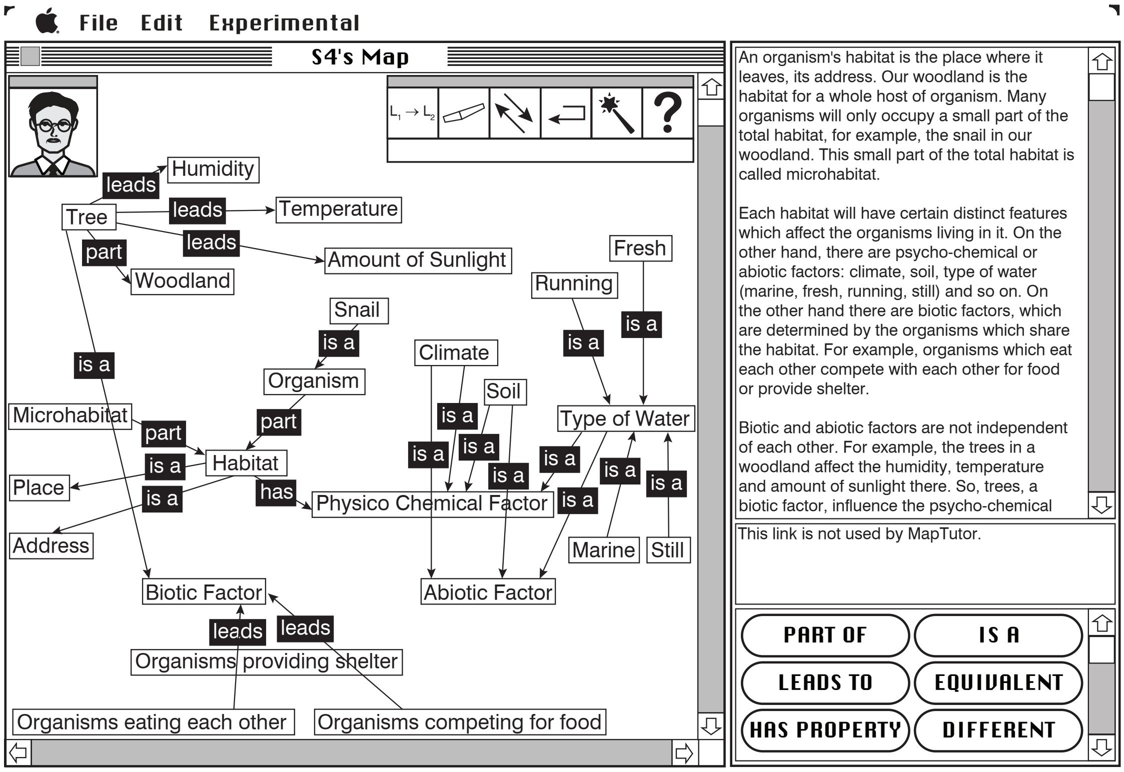

Figure 1–1: A Map Drawn Using MapTutor

The main advantage of MapTutor over Sherlock are that:

- MapTutor tackles all steps involved in constructing a graphical representation of a text, sketched above. MapTutor's interface provides the text from which the learner selects concepts, a set of link types to connect those concepts, and a collection of mapping tools which provides shortcuts for typical mapping tasks. By contrast, Sherlock provides no actual facilities for mapping. In the latter, not even the text is on-line and the learner has the concepts already drawn on her behalf at the onset.

[2] This map was constructed by Subject $\mathbf{S}_4$ during a series of experimental studies aimed at appraising the program.

- MapTutor defines criteria which allow the program to know when the learner has successfully demonstrated an acceptable solution to the mapping task, and thus various map configurations are admitted. There is no similar criterion in Sherlock which could be compared.
- MapTutor takes into consideration that different domains may require distinct sets of canonical links. Therefore, virtually any number of links can easily be represented into the program. Furthermore, MapTutor is domain-independent in relation to the primary subject matter. This means that virtually any text can be use by the program. Sherlock cannot represent another text or set of links without suffering substantial reprogramming.
- MapTutor's knowledge representation is relatively simple when compared with other artificial intelligence programs, such as Sherlock, which uses fine-grained knowledge representation. MapTutor's knowledge base is decomposed at a high-granularity level which is sufficient for its purpose.
- MapTutor takes into consideration that the main obstacles the learner faces when drawing a map are: finding concepts in the text, slips and misconceptions. Sherlock's research addresses only misconceptions.
- MapTutor provides the researcher with facilities — such as high-level, end-of-session reports — which allow to some extent that certain learner's behaviours be investigated. Sherlock's closest counterpart are low-level program traces.
- Last but not the least, MapTutor's multiple forms of feedback appear to be effective and learners usually accepted the program's suggestions on how to correct their mistakes. By contrast, Sherlock's feedback based on facts about the domain and on plans (i.e., production rules) representing strategies for drawing lines does not seem to be very helpful, as Feifer (1989) himself suggests.

MapTutor has not been fully tested and does have some acknowledged weaknesses, but initial results from a small-scale formative evaluation conducted to appraise the program are encouraging and seem to indicate that this is right direction to be followed in future research.

In summary, a graphical learning strategy, in which the learner constructs a diagram depicting the main ideas underlying a text, can be an effective means of studying from texts. However, these strategies can be difficult and time-consuming to learn. This book proposes that a computer-based tutor can provide an effective means of teaching the mapping technique. Such a method has been tried in Sherlock — an earlier intelligent tutoring system. Yet, that system had fundamental limitations that the current work aims to overcome.

1.4 Overview of this Book

The rest of this book is organised as follows:

- **Chapter 2** focuses on research into graphical learning strategies. This chapter reviews graphical learning strategies research, provides some examples of graphical strategies, and discusses some approaches which have been used to teach them. It argues that none of the approaches investigated can be implemented in the computer in a straightforward manner. Finally, this chapter presents and criticises a computational attempt to model the problem of teaching a graphical learning strategy.
- **Chapter 3** is dedicated to the central idea underlying this research. This chapter presents a new computational approach for teaching a mapping strategy. It begins with an overview of this approach, and then presents MapTutor — a concrete example which implements on the computer the ideas originated from the proposed training approach. The discussion proceeds by describing the architecture as well as each individual component of the system necessary for implementing this training approach. This overview includes how these components interact to provide individualised tutoring. The rest of this chapter is devoted to the knowledge component of MapTutor, including how text and links are represented in the program's knowledge base.
- **Chapter 4** describes a mechanism dedicated to identifying causes of errors. Most of this chapter concentrates on MapTutor's diagnostic procedures, which are based upon the knowledge representation schema employed to represent and update beliefs, discussed in the previous chapter. Teaching procedures which are able to deal appropriately with each value returned by the diagnostic procedures are also presented. These teaching procedures show how each diagnostic value can be accommodated into a kind of feedback tailored for each specific form of misunderstanding.
- **Chapter 5** is the most technical. It presents implementation details behind the construction of MapTutor. This chapter also sets out a guided tour through the program's interface and shows its operation. Finally, is presents the tutorial and help programs — two auxiliary programs associated to MapTutor which present tutorial and help, respectively.

- **Chapter 6** presents a preliminary evaluation of MapTutor carried out by means of an experimental study aimed at: (1) assessing the extent to which the program is able to diagnose correctly; (2) evaluating the impact of the program's feedback; and attempting to validate some design decisions presented in this book. This chapter also outlines a systematic large-scale evaluation which could be conducted by means of experimental research in order to evaluate the merits of the program as a mapping training tool.

- **Chapter 7** discusses the outcome of the research, the merits and limitations of MapTutor, and explores some directions for further work.

- **Appendix A** presents the analysis of the sample-text employed, as well as the representation of the relationships derived from it. **Appendix B** presents the representation of the important concepts extracted from the text, and **Appendix C** does the same with the set of link types utilised in this project. **Appendix D** presents analysis of the data gathered during the experimental studies described in **Chapter 6**, whereas **Appendix E** displays the questionnaire used in those experiments. Finally, currently, there are a number of computer applications (e.g., learning tools, graphical browsers, writer assistants) which use graphical representation of information to convey knowledge. Hence, familiarity with this type of representation may become an important qualification in the near future. This book concludes in **Appendix F**, by presenting a summary of some computer programs which purport to implement graphical learning strategies, such as networking and concept mapping. These programs are presented in this appendix with the objective of spotting some differences between them and MapTutor.