Chapter 3

Metacognition in Interactive Learning Environments

3.1 Introduction

The previous chapter reviewed research on metacognition in education. This chapter focuses on attempts to incorporate metacognition instruction into Interactive Learning Environments (ILEs) and Intelligent Tutoring Systems (ITSs).

Computers enable new and unique ways of capturing and representing metacognitive knowledge and metacognitive skills, such as graphical reification, and (abstracted) replays and reviews. They can also improve on traditional metacognitive approaches, like self-explanation, and collaboration by presenting them in a more dynamic and interesting form.

The main issues regarding the design of such metacognitive activities in ILEs are discussed in this chapter and a classification schema of the ways in which ILEs can target metacognition is proposed. Examples from the literature are brought together to illustrate the use of this schema and to give a summary of previous research on metacognition instruction and self-reflection in computer-based learning systems.

3.2 Computers as Metacognitive Tools

Computers have a lot of potential as metacognitive tools. For example, through their ability to record interactions with users, they can become powerful reflection tools. Having captured the actions of the student carrying out a task, these can be played back to her, properly abstracted and structured. This will help the student to become aware of her processes and help her improve performance on the task in question through reflection on the how’s and why’s of the chosen problem solving paths.

As collaborative learning devices, they can be programmed to support group planning, monitoring and evaluation of the learning process. Students in a small learning group can, for example, look back over their solution paths and compare them with other members of the group. This should trigger reflection on which changes could be improved (Reusser, 1993). Another interesting possibility is that of simulated “Learning Companions” acting as peers who encourage the student to reflect and articulate her actions (Goodman et al., 1998).
Despite this potential, the majority of computer-based learning environments to date have focused on supporting students develop domain-related skills and knowledge.

Some attempts have been made to incorporate metacognitive components, mostly in the form of embedded reflection on the learning task or processes. And a very small number of systems have included explicit metacognition training as their main target. Detecting, tracing, modelling, and fostering students metacognitive and self-regulatory behaviours during learning in ILEs is a research challenge that is almost intact. Nevertheless, the examples from existing literature shows the prospects of this important area.

3.3 The Design of Metacognition in ILEs

Major instructional and design issues arise when learning systems intend to promote metacognition. The instructional issues are the creation of activities that are sensitive to the context and complexity of the learning task, and that take into account the student’s competence and previous metacognitive behaviours. Also, the definition of metacognitive mechanisms and components involves the careful design of their interface, the language used to talk about metacognitive topics, the timing for metacognitive instruction, etc.

The criteria used to decide what is the most suitable combination of the possible options can vary from domain to domain and depend on the kind of task proposed. But there are nonetheless two basic requirements which the designer should always take into account:

- be careful not to increase the student’s cognitive load;
- get students to recognize the importance of the metacognitive activities.

From the beginning, the design decisions have to match the goals of the metacognitive instruction within the ILE. The metacognition may have a limited role in the system, providing only an underlying frame for developing domain-related skills. In this case it can appear as a single component related to a specific task. It can for instance take the form of an animated assistant who reminds students to use the on-line help. At the other extreme, explicit scaffolding of metacognition may be the main goal of the ILE. In this case, most of the system’s components have to be designed to promote metacognitive development (e.g. a system that teaches students to plan and monitor their study time).

In the next section, we propose a classification schema that contemplates important issues that have to be considered in the design of metacognitive training in ILE.

3.3.1 A classification schema of metacognition in ILEs

We have organized what we consider the major generic issues that have to be taken into account in the design of metacognition instruction in ILEs into a classification schema. This schema can be used to guide decisions regarding both individual metacognitive components to be incorporated into the ILE as well as those relevant to the whole metacognitive module.

As shown in Figure 3.1, the classification schema has three dimensions: timing, target level, and instructional approach. Each dimension is independent from the others so that the option chosen in one dimension does not affect the possible choices in the other dimensions. The possible
values for the timing are not necessarily mutually exclusive. For example, the designer may decide to create a metacognitive activity that aims at fostering the planning of heuristics before and after a problem solving task, both activities targeting heuristics that are domain-dependent (i.e. normally useful for that particular domain) and decide to implement such activity using a collaboration approach.

![Classification Schema for Metacognition Design in ILEs](image)

**Figure 3.1:** Classification Schema for Metacognition Design in ILEs. The diagram shows the independent dimensions and representative values (not mutually exclusive) for each dimension.

### Timing

This dimension refers to the moment chosen for promoting training on metacognitive skills and self-reflection. According to Flavell (1979) metacognitive experiences can occur at any time before, after, or during a cognitive enterprise. Thus, the moment in time where the system provides metacognitive instruction is an important issue to consider. As mentioned before one basic requirement is to not increase the student’s cognitive load. In a learning environment the learner is presented with new information and is asked to perform tasks of different complexity. Balancing the amount and complexity of tasks is important and the instructional designer has to be aware that metacognitive activities are probably new and demanding to students.

Rutz et al. (1999) affirm that questions of a generical and strategic nature are not as sensitive to timing as task specific ones. They developed an animated reflective assistant and tested the trust students had for it as a means to trigger reflection. They remarked that the hardest design issue with respect to trust is timing. Knowing when to interrupt the student, and having a good match between the content of the comment and the current task are key to capturing students’ confidence and consequently making them reflect.

The designer may choose to present the metacognitive activities at one or more of the following points in time:

**Before the learner starts a new learning task:** it can put the learner in the correct frame of mind for the task (before a new problem, before a new lesson, etc.);

**During the learning task:** it can support the student in the self-monitoring process, but the danger is cognitive overload;

**After the completion of the learning task:** it is a natural time to get the student to reflect on her performance and learning process.
**Target Level**

This dimension refers to the way the metacognitive component is embedded in the learning system. Since the same metacognitive skills are used in different contexts, they can be worked on at different levels of integration with the domain of study. Specifically, this classification considers the following levels of instruction:

**General level:** the metacognitive component works on metacognitive skills which are not specific to any domain.

**Domain-related level:** the metacognitive component targets metacognitive skills most useful to the domain e.g. planning reading goals for the domain of learning from texts, or monitoring step-checking if the domain is maths equations.

**Task-specific level:** the metacognitive component focusses on a task the student has to perform (solve an equation, read a narrative text, etc.) making her reflect on the steps she is taking to accomplish that task and on her overall performance.

In the last two cases, there is an additional distinction that can be drawn. In both these cases metacognition is not the only subject of attention of the student given that there is a context which is either a task or the domain. Thus either the student’s attention is drawn on the fact that extra work is being done on metacognition or the work on metacognition takes place but nothing is done to bring this fact to the attention of the student. We call the first case *explicit* metacognitive instruction and the second one *implicit*. Explicit metacognitive instruction creates more load on the student but it has the potential of having a more lasting effect since the student is aware of the work she is doing on her metacognition.

**Instructional approaches of the metacognitive component**

To implement metacognitive activities in computer-based learning environments, several representation mechanisms have been adopted. The most common ones are collaboration (like peer collaboration and menu-driven dialogues), graphical reification, task replay and review (reflective follow-up) and self-reflective activities. The next section explains each one of these approaches and presents examples of existing training systems that incorporate each of those mechanisms.

Computer-based learning environments typically support metacognitive development by providing students either with graphical and representational tools for reflection and interpretations (Reusser, 1993; Schauble et al., 1993), or facilities for prompting self-reflection or help-seeking behaviour (Aleven and Koedinger, 2000). Examples of self-reflective activities in such environments include: reflective prompts or guided questioning that requires learners to justify their ideas and make them explicit (Lin and Lehman, 1999; Goodman et al., 1998), prompts to self-explain (Conati and Vanlehn, 2000), and features that support self-assessment (Schauble et al., 1993; Katz et al., 1998; Niemi and Latva-Karjanmaa, 2001; Katz, 2000). Some ILEs may combine more than one approach to deliver metacognitive training. MIST (Puntambekar and du Boulay, 1997), for example, combines reflective prompts, self-questioning and peer collaboration.

The approaches are detailed in the next section together with illustrative examples of ILEs that incorporate them.
3.4 ILEs that Implement Metacognitive Training

Different combinations of timing, target level and representation approach were used in previous ILEs to define the metacognitive activities. Table 3.1 shows representative examples of these systems organized according to our classification schema. One could say that there are many others ILEs which implicitly support metacognition or where the interaction with the system eventually generates metacognitive development (e.g. using collaborative systems). We only show here those that either explicitly train metacognition, or have metacognition training as part of their project goal.

3.4.1 Graphical reification in ILEs for metacognitive development

Reification is the operation by which something that was previously implicit and unexpressed or even possibly inexpressible (like an abstract concept, an idea, or a series of actions) is explicitly formulated and made available to conceptual manipulation.

Many of the actions a learner performs to solve a problem in an ILE can be reified. For example, imagine that during a problem solving session the student opens a help system twice, she accesses textual explanations about the type of problem, she sends a message to the tutor asking for clarification, etc. These actions are normally implicit to the process and may never be used as a source of study by the learner or the tutor. Applying reification to these actions, using tree or network structures, allows the ILE to present the trace of the learner’s actions. It is a pedagogical device that enables reflection on the problem solving and reasoning process.

The effects of reification could be enhanced with an active coaching strategy and scaffolding. This is a less attempted approach, not least because it is a very difficult one.

Early examples from research on tutoring systems already present graphical reification as a means to improve reflection on the problem-solving process. The Geometry tutor (Anderson et al., 1985) and its second improved version, the ANGLE tutor (Koedinger and Anderson, 1993), and Algebraland (Brown, 1985) are the best known examples of such systems. They provide a meta-level representation of the problem solution (also called “problem spaces” by Collins and Brown (1988)), making it possible for students to interact with the representation and to observe or change the path of the solution.

Later on, Derry and Hawkes (1993) developed TAPS, that besides providing graphical reification of the process of solving algebraic word-problems, incorporates a diagnostic model of errors using fuzzy logic that interprets patterns of errors in problem-solving performance. Another example is the Heron system (also called Heron-WATGRAF) (Reusser, 1993), a learning environment designed to help children in understanding and solving a wide class of complex mathematical word problems. It provides tools for assisting students to comprehend the language of a problem, select the main components that will be used in the solution, and construct an explicit and reified mathematical problem model from which a linear equation can be derived. Apart from the Geometry tutor (and ANGLE), all the others mentioned above are problem-solving environments in the domain of algebra.

It is consensual that through the use of structured visual displays of reasoning processes, monitoring and regulation are reinforced. Nevertheless, none of the systems above incorporated explicit scaffolding on planning, monitoring or regulation. Furthermore, from the examples above, only
## ILEs that include Metacognition Training

<table>
<thead>
<tr>
<th>System</th>
<th>Domain</th>
<th>Metacognition Development Intended</th>
<th>Timing</th>
<th>Target Level</th>
<th>Approach</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Algebraland (Brown, 1985)</td>
<td>Algebra equations</td>
<td>Monitoring Regulation Reflection on problem-solving process</td>
<td>During learning task</td>
<td>Task-specific skills</td>
<td>Graphical Reification</td>
<td>Limited empirical studies: indicated that subjects that acquired error-detection skills using the system also used these skills in problem solving done without the system.</td>
</tr>
<tr>
<td>Sherlock 2 (Katz et al., 1992)</td>
<td>Faults diagnosis in avionics</td>
<td>Regulation Evaluation</td>
<td>After learning task</td>
<td>Domain-related skills</td>
<td>Reflective follow-up activities</td>
<td>Simulated environment: learning by doing. Experiments show technicians improved troubleshooting skills and transferred them to other tasks.</td>
</tr>
<tr>
<td>TAPS (Derry and Hawkes, 1993)</td>
<td>Algebra word-problems</td>
<td>Planning Monitoring Error-recovery skills</td>
<td>During learning task</td>
<td>Task-specific skills</td>
<td>Graphical Reification</td>
<td>Uses Fuzzy Logic to match students’ solution to systems’ set of possible solutions. Planning is facilitated with menu of problem “schemas”. No experimental studies reported.</td>
</tr>
<tr>
<td>Heron (Reussner, 1993)</td>
<td>Complex algebra word-problems</td>
<td>Self-directed problem comprehension and conceptualization Planning</td>
<td>During learning task</td>
<td>Task-specific skills</td>
<td>Graphical Reification</td>
<td>Support problem understanding with explanatory help on text vocabulary. Strong use of the mouse to perform tasks. Uses cognitive task analysis (instead of student modelling) to perform behavioural diagnosis.</td>
</tr>
<tr>
<td>System</td>
<td>Domain</td>
<td>Metacognition Development Intended</td>
<td>Timing</td>
<td>Target Level</td>
<td>Approach</td>
<td>Observations</td>
</tr>
<tr>
<td>---------------------------------</td>
<td>-----------------</td>
<td>--------------------------------------------------------------------------</td>
<td>-------------------------------</td>
<td>----------------------</td>
<td>---------------------------------</td>
<td>-------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>HyperClinic (Gama et al., 1997)</td>
<td>Cardiology</td>
<td>Reflection on prior knowledge Planning</td>
<td>Before learning task</td>
<td>Domain-related skills</td>
<td>Peer collaboration Reflective prompts</td>
<td>Case-based system. Used in real settings. No statistical analysis done.</td>
</tr>
<tr>
<td>LuCy (Goodman et al., 1998)</td>
<td>Satellite activity</td>
<td>Reflect on knowledge Articulate knowledge Review</td>
<td>After learning task</td>
<td>Task-specific skills</td>
<td>Collaboration: Learning Companion (LC)</td>
<td>The LC acts as an advisor and has expert knowledge. LC asks questions in natural language.</td>
</tr>
<tr>
<td>The Pedagogical Assistant (Rutz et al., 1999)</td>
<td>Object-oriented analysis</td>
<td>Monitoring Regulation</td>
<td>During learning task</td>
<td>Task-specific skills</td>
<td>Collaboration: Learning Companion (LC)</td>
<td>Animated LC in the format of a parrot Study conducted to check the degree of “trust” on the LC</td>
</tr>
</tbody>
</table>
### ILEs that include Metacognition Training

<table>
<thead>
<tr>
<th>System</th>
<th>Domain</th>
<th>Metacognition Development Intended</th>
<th>Timing</th>
<th>Target Level</th>
<th>Approach</th>
<th>Observations</th>
</tr>
</thead>
<tbody>
<tr>
<td>IQ Form (Niemi and Latva-Karjanmaa, 2001)</td>
<td>Domain-independent</td>
<td>Awareness of self-as-learner</td>
<td>Before and during learning task</td>
<td>General skills</td>
<td>Self-assessment Personnal diaries</td>
<td>The tool creates students’ profiles. Based on previous self-assessment questionnaires. Tested with 4 different subject groups from distinct domain areas from distance higher education courses.</td>
</tr>
<tr>
<td>Embedded Assessment System (Katz et al., 2001)</td>
<td>Hypothesis testing in statistics</td>
<td>Knowledge monitoring</td>
<td>During and After learning task</td>
<td>Task-specific skills</td>
<td>Self-reflection Self-assessment</td>
<td>Used to create a automatic assessment of metacognitive behaviours in ILEs. Tested experimentally (but with few subjects). Uses the Knowledge Monitoring Assessment as a comparison measure test.</td>
</tr>
<tr>
<td>ECOLAB II (Luckin and Hammerton, 2002)</td>
<td>Science: food chains and webs</td>
<td>Help-seeking awareness task selection skills</td>
<td>During learning task</td>
<td>General skills</td>
<td>Metacognitive prompts and messages</td>
<td>Scaffolds children to select optimally useful help Creates a metacognitive ability tag about the user (i.e., a simple metacognitive model)</td>
</tr>
<tr>
<td>Basic Inquiry Platform (Woolf et al., 2002)</td>
<td>Geology, biology, civil engineering, etc.</td>
<td>Inquiry Skills</td>
<td>During learning task</td>
<td>General Skills</td>
<td>Questioning Inquiry learning Cases</td>
<td>Complex software with different tools, phases and resources Domain independent infrastructure Main goal is to develop inquiry skills and self-critical learners</td>
</tr>
</tbody>
</table>

Table 3.1: Examples of ILEs that support metacognition.
the Geometry Tutor and Angle were used in real classrooms settings, with high-schools students.

All these systems use reification for slightly different purposes. In *Algebraland* the student is presented with an algebraic equation and a choice of basic operations she can perform to simplify the equation. The system structures the audit trail of the search space explored by the student as she applies the operations offered to her. So the student does not interact directly with the reification; instead she uses it as a visual display of her path towards the solution, as it reveals dead ends, and the multiplicity of correct solutions paths. The student can see exactly where she backed up, where she reached the same state twice, where she was getting farther away from a solution, and so on. Thus, the student is encouraged to generate multiple problem-solving trees. Also, the structured representation of partial solution paths provides an opportunity to reflect on problem-solving and evaluation strategies in the context of use (Collins and Brown, 1988). Studies conducted with the system investigated whether students who reviewed their problem-solving traces acquired general skills for avoiding errors or for detecting errors and recovering from them (Foss, 1987). However, the empirical studies were limited and indicated that the desired effect hardly ever occurred (Foss, 1987). But whenever the students acquired the error-detection skills it was positively associated with subsequent successful use of those skills in problem solving done without the system.

The *Geometry Tutor* provides a representation of the concept of proof. Using the reification it is shown that a proof is a path between axioms and theorem, and certain actions are shown to be fruitless because their results lie on incomplete paths. Students have to use both forward and backwards inferences to connect the premisses and the conclusions of the proof. This graphical visualisation of geometry proofs emphasizes the overall process of completing the proof rather than the final product. This points to Brown (1985) who says that by placing the emphasis on the entire thought process, reification favours deeper forms of learning rooted in reflexive recapitulations of mental processes. Later on Koedinger and Anderson (1993) designed the *ANGLE* tutor (*A New Geometry Learning Environment*), a second generation of the Geometry Tutor. They intended to further address the issue of “implicit planning”, i.e. the thinking process behind competent performance that is under the surface and not revealed in the actions performed, nor made explicit in textbooks or typical instruction. So, they created interface notations to allow students to make a plan of the proof, not worrying about the details of the proof (e.g. icons for representing generic schema categories). After a proof plan has been discovered the student could fill in the details, adding rules and statements to the solution.

The *TAPS* system (Teaching Arithmetic Problem Solving) is an ITS system, which was designed to help students solve arithmetic word problems (Derry and Hawkes, 1993). Its major cognitive tool is a graphical interface that facilitates the construction of problems trees. The student chooses from a menu of blank subtree diagrams (called “schemas”) and fills the blank nodes of the tree with labels, values and operators. An interesting and innovative feature is the use of fuzzy logic to diagnose errors and patterns of errors, enabling the system to carry out on-line performance monitoring and error recognition. For this purpose the system has a knowledge base with possible sub-trees for each problem. However, TAPS does not use any measure of student’s progress for choosing future activities. Although the error recognition could trigger local tutoring decisions, a tutoring module was not implemented. In sum, the system has great potential to encourage students to engage in planning and self-monitoring, but experimental studies need to be
Heron goes beyond the problem space reification and combines it with tools for understanding the description of complex algebra word-problems. Two kinds of strategy support for text comprehension are proposed: (1) explanatory help with respect to the vocabulary, syntaxes and semantics of the problem, and (2) aid with identification and conceptualization of problem components together with support to build the solution tree (or planning tree, as he calls it). This graphical reification in the format of a solution tree bridges the gap between the text of the problem and the linear equation that has to be produced. Heron was explicitly designed as a non-directive cognitive tool with high level of learner control.

An empirical study with four pairs of students was conducted and the subjects’ dialogues were videotaped (Reusser et al., 1996). An analysis of the dialogues was done focusing on Heron’s assistance on the qualitative understanding of the problem situation. The results provide evidence of quantitative and qualitative changes in the conversation, related to both the communicative and the task-related aspect of peer collaboration. The results also clearly show that for some students there is a need for more help or control by either the system or a participating teacher or expert. As a consequence of the results of this initial analysis, a new version of Heron was developed including a feedback component that provides some degree of help. Unfortunately, we found no indication that another, more extensive, and conclusive evaluation was conducted.

3.4.2 Reflection through follow-up activities

Using the recorded process trace, or audit trail to encourage reflection on problem-solving strategies is another useful approach. This kind of post-task reflection is called reflective follow-up (Katz et al., 1992) and provide students with capabilities for seeing replays or reviewing their performance and actions. The difficulty of this approach is to design an appropriate abstraction of the audit trail. The Sherlock 2 tutor (Katz et al., 1998) is a good example of a tutoring system that uses reflective follow-up activities.

Sherlock 2 is one of a series of systems developed as tools for training avionics technicians in diagnosing faults and repairing faulty aircraft and the systems used to maintain them. It is a realistic computer simulation of the actual job environment. Besides training in the domain, it aims to develop general diagnostic skills that can be readily transferred to related job specialties. Sherlock 2 presents trainees with a series of exercises of increasing difficulty. There are two main phases of a problem in the system: problem solving and review. The review is called reflective follow-up (RFU); it is a post-problem reflective phase where students reflect on their problem-solving performance and receive feedback from Sherlock 2. More specifically, during the RFU, students can step through their solution and receive feedback from the computer coach, replay the simulated expert’s solution, review the instructional goals of the tutor, ask questions about their own or the simulated expert’s actions, and get suggestions about how to improve in the next session.

The basic idea of Sherlock’s coaching scheme is to give students control over their own learning and to help them develop metacognitive skills by requiring them to figure out for themselves what type of information they need. Likewise, the reflective follow-up activity promotes learning by helping the student see another viewpoint on solving the problem grounded in the context of
the student’s own solution. This approach has been demonstrated to be effective on the hardest tasks.

Studies showed that learners who used Sherlock 2 improved dramatically in their troubleshooting skills, during training as well as on a post-test (Gott et al., 1996).

### 3.4.3 Metacognition through collaboration

Based on the Vygotskian concept that understanding one’s own learning is enhanced by interaction with others during learning, some systems have adopted collaboration as a basis to promote reflection (Vygotsky, 1978).

Systems, like LuCy (Goodman et al., 1998) and the Animated Pedagogical Assistant prototype (Rutz et al., 1999), use peer collaboration in the form of learning companions. Goodman et al. (1998) justify this approach stating that reflection and articulation are two key pedagogical functions underlying student-learning companion dialogues. A learning companion can help a student reflect on her thinking by critiquing, questioning, or evaluating particular steps. Similarly, students and learning companions may articulate those steps through further explanation or elaboration.

Other systems combine reflective prompts with collaboration, triggering students’ dialogues through system’s questioning. MIST (Puntambekar and du Boulay, 1997, 1999) for example uses menu-driven system-student dialogues with students working in pairs. MIST encourages students to think about what they have been doing and what they want to do using an active strategy of questioning.

The HyperClinic system (Gama et al., 1997) is another example of a collaborative environment with reflective prompts. It is a web-based case-based application in the domain of cardiology that proposes reflective questions to a group of students on a specific clinical case as a preparation to solving the case. HyperClinic provides a newsgroup tool to promote discussion about clinical knowledge relevant to the case and possible strategies to elaborate an appropriate diagnosis. The main goal of the discussion is to activate students’ prior knowledge and help on articulating their thinking. Trials showed that the participation of senior physicians in the discussion, offering tips and advices, helped students to concentrate on the hard aspects of the case and articulate their thinking better. One interesting result was that, even being an informal discussion, some (not so novice) students felt embarrassed expressing their confusion or lack of knowledge in the discussion list that included the experts. So we removed student identification which generated an increased participation by the students.

**MIST** (Metacognition in Studying from Texts) is designed to help students develop a systematic approach to learning from texts by supporting a range of planning and monitoring activities. It helps students become aware of the processes they might engage in when learning from texts. The emphasis is on the process and not on the product of learning. Thus, the interface presents opportunities to discuss and reflect on the different types of activities students might engage in when learning from texts rather than on the specific text they are learning. Therefore, the system addresses the metacognitive skills relevant for the domain, focusing on the students’ repertoire of strategies and how they will use them during the study session. MIST deals with the major goals of learning from texts, which are: work out the core information contained in the texts, reduce it to its main ideas, comprehend it, and integrate it with the learners’ existing prior knowledge.
The first part of the system helps students become aware of the various strategies that can be used as well as where, when, and how they should be used; it also teaches students to pay attention to the different types of tasks and texts and activate any prior knowledge that they may have (Puntambekar and du Boulay, 1999, p. 249). In the second part, students can choose options relating to planning, text processing or reading, and memory enhancing. Because MIST has knowledge about the types of activities that students might engage in during learning from texts, rather than specific knowledge about the texts that students wish to study from, intervention by the system is necessarily at the meta level and is meant to help students think about what they are doing and why they are doing it. Students use MIST in pairs and the system’s questions and activities trigger a collaborative reflection on the metacognitive skills used when learning from texts. MIST is one of the few examples of systems whose primary objective is to train metacognition.

LuCy is a simulated learning companion, acting as a peer in an intelligent tutoring system, called PROPA. LuCy was developed to encourage the student to reflect on and articulate her past actions, and to discuss her future intentions, like alternative courses of action, and their consequences. LuCy has an inquiry interface. Goodman et al. (1998) state that peers encourage each other to reflect on what they are learning and to articulate their thinking, which enhances the learning process. So LuCy aims to promote more effective instructional exchanges between students and the intelligent tutoring system. By initiating a menu-driven dialogue with a student, the system forces reflection and articulation on the student’s learning.

3.4.4 Self-explanations in ILEs triggering metacognition

Self-explanations encourage the student to reflect on and articulate her past actions (Chi et al., 1989). However, as already mentioned in the previous chapter, studies show that students do not spontaneously engage in self-explanation and need guidance to do it (Bielaczyc et al., 1995). One issue is how to motivate students in an ILE to do so and how the self-explanations can be used for tutoring purposes.

Conati and Vanlehn (2000) developed the SE-Coach, a system that support students learning from examples through self-explanations. The system acts within Andes, a system that tutors on Newtonian mechanics. The SE-Coach includes an interface to scaffold students’ self-explanations, a probabilistic student model and a coaching component. The student model integrates information on the students actions, the model of correct self-explanation and the students domain knowledge and thus enables the assessment of example understanding from reading and self-explanation actions. Based on these evaluations the SE-Coach changes the level of scaffolding by eliciting further self-explanations whenever necessary. There are three different and incremental levels of self-explanation scaffolding.

The SE-Coach interface uses a clever masking mechanism that requires the student to explicitly move the mouse over the part of the example she wants to read to uncover it. This makes it possible for the system to know which parts of the example the student has seen and for how long. Unfortunately, the system does not allow self-explanations in natural language (not least because it would be a very complex task). Instead, the student builds a sentence through a series of choices in lists of options. The student is not verbalizing using her own words thus not guaranteeing that true self-explanation is occurring.
Another limitation is that the system only provides guidance to self-explanations that are local to a certain solution step, asking students to justify the step based on inferences of previous knowledge demonstrated. It is not possible then to ask students to explain steps that are missing, which is common in experts’ solutions.

A pre-post test experiment was conducted with 56 college students who were taking introductory physics classes. The results suggest that in the experimental condition (N=29) (i.e. students using a fully functioning version of SE-Coach) students at an early learning stage were the ones who benefited best from the structured scaffolding. The explanation was that because they were still unfamiliar with the subject matter, they are more motivated to put substantial effort in exploiting the help at best. They did not find similar results with the more proficient students. Although very interesting and promising, the SE-Coach needs more conclusive indications of its benefits for developing students’ self-explanations and monitoring of comprehension.

3.4.5 Self-assessment in ILEs

A more recent attempt to develop an environment focused on metacognition is the IQ FORM (Intelligent Questionnaire Platform). It is part of a research project conducted at the University of Helsinki by an interdisciplinary research group (Niemi and Latva-Karjanmaa, 2001; Virtanen et al., 2003). Differently from all the others examples in this chapter, this learning environment is not attached to any specific domain and, therefore, targets metacognition development for learning in general.

The IQ FORM is a set of interactive Web-based learning tools for promoting higher education learners’ self-regulation, learning skills and strategies, and supporting collaborative processes in virtual courses for distance education. The tool set has two parts: (1) a tool for assessing a learner’s individual qualities and learning skills (The IQ Learn) and (2) a tool that offers information about group processes to promote students participatory skills, collaborative learning and knowledge creation in e-learning (The IQ Team). Both parts consist of interactive questionnaires for self-assessment and tutorial packages that provide information and concrete advice about how one can develop metacognitive skills and behaviour. The system also includes a learning diary where the learner can record, for example, the ideas inspired by the assignments of the tutorial package. In addition, the system offers the teachers a special tutoring package with information and advice about how to make the learning easier and more meaningful for learners studying on a Web course.

One of the tools provides reference information about the study group and each student may compare her learning skills to those of other students in the group.

The The IQ Learn tool consists of three elements:

1. An interactive test bank with three questionnaire sets for students’ self-evaluation (Figure 3.2) and graphic tests result:

   - **forethought of learning**: expectations of success, performance anxiety, meaning of studies, self-efficacy, and self-confidence.
   - **strategies in learning**: time management, self-management, persistency, and help-seeking strategies.
   - **learning skills**: rehearsal, critical thinking, finding essential points, connecting new and old knowledge, keywords and advanced organizers, application of theories, and self-assessment.
2. The tutoring sets, with a hypertext structure for each sub-component of the tests. It has texts directed to the learner and the teacher.

3. A learning diary for the reflection on profiles and learning experiences.

![Image of IQ Form System](image)

**Figure 3.2:** Snapshot of IQ Form SystemExample of the IQ Learn interactive self-questionnaire.

The IQ-Team tool is still under development and consists of:

1. An interactive test bank with three questionnaire sets:
   
   **Group Roles:** rejection, dominance, encouraging, conforming, sharing know how, avoidance.
   
   **Social Interdependence:** individualistic, competitive, cooperative.
   
   **Group Processes and Knowledge Creating Process:** atmosphere, goal orientation and commitment, innovation and creativity, the benefits of doing together, the utilization of differences, the role of the tutor.

2. The tutoring sets.

3. A joint learning diary called “the Log book” for reflections, discussion and knowledge creation.

The IQ Learn component of the IQ Form was used and evaluated in a pilot study with 4 different groups of higher education students from different subject areas. Students were asked to use it freely, according to their own interests, while studying on an on-line distance course and to write in their learning diaries how they used the IQ Learn questionnaires and tutoring tool. At the
end of the course, they were asked to compare their comments in their diaries with their IQ Learn results.

This pilot study revealed that some students benefit from the virtual tutoring, mainly those who are at an early stage of their studies, or have difficulties in learning. For example, some younger students on one subject group reported that they felt that the IQ Learn had helped them to find the reason for their lack in learning (e.g. lack of time management and confidence). The concrete issues of self-regulation such as time-management and tips to cope with test anxiety were considered as very useful by the subjects. Students with stable and effective learning strategies reported that the system strengthened their understanding of their learning. However, many students only used the tool superficially, and only gained most from the system when they were guided in to use it, by giving clear assignments.

As general as it is, the IQ Form resembles the Cognitive Strategy Instruction (CSI) programs (mentioned in Section 2.5) and faces the danger of becoming abstract and disconnected from real learning experiences. Students may acquire new information about themselves as a learner and possible remedial strategies but may not detect when and how to use them. Moreover the tutorial texts are static and do not adapt to students’ needs or courses’ needs. So, the tips and orientations may not all be relevant to the learning task at hand.

Nevertheless, the IQ Form may have interesting uses, if it is tailored to specific instructional goals of a given course. The questionnaires tackle many important issues, but perhaps too many issues for a student to explore in a single course. Selecting sub-components of the questionnaires, adapting the tutoring texts to content areas, and combining them with the course exercises could provide a powerful environment for metacognitive development. Also, it is necessary to develop tools for monitoring students’ progress. It would be interesting for the student to gain awareness of which activities suggested provided best learning improvements when used and which ones did not help them.

Our guess is that IQ-Team tool, when developed and tested, may prove to be more effective than the IQ-Learn, because in distance learning courses students need a good framework for group interactions.

Next, we present a system called “Embedded Assessment System”. It is an interesting example of an attempt to create measures for assessing metacognition in an ILE.

### 3.5 Designing automatic measures of metacognition

Katz et al. (2001) have developed a web-based prototype called **Embedded Assessment System** for problem solving in the domain of statistics. The topic chosen was hypothesis testing because it was identified as particularly challenging for students to learn and for instructors to teach.

The prototype system is part of a larger research project that aims at designing automatic models for adjusting metacognitive scaffolding in computer-based learning environments. As such, this prototype aims at detecting types of metacognitive behaviour students engage while solving problems in order to identify measures that have general applicability and the potential to be incorporated into computerized learning environments.

The prototype is structured as a scaffolded learning environment for problem solving. Figures 3.3, 3.4 and 3.5 show snapshots of the prototype. For each problem proposed the student
is expected to perform four types of activities: problem categorization, construction of problem solution, reflection, and self-assessment. The components and activities are summarised below.

![Embedded Metacognitive Assessment Prototype- snapshot 1. Start new problem: problem categorization.](image)

**Figure 3.3:** Embedded Metacognitive Assessment Prototype- snapshot 1. Start new problem: problem categorization.

The problem solving activity is organized as a sequence of goals that need to be accomplished to solve the hypothesis-testing problem. So, the solution to the problem is obtained by answering separate questions. Students have to fill 6 sub-solutions, starting with categorizing the problem, and justifying their choice as shown in Figure 3.3. Then, the student has to establish the hypotheses, select a statistical test, justify the statistical test, perform the hypothesis test, and finally draw conclusions - Figure 3.4.

Next to each guided question there is a reflective prompt (e.g. “What information led you to these hypotheses? Why did you choose a two-sided solution?”) as shown in Figure 3.4. Students are expected to answer the goals questions and the prompting questions. The purpose of embedding these prompts is to encourage students to reflect on and explain their problem solving as they work towards a solution, i.e. generate explanations for each solution given. As the answering order is not controlled by the system and it is possible for students to respond to the solution steps and reflective prompts in the desired order.

The prototype incorporates a help system. For each question students have to answer there are two kinds of help: guiding questions, that aim at focusing learners on important parts of the problem; and textbook, which provides more detailed information on how to accomplish the goal.

After solving the problem, students have to assess their own solution. For that purpose, the student’s solution is presented (both solution steps and explanations) together with an expert’s solution (both solution steps and explanations). Students are requested to judge whether their solution is equivalent or not to the expert’s; if not, learners are prompted to compare their solutions and explanations with those of experts, providing additional opportunities for self-explanation and remediation - Figure 3.5.

An experiment was performed using twelve students who had just completed the yearlong AP Statistics class. The study was conducted across three sessions and all students participated in the
All participants worked on the same problem, which was of medium-to-low difficulty. Some students attempted additional problems, but because these extra problems differed among the students, the experiment analysis focuses on student performance only on the first problem.

One interesting aspect of this study are the measures created to analyse metacognitive behaviour. Metacognition was measured in this experiment through students’ interaction with the metacognitive prompts, help system, and self-assessment. The level and quality of student engagement in the proposed activities in the prototype were indicative of their knowledge monitoring behaviour. All nine measures are summarised in Table 3.3 below. As an external validity check for the measures they assessed students’ knowledge monitoring skill in statistics independent of the learning environment. For this purpose they used the Knowledge Monitoring Assessment instrument created by Tobias and Everson (as defined in Section 2.4.2) adapted to statistics problem solving (they called it StatKMA). The intent was to determine the extent to which any of the metacognitive online measures matched the metacognitive skill as assessed by the StatKMA.

The interaction with the prototype was recorded and the log was used to detect patterns of problem solving, omission of responses to solution steps, metacognitive prompts, self-assessment activities, editing behaviour students might have engaged in and their use of help.

In their analysis Katz et al. (2001) found a strong relation between SAT-M and Knowledge
Figure 3.5: Embedded Metacognitive Assessment Prototype - snapshot 3. Self-assessment.

Table 3.2: Experimental Design of the Embedded Assessment System: sessions’ organization.

Monitoring Assessment scores thus confirming results from previous research (Tobias and Everson, 2002). Interestingly the Knowledge Monitoring Assessment score did not correlate with the problem score. Moreover, contrary to their expectations, they found little relation between the knowledge monitoring assessment measured with the StatKMA and the online metacognitive measures. However, the relation between reflective prompts scores and the Knowledge Monitoring Assessment were stronger than those between the Knowledge Monitoring Assessment and the self-assessment scores. This result suggests that scoring responses to the reflective prompts may be a better reflection of knowledge monitoring skill compared with responses to the self-assessment task.

The online metacognitive scores were more closely associated with overall proficiency; i.e. the students who knew statistics well (looking at Statistics Score and Problem Score) tended to do well on the reflective prompts.
Table 3.3: Measures for the Embedded Assessment System: metacognitive and domain-related measures.

Hence, of the two hypotheses put forward none was sustained by the experiment.

The first hypothesis was that the explicit presentation of steps and sub-goals of the problem would prompt students to monitor and recheck earlier solution steps. Actually, from the 12 students, 11 followed the structured sequence imposed by the system, first elaborating a response for a solution step, then generating the reflective explanation and then moving on to the next step. Only six students engaged in editing behaviours, backtracking to earlier steps to make changes to their responses. This could suggest that these students engaged in reflective thinking, where the changes in their responses were as a result of responding to the reflective prompts. Unfortunately, this conclusion is not confirmed by their statistical analysis. In order to draw a conclusion it would be necessary in our opinion to ask these students to perform a later experiment where they had to solve another similar problem without the reflective prompts and observe their behaviour. If they did not replicate their editing behaviour, then the claim that the reflective prompts were the causal factors for the change of pattern would be strengthened.
Chapter 3. Metacognition in Interactive Learning Environments

The other hypothesis put forward was based on the idea that the use of help is indicative of a student’s knowledge monitoring ability and strategic action to bridge the knowledge gap. It was hypothesized that students with high metacognitive awareness would be sensitive to their lack of knowledge and so would seek help more consistently than their less aware counterparts. Surprisingly, they observed that students rarely used help even in situations when they were not accurately completing the solution. From the twelve participants, only five students accessed the help system, typically after entering a solution step but before responding to the corresponding reflective prompt. Thus, the reflective prompts did not trigger further help seeking. Instead, students used help to check their responses to a problem solving step. The researchers suggest that one explanation for that is that during the experiment students collaborated with each other as they worked on the system.

The lack of significant results was partially be attributed to the small sample size (N=12).

One criticism that can be made to this study is that, besides the attempt of creating interesting measures for metacognition some designs decisions corroborate making the claims weak:

- the analysis was made based on only one problem which is not enough to observe metacognitive patterns of behaviour;
- students were not able to practice with the system before attempting their first problem;
- problems were not tailored to be challenging for each student, making the help and reflections less necessary; A Medium-low difficulty problem does not provide the trigger for metacognitive skills to take over.
- prompts were always visible preventing the researchers to observe when and if students look for further aid for understanding - similarly to what happened with the help system;
- participants should have been instructed to work individually on the problem.
- the StatKMA presented questions with content somewhat different than that specifically assessed by the prototype; also it was a multiple-choice test, whereas the system asked for written answers. Moreover, students did not replicate prediction actions in the system. How to compare this measure with the other measures in the system taking that the Knowledge Monitoring Assessment measures specifically the knowledge monitoring skill?

In conclusion, the inclusion of metacognitive prompts to elicit explanations is insufficient if the students are unaware of the expectations of the task. Moreover, the mere presence of the prompts is not enough to improve learning unless students are motivated to use them. Additional modelling might be necessary to facilitate the generation of appropriate explanations.

3.6 Conclusion

In this chapter we have presented a review of research on metacognition in ILEs, which is one of the contributions of this thesis. The review used a classification schema for the major issues concerning the design of metacognition instruction. Some representative examples of interactive learning environments that incorporate one or more components dedicated to metacognition instruction were offered, together with the description of experiments conducted with some of these ILEs that shade light to our own research.
As we have seen, only two of these have been developed with the specific goal of teaching metacognition: MIST and IQ Form. In all the other systems, researchers have sought improvements in performance in the domain or on cognitive skills and metacognitive training is only a means to that end. Only those two explicitly help students to learn about the process involved in learning and the role of metacognition on this process. On the other hand, the Embedded Assessment System tries to do something different from most others: to create online measures of metacognition from students’ behaviours. Unfortunately, it did not have much success.

Of all the systems reviewed, only SE-Coach uses what can be called a student model to direct the behaviour of the system with respect to metacognitive instruction; however, only the self-explanation ability is modelled. We believe that although it not an easy task, this is an effort that must be emulated. Such a model must be built taking as input any aspect of the student-system interaction from which information on the metacognitive level of the student can be extracted. Then, using the model, it becomes possible to adjust the quantity, timing, and content of the metacognitive training according to student’s needs.

Another challenge for ILEs is the design of adequate external representation to represent the “invisible” metacognitive skills the learner possess, such as monitoring and regulatory behaviours during learning tasks. The SE-Coach devised a clever way of modifying the user interface to detect which parts of the problem the student is looking at. Similar novel mechanisms to reify other subtle behaviours need to be designed.

In the next chapter we present the Reflective Assistant Model. It is a framework for explicitly fostering metacognitive skills that suggests specific timing, target level and approach metacognitive scaffolding in problem-solving learning environments for maths and includes a rule-based algorithm to assess students’ metacognitive state.