A Framework to Specify a Cognitive Diagnosis Component in ILEs

Josephine Tchetagni, Roger Nkambou, Jacqueline Bourdeau

Jl. of Interactive Learning Research (2006) **17***(3)*, 269-293

A Framework to Specify a Cognitive Diagnosis Component in ILEs

JOSEPHINE TCHÉTAGNI
University of Quebec at Montreal, Canada
tchetagn@info.uqam.ca

ROGER NKAMBOU
University of Quebec at Montreal, Canada
nkambou.roger@uqam.ca

JACQUELINE BOURDEAU
LICEF Research Center-Tele-University, Canada
jacqueline.bourdeau@licef.teluq.uqam.ca

This article presents a framework for the cognitive diagnosis of learners' errors in an interactive learning activity occurring in an intelligent learning environment. The proposed framework supports the implementation of an authoring tool. This tool helps instructional designers to specify the features of a component for cognitive diagnosis. Two key issues are addressed. First, the cognitive diagnosis process should promote reflection which may in turn enhance the accuracy of the diagnosis hypotheses: (1) reflection leads learners to contemplate their own understanding of the cognitive skills required by a learning situation; (2) when the pedagogical context is devoid of the physical presence of human tutors, reflection also allows for a more reliable representation of the learner's needs. Second, the pedagogical context in which the learner is diagnosed must be taken into consideration: the parameters of this context affect what is observed and diagnosed during the learning experience. First, this view of cognitive diagnosis is formalized into a framework. Second, the article presents the main functions of CD-SPECIES, a framework-based authoring tool which supports the specification of a cognitive diagnosis component in intelligent learning environments.

Introduction

So far, many of the research issues in the area of instructional design of intelligent learning environments (ILE) have focused on the instructional process. These issues include the setting of instructional and learning goals, the curriculum planning (Nkambou, Frasson, & Gauthier, 1997), the organization of instructional and learning activities and the standardization of learning resources (Paquette, Bourdeau, Henri, Basque, Leonard, & Maina, 2003). Another important issue concerns the design of intelligent assistance provided to learners using these systems. Intelligent assistance in an ILE is based on two main technologies: intelligent tutoring and adaptive hypermedia (Brusilovsky & Peylo, 2003). Intelligent tutoring is particularly useful in problem solving situations and includes intelligent curriculum sequencing (Weber & Brusilovsky, 2001), intelligent solution analyzers (Heift & Nicholson, 2001) and interactive problem solving support (Melis, Andrès, Büdenbender, Frishauf, Goguadse, Libbrecht, et al., 2001). These techniques can only be effective when used in conjunction with an accurate cognitive diagnosis (CD) of learners' errors which enables the system to provide relevant and helpful feedback to the learner.

The goal of a CD component in an ILE is to determine *accurately* why the student made an error in a problem solving activity in order *to identify where to focus* the subsequent learning activities. The CD process interprets the learner's errors on the basis of two main factors: the *hypotheses* of the system pertaining to the learner's current knowledge and the *pedagogical context* in which the learner is diagnosed. Two main issues related to these factors emerge from state-of-the-art research about CD in ILEs.

The first issue questions how the phenomenon of reflection could be enabled, while hypotheses about the learner are established in a CD process. Establishing hypotheses about the learner's cognitive state is at the core of CD in ILEs, since these hypotheses guide the pedagogical orientations of the system. In this respect, research has mainly been focused on the *operational aspect of CD*, that is the application of artificial intelligence (AI) techniques that allow the system to accurately infer the learner's current level of knowledge (Anderson, Corbett, Koedinger, & Pelletier, 1995; Conati, Gertner, & VanLehn, 2002; Katz, Lesgold, Hughes, Peters, Eggan, Gordin, et al., 1998). More recently, it was acknowledged that there is a need to go beyond these technical considerations. The pedagogical aspect of the CD process should be explicitly included in its implementation in an ILE (Dimitrova, Self, & Brna, 2000). One perspective of the pedagogical aspect of CD focuses on: (1) how to foster a remedial state upon the system's hypotheses about the learner's cognitive state *while* performing the CD and (2) how to continually adjust the system's hypotheses regarding the learner's cognitive state in order to obtain more accurate representations of the learner's needs. These two questions embrace the phenomenon of reflection. The first question is related to fostering reflection about the knowledge elements that were diagnosed as not acquired or which remain problematic or misconceived. This question is very relevant in this context given Dewey's statement (1933) according to whom reflection is a reliable means for real learning. The second question is related to adjusting the hypotheses inferred by the system concerning the learner's cognitive state on the basis of a feedback loop from interactions with the learner. In this case, CD in an ILE should be implemented in such a way that learners' reflections allow the system to refine, confirm or infirm the hypotheses about what is learned and what is not. The expressions "system belief" and "system reflection" are widely used in the field of artificial intelligence and education (AIED) when referring to ILE (Self, 1994). The term *system belief* will be used to refer to the hypotheses inferred by the system about a learner's cognitive state. The term *system reflection* or *reflection from the ILE* will be used to refer to the process of adjusting these hypotheses during the CD process in an ILE.

The second issue concerns the explicit account of the pedagogical context in which a CD process evolves. The pedagogical context is defined by the learning goal and by the way the learning activities are designed. In the particular case of an ILE, one approach to capture that context is to consider the paradigm of cognition, the learning theory and to a lesser extent, the instructional and instructional design (ID) theories which govern that activity. Indeed, a cognition paradigm proposes a view of the nature of knowledge which inspires learning theories from which, in turn, instructional and ID theories are elaborated. Learning theories propose a view of learning goals as well as a definition of the learning processes which lead to these goals. Instructional and ID theories propose principles to design learning activities intended to lead to some specific learning goals. Therefore, it is reasonable to consider that these three parameters should be taken into consideration in order to understand the pedagogical context of a learning activity. While the relevance of these elements has been acknowledged in the design of ILEs learning activities (Mizoguchi & Bourdeau, 2000; Reigeluth, 1999), they have not been explicitly considered in the design or the implementation of CD in ILEs. Indeed, the CD infers what is learned and what needs to be learned. Thus, it must consider the nature of what is to be inferred (cognition paradigm), the processes upon which the inferences are based (learning theory) and the entities and the sequence of events of the situation in which the inferences are made (instructional and ID theories).

As stated above, most of the research on CD in ILE addresses the use of AI techniques to provide intelligent assistance to these systems. Enabling the aforementioned pedagogical aspects of CD will foster a more integrated approach to the CD process in ILE. The goal of this study is to allow CD design and implementation practices in ILEs which take into account both of these issues. In this respect, this article has two objectives. The first objective is to present a framework which defines the key elements of the CD process in ILE. The content of this framework is proposed in order to address both issues. The framework is designed to support the construction of a tool to help instructional designers to specify a CD component for an ILE. The second objective of this article is to present how such a tool could be designed and implemented in order to assist Instructional Designers in the specification of the CD process in an ILE.

This article is organized into four sections. The first section reviews state-of-the-art research regarding CD specifications in an ILE. The second section presents the framework: how could *reflection* and *pedagogical context* be integrated into a basic CD framework in an ILE? The third section describes an implementation of the framework into CD-SPECIES, a tool for CD specifications: how could this tool assist instructional designers in their tasks? The fourth section addresses the benefits of the proposed approach.

RELATED WORKS

Generally, computer tools that help educators and instructional designers in the building process of an ILE focus on producing a tutoring component (Blessing, 2003; Towne, 2003), on modeling tutoring actions and strategies (Ainsworth, Major, Grimshaw, Hayes, Underwood, Williams, et al., 2003; Murray, 2003; Van Marcke, 1998) and on instructional planning (Nkambou, Frasson, & Gauthier, 2003). Tools such as GTE (Van Marcke, 1998) and REDEEM (Ainsworth et al., 2003) develop a specific aspect of the CD process. Generally, they do not take into account the pedagogical context in which learners are observed and, as explained below, the CD pedagogical context allows for an explicit and consistent account of data that can be manipulated by the instructional designer. The IRIS authoring tool (Arruarte, Ferrero, FernÃndez-Castro, Urretavizcaya, Ãlvarez, & Greer, 2003) allows for the automatic construction of a CD component in an ILE. However, it is specific to a learning task and only addresses the CD algorithmic perspective without considering the pedagogical aspect. The approach adopted by Shute & Torreano (2003) is similar to the one presented in this article: a tool is proposed to gather information about how the cognitive analysis of a learning task should evolve. This approach differs from the one proposed in this article in that the corresponding tool is not intended to assist in the design of a a pedagogical function like the CD of the learners, actions.

A Framework to Specify a Cognitive Diagnosis Component in ILE

In this section, a general framework for the CD process is outlined, at first, on the basis of an analysis of prior research on the subject and then, the aspect of *reflection* is integrated into this framework. An illustration of a CD instance specified from that framework is illustrated in PROLOG-TUTOR (PT), an ILE for Logic Programming (Tchetagni, Nkambou, & Bourdeau, 2005). Finally, the aspect of the pedagogical context is integrated to obtain the final framework.

The Fundamental Components of Cognitive Diagnosis

The conceptual components of the CD process were defined according to Wenger's analysis (1987). Depending on the diagnosis of the learner's cognitive state, Wenger suggests three CD perspectives: epistemic, behavioral and individual. Epistemic diagnosis deals with the students' knowledge of the field of study as well as their strategic and metacognitive skills. Behavioral diagnosis refers to learners' behavior (reasoning) during a problem solving activity. Individual diagnosis concerns issues such as the learner's personality, preferences, learning attitudes, and so forth. Moreover, according to Mislevy (1994), the CD process requires the execution of the same operations as in medical diagnosis. Thus, the procedural components of CD could be defined as the phases of observation, interpretation and inference of hypotheses pertaining to the learner's cognitive state. The observation phase consists of collecting data that are relevant to the kind of inferences that the CD process intends to make. In order to be useful, the CD process should establish relevant distinctions in order to retain such observations. The interpretation phase consists of rationalizing observations: the CD process should construct a problem solving story or a coherent sequence of events which explains the observations. In its simplest form, the interpretation of an observation could be seen as an evaluation of the learner's input. The terms *interpretation* and *evaluation* are used interchangeably in the remainder of this article. Finally, the inference phase consists of formulating hypotheses concerning the learner, which explains the interpretation of the observation (Figure 1).

Designing Learners' CD in ILE: Integrating the Reflection in the CD Process

When considering an instance of CD processing, an ILE first observes and evaluates the learner in a problem solving situation. Secondly, depending on the results of the evaluation, the ILE triggers the inference process that will generate the diagnosis hypotheses. The system makes inferences by using two main sets of data sources: a learner model and a task model which is a structure that defines the relationship between the problem and the elements of the domain under study. The form and the content of these models rely upon the conceptual nature of the CD process. Epistemic CD uses a representation of the learner's knowledge based on formalisms to represent declarative knowledge. Behavioral CD uses a representation of the learner's reasoning processes, based on production rule systems for example. The system infers a set of hypotheses to explain the student's incorrect answer. How could reflection be integrated into such a process? *After each inference*, the ILE can direct its

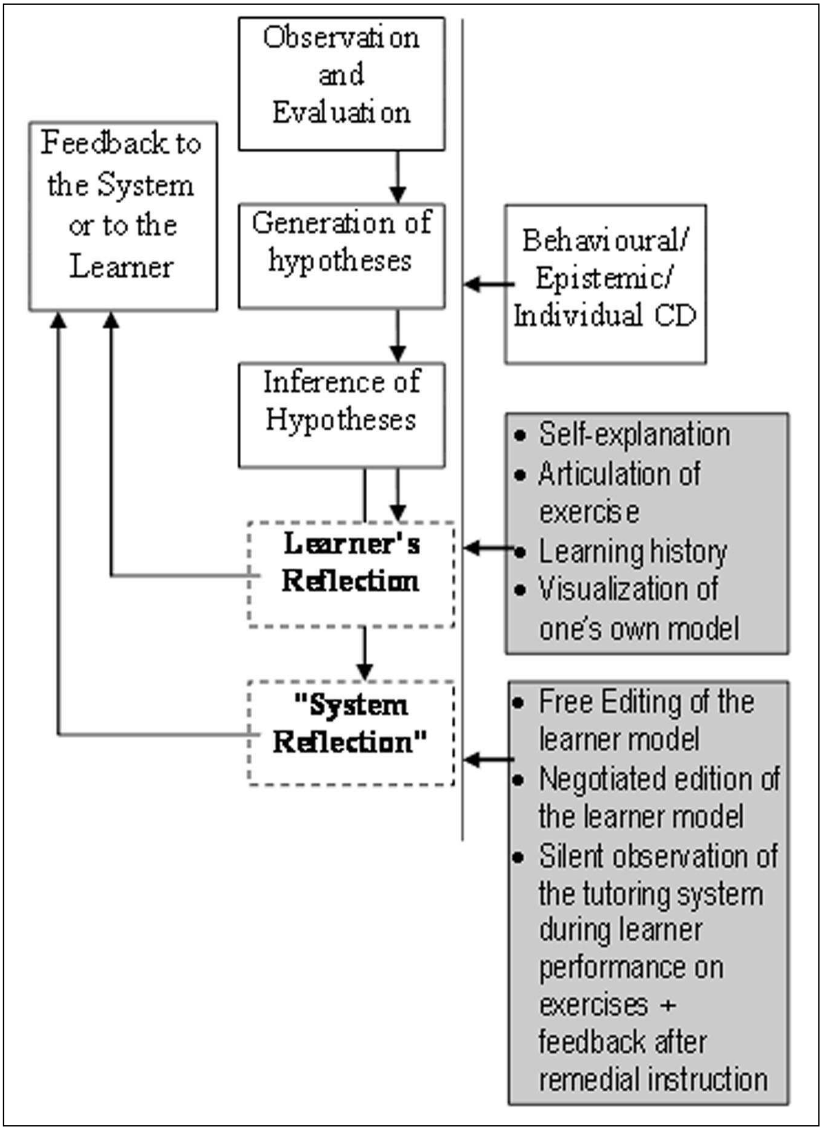

Figure 1. Integrating reflection into a basic framework for cognitive diagnosis

interactions with the learner to trigger reflection about the knowledge elements or the skills corresponding to the diagnosed hypotheses (learner's reflection). The ILE can *then use the result of the learner's reflection* in order to confirm, infirm or refine the diagnosed hypotheses (ILE reflection).

Learners' Reflection

From the learners' perspective, the goal of reflection is to make them reflect upon their own knowledge, skills and misconceptions. Learners' reflection is achieved by fostering insights pertaining to certain concepts and principles. Methods such as self-explanations and explanations to peers or tutors are used in this respect (Figure 1). For example, self-explanation (Chi, De Leeuv, Chiu, & Lavancher, 1994) engages learners in a reflexive process where they must justify the answer provided in an exercise. Tutoring strategies such as the use of hints, feedback, explanations and articulation could be adapted to trigger reflection. For example, in the articulation method described by Tchetagni, Nkambou and Kabanza (2004), learners are asked questions after they erroneously solved a problem in Logic Programming. These questions are related to the relevant features of the problem that support the reasoning towards a solution. It is expected that these questions will compel the learner to think about these strategies in relation to these features.

Learners' reflection may also be achieved by encouraging learners to consider and evaluate their own progress with respect to each learning goal defined in a course. Learners can be given the opportunity to visualize and edit their own model in order to change certain values and to compare their model to those of other learners who are using the system.

System Reflection

As for the system that underlies the ILE pedagogical component, reflection should be viewed as a means to better understand the learner's needs based on data directly provided by him. This is achieved through the practice of interactive open-learner modeling (or interactive cognitive diagnosis). The ILE gets information through explicit interactions with the learner or through implicit inference.

Overt reflection within the system occurs when learners explicitly indicate to the system the components that they believe they have mastered and those which they have not. In this case, two modes of system-learner interactions were identified:

(1) Non-negotiated editing and model modification: In this case, learners can freely edit their model and change the data concerning their cognitive state in the system's beliefs. Most often, this change concerns an upgrade or a downgrade of the learner's level of mastery with respect to certain learning goals. This is useful when the focus of the intelligent assistance is related to adaptive navigational support. However, when the focus is on performance, learners could have to justify the changes they wish to bring to their models.

(2) Negotiated editing: In this case, learners must justify all modifications they wish to make to their model. Two main actions allow

them to do this. Learners provide a trace of the learning activities which triggered the model modification. Learners may also perform an exercise related to the learning goals where the mastery level is either upgraded or downgraded.

Implicit system reflection occurs when it silently observes the learner in order to better understand where to focus assistance. Implicit system reflection uses methods that range from macro-level tools (such as recording learning traces) to micro-level actions (such as interactive tutoring dialogues with the learner).

Illustration: Integrating Reflection in the CD Process through a Tutoring Dialogue

In order to illustrate how reflection can be integrated into the CD component of an ILE, an example of the implementation of CD for PT was designed. PT is an ILE designed to teach the paradigm of logic programming with the Prolog programming language. In the following, the Courier font is used to refer to Prolog syntax and terminology. PT considers that a skill integrates two components: a capability (such as those in Bloom's taxonomy) and a knowledge element in the domain of logic programming. This formulation is used to make learning goals explicit since the combination of these two elements results in the formation of a learning goal (Tchetagni et al., 2004). While this method may not be relevant in a stand-alone ILE, it becomes important when an ILE is built from a principled instructional design specification. In this case, one has to ensure the correspondence between the inferences that will be made by the CD in such an ILE and the learning goals defined in the instructional design phase.

Figures 2 and 3 respectively illustrate the conceptual model of this example and the corresponding PT implementation. The basis of the CD is a process of remedial instruction achieved through a method called *articulation of salient features of the skills required*. In this process, the tutoring system asks the learner questions after a failed exercise. Each question is related to a salient feature of a skill that is required to perform the exercise. A salient feature is a feature that captures the key reasoning steps (and the corresponding skills) leading towards the solution of a given problem. Salient features are predefined after a problem is analyzed.

From the learner's perspective, articulating the salient features of a problem triggers reflection. Indeed, bringing out these features compels the learner to become aware of the key notions associated with the learning goal of the exercise (interactive remedial instruction). This leads learners to reflect upon their own understanding of the domain with respect to these notions. If the learner fails an articulation step, the ILE diagnoses all of the skills which are associated with that step (coarse-grained CD based on interactive remedial instruction). It then generates an exercise that specifically elicits the skills

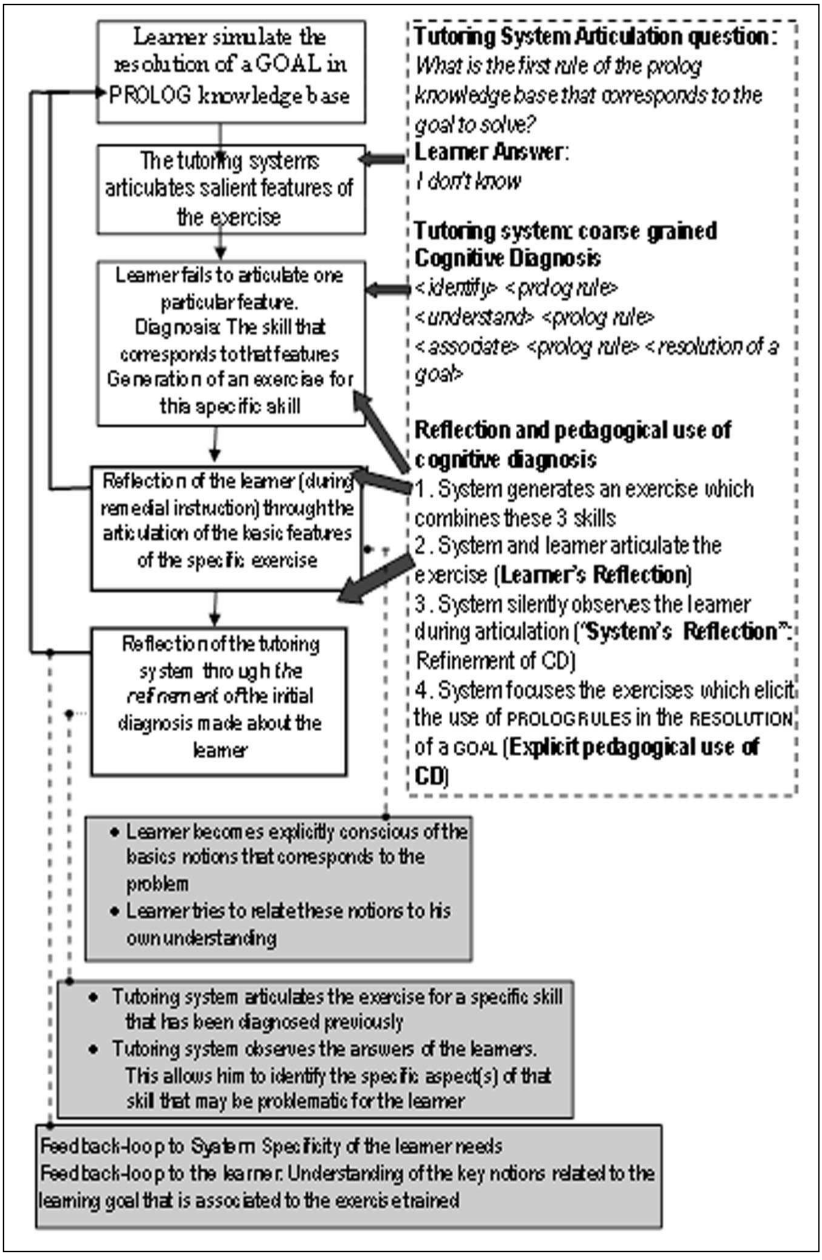

Figure 2. Integrating reflection into cognitive diagnosis in Prolog-Tutor: A model

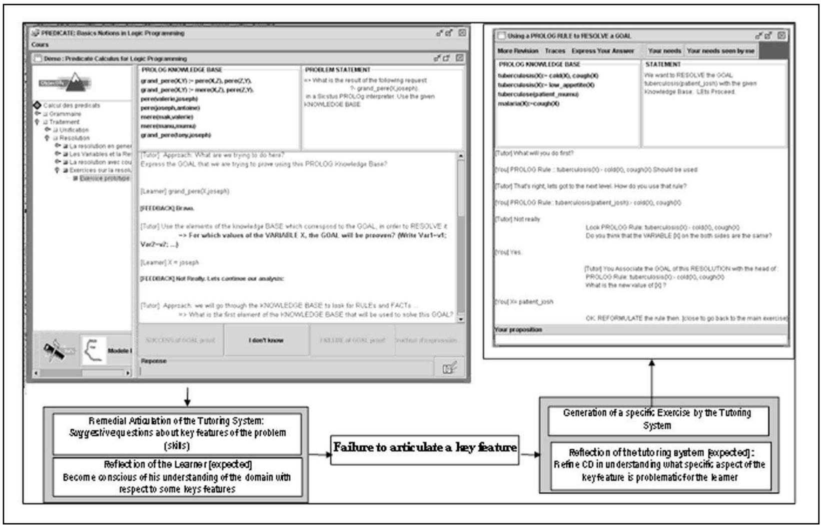

Figure 3. Integrating reflection into cognitive diagnosis in Prolog-Tutor: An implementation

that correspond to the featured question. For example in Table 1, all of the skills diagnosed are related to the knowledge element "prolog rule."

From the perspective of the pedagogical module that underlies PT, this articulation allows the system to *refine the diagnosis hypotheses* that were made initially about the learner. In this way, the ILE can identify the specific aspect of the knowledge element that is problematic for the learner's understanding (fine-grained CD based on interactive remedial instruction). This is an implicit reflection that will allow the ILE to delineate the relatively fine degree of the learner's problems. This information could be used in several manners in an ILE:

(1) It could be reported to the learner model for future use by the remote human tutor;

(2) It could also be used immediately to generate problems which specifically elicit that skill;

(3) It could be used to support adaptive navigation or support the sequencing of learning activities.

Table 1 illustrates an application of an instance of CD with reflection. In Logic Programming, students can be asked to simulate how the process of resolution of a goal would evolve in a Prolog interpreter. The learner's goal is to solve a goal using a Prolog knowledge base. In this case, the learner cannot provide the correct answer. Therefore, the system starts to articulate the salient features of the exercise with the learner. For example, this case includes

Table 1
The Resolution of a Goal in Prolog-Tutor

Actor & Reflection	Feedback Loop
ILE [Articulation: Question 1] What is the first prolog rule of the knowledge base which could be used to solve: grand_father(X,joseph)?	NO
Learner: [Answer] "I don't know."	YES: **Expected Reflection** - What is the role of a rule in performing a Resolution? - What is the link between a prolog rule and the goal of a resolution?
ILE: [CD] **Observation & Evaluation:** Learner fails **Coarse-Grained Diagnosis Hypotheses:** <*Identify*> <Prolog Rule> (Skill1) <*Understand*><Prolog Rule> (Skill2) <*Associate*> <Prolog Rule and a Goal to solve > (Skill3) **Refinement of Diagnosis** ("System Reflection") - ILE generates an exercise which requires the elicitation of the three specific skills - Learner fails to elicit Skill1: [Fine-grained diagnosis]	YES: **ILE focuses on the "learning goals"** - The system infers what specific aspect of the knowledge element: Prolog-rule should be focused on. The learner needs to master how to *Associate* a Prolog Rule with the "Goal" to solve

three salient features: (1) What is the link between a Prolog rule and the goal of a Resolution? (2) When is there a need to perform backtracking while solving a goal? and (3) What happens after backtracking?

For example, if the learner fails after the first articulation, the system diagnoses that the knowledge element: Prolog-rule is problematic. However, this element includes several aspects: *identify* a Prolog-rule, *understand* the meaning of a Prolog-rule and *associate* a Prolog-rule with a goal to solve. In order to refine this diagnosis, the system generates a specific exercise. This specific exercise focuses on making the learner solve a goal using a simple Prolog knowledge base which contains exactly one Prolog-rule and a set of Prolog facts. The specific exercise should be simple as the system objective is to use it to determine whether the learner's difficulties arise from: the *identification* of a Prolog-rule, the *understanding* of its meaning or its *association* with the goal.

Designing Learners' CD in ILE: Explicit Consideration of the Pedagogical Context

So far, an implementation of a CD component based on the elements of the proposed CD framework was presented. However, this task is not performed by an educator, but rather, by a programmer. Thus, the programmer needs the specifications of the content expected from an ILE CD component, from an educator or an instructional designer. How can the gap between these two interveners be filled? Indeed, the concern is to transform the tutoring purposes that constitute the pedagogical context where the CD will occur into an explicit specification for the programmer or for any other interested intervener. The goal is to end up with a computational system whose behavior actually corresponds to the authentic pedagogical method intended by an instructional designer/educator. For the particular case of CD, the pedagogical context should indicate: (1) what is diagnosed, (2) what is observed and interpreted to this end and (3) the moment when the CD process happens. As mentioned in the introduction, the pedagogical context relies upon three main elements: a cognition paradigm, a learning theory and some instruction and instructional design theories.

Cognitive Diagnosis and Cognition Paradigms: What to Diagnose? What to Reflect Upon?

The cognition paradigm describes the nature of knowledge, henceforth the nature of what will be diagnosed. According to Greeno, Collins and Resnick (1996), three main paradigms are the most influential in educational psychology: the empiricism with the behaviorist learning theories, rationalism with the cognitivist learning theories and pragmatism (coupled with the socio-historic perspective of knowledge) with the socio-constructivist learning theories. Considering the assumptions and principles of these paradigms, the learners' CD can be interpreted as follows:

(1) From an empiricist/behaviorist perspective: learners' CD consists of *establishing* the correspondence between their behavior and an expected behavior in the presence of a specific set of features and clues. From this perspective, the reflection process is mainly applied by the ILE and not required by the learner. In this sense, the ILE reflection consists of determining which desirable behaviors have yet to be acquired and what information should be presented in order to acquire these behaviors.

(2) From a rationalist/cognitivist perspective: learners' CD consists of *establishing* their understanding of concepts and principles in terms of appropriate mental representation. Learners' CD also helps to establish their meta-cognitive skills such as problem solving strategies. Learners' reflection consists of reasoning about their mental representation. From the ILE's perspective, reflection consists of establishing the accuracy of its beliefs regarding the learner's mental state.

(3) From a pragmatist/socio-constructivist perspective: learners' CD consists of *establishing* their inquiry skills, their meta-cognitive skills (planning, reflection, analysis and self-criticism) and their collaborative work skills. Overall, learners' reflections consist of reviewing

their inquiry process, reflecting upon their interactions with other actors in the learning environment and the reviewing their learning methods (reviewing meta-cognitive skills). The system reflection consists of determining which actions would be most relevant in order to support learners in the process of constructing knowledge. The answer to this question pertains to the tools provided for learners in their environment. These tools include the inquiry instruments, the resources that support the inquiry and the planning of the learning process (meta-cognitive skill) as well as the resources which facilitate collaboration.

Cognitive Diagnosis and Learning Theories: What Must Be Observed?

Based upon a paradigm of the nature of knowledge, the learning theory defines which mental and physical behaviors lead to the acquisition of that knowledge. For that reason, the learning theory that underlies the implementation of a learning activity should be considered when defining a CD process that evolves during the activity: it indicates what learning processes are relevant to observe and the types of inferences which are best suited for it. However, the relevant observations that support the learners' CD will differ from one learning theory to another since the general problem solving behavior differs from one theory to another. For example, one can ask what is relevant to observe in a pragmatist/socio-constructivist pedagogical context: the performance, the inquiry behavior or the reasoning strategy required to achieve a goal? The learning theory also affects the kinds of reflection processes that will be promoted by an ILE (Table 2). In a theory of inquiry learning for example, reflection upon the way data is collected and interpreted (inquiry skills and inductive skills) is more relevant while in a cognitivist learning theory, reflection upon one's own mental process is prevalent.

Instructional and Instructional Design theories: When to Perform CD?

Based on learning theories, instructional and ID theories define the organization of the learning experience in order to reach the learning goals. These theo-

Table 2
Reflection Operations Observed in a CD Process

Learning Theory	Type of Reflections
Cognitivist Learning Theories	⇒ Learners reflect upon their problem solving and meta-cognitive skill ⇒ System reflects upon the way performance tests are presented ⇒ System reflects upon the way content is presented
Inquiry Theories	⇒ Learners reflect upon their inductive inquiry skills ⇒ System reflects upon the way data is presented to learners in order to lead them to make inferences

ries do not directly influence the CD process, although they must allow the instructional designer to indicate when the CD process should intervene in the learning experience. Therefore, consistency remains an issue since some of these theories do not include learning activities within which a reliable CD can be executed. For example, in Table 3, the Motivational Design Theory provides principles to enhance learners' motivation and improve learning and performance. Thus, it may be inconsistent to try to apply the learners' CDs in that context since there are no learning activities to assess the learners' cognitive states.

In the process of developing explicit CD specifications by an instructional designer, the challenge is to obtain a specification that is consistent with the associated pedagogical context. This shows the importance of a design methodology that is *aware of* the cognition paradigm, of the learning theories and of the instructional and ID theories upon which the learning activity of a CD evolves. In order to achieve that consistency, the relationship between all three parameters of the pedagogical context and the CD process should be explicitly stated as summarized in Table 4.

The following section presents CD-SPECIES: a tool intended to support an educator or an instructional designer in order to specify the relevant CD components of an ILE. This tool is based on the framework described in this section. The relevance of CD-SPECIES as a computer tool is that it allows instructional designers to *interactively* articulate their specifications so that it appears as clear as possible to a programmer or any other interveners concerned.

Towards a Computer Tool to Support the Specification of a Cognitive Diagnosis Component

The framework is intended to support the implementation of a tool designed to help instructional designers to specify a CD component in an ILE: CD-

Table 3

Consistency Between CD and Instructional and ID Theories (Reigeluth 1999)

Instructional Theory	Learning Activities
Gagne Nine Events	Event 6: Elicit Performance includes Elicit Learners' Activities, Facilitate Learners' Elaboration
Theory of Teaching Inquiry	Inquiry problems with discrepancies
Elaboration Theory of Instruction	A learning activity may be associated with any level of the course structure prescribed by the theory
ARCS Model of Motivational Design	A learning activity may be introduced to allow for meaningful success, thus increasing confidence
Open-Learning Environments	A learning activity may be introduced anywhere in the open learning environment
Designing Constructivist Learning Environments	Learning occurs through the presentation and the resolution of a problem in a ill-structured domain, thus a learning activity

Table 4
Accounting for the Pedagogical Context in a CD Framework

CD Process	Cognition Paradigm	Learning Theories	Instructional and ID Theories
Observation [Epistemic, Behavioral]	No significant influence	What relevant elements must be inferred?	1. When and where to apply the CD process within an ILE? 2. Who is going to be the object of CD?
Interpretation [Epistemic, Behavioral]	No significant influence	How can observations be interpreted?	
Inference [Epistemic, Behavioral]	**Epistemic Diagnosis:** What type of knowledge elements will be recognized as acquired or not acquired?	**Behavioral Diagnosis:** What reasoning processes will be diagnosed as evolving?	

Table 5
The Three Main Functions of CD-SPECIES

	Manage Knowledge Base	Assist ID in CD Specifications	Inform Educational Software Programmers
Description	Manage a knowledge base of: declarative knowledge about CDs with a set of associated rules and a database of CD specifications and objects	Assist the ID to specify the CD. Assistance is provided during the specification process.	Inform the educational software programmer about the computational implications related to a given specification
CD-SPECIES Interface Functions for Instructional` Designers	- Select a Paradigm - Select Learning Theories - Select Instructional Design Theories - Specify the CD Components - Specify the CD Process - Provide examples of CD specifications/instances in a pedagogical context	- Provide information about the CD given a pedagogical context - Check consistency, relevance of CD components values given a pedagogical context function - Generate specification objects given a specification and save it in the knowledge base	- Generate pseudo algorithms given a specification - Generate a skeletal architecture for the CD module of an ILE given a specification - Recommend an AI technique given a specification

SPECIES. This section presents how such a tool should be designed and implemented. CD-SPECIES has three main functions (Table 5): (1) it comprises and manages a knowledge base whose contents are used to assist the instructional designer when specifying a CD process instance. This knowledge base is constructed mainly from the framework, (2) it assists an instructional designer or an educational psychologist in the specification of a CD process instance and (3) it bridges the gap between the instructional designer/educational psychologist and the educational software programmer.

In the following section, a discussion is proposed in order to situate the use of a tool such as CD-SPECIES in the instructional design process. Then, the aforementioned functions are described.

Situating CD-SPECIES in the Instructional Design Process

According to Morrison, Ross and Kemp (2004), the instructional design process represents the *systematic development of instructional specifications* using learning and instructional theories to ensure quality instruction. It starts with the analysis of learning needs and goals and it evolves with the development of a delivery system to meet those needs. It includes the development of instructional materials and activities as well as the validation and evaluation of all instructional and learning activities. *Instructional design refers to the science of creating detailed specifications* to develop, implement, assess and maintain situations which facilitate the acquisition of large and small units of subject matter at all levels of complexity. In order to meet the learning objectives, most instructional design theories include specific learning activities. These activities are the means which permit observing the achievement of the learning goals. According to the view proposed in this article, the CD is the means by which the observations scanned in a learning activity are interpreted and *pedagogically exploited* with respect to the learning objectives. Therefore, a tool such as CD-SPECIES may be solicited in the phase of specifying a learning activity in the instructional design process. However, these learning activities should be such that they elicit an observable learning/performance behavior.

For example, ID Expert (IDE) is an authoring tool based on the instructional design theory known as the Instructional Transaction Theory (Merrill & ID2-Research-Group, 1998). IDE approaches the instructional design through units of transactions: a set of interactions sequences between the learner and an ILE in order to reach a *specific* learning goal. An instructional transaction supports different kinds of instructional interactions including presentation, exploration, practice, and assessment. CD-SPECIES could be used prior to the specification of a transaction by an IDE expert. In this case, CD-SPECIES would support the specification of the CD process associated with any transaction. This specification can then be implemented as a complete transaction (each phase of the CD process is an instructional interaction) or as a part of a transaction (the whole CD process is implemented as an instructional transaction).

From the Framework to the Knowledge Base of CD-Species

In order to assist instructional designers, educational psychologists and educational software programmers in tasks related to the implementation of the CD component for an ILE, CD-SPECIES must have an appropriate knowledge base. Two main components were identified to form such a knowledge base: (1) a knowledge base which defines declarative knowledge about the CD, this base is drawn directly from the framework described in the first section of the article; (2) a database of CD process patterns in a pedagogical context, each pattern being associated with metadata about the properties of the corresponding CD specifications (For example: the conceptual CD type, the type of hypotheses inferred according to the considered cognition paradigm).

The knowledge base contains a vocabulary list and a set of rules. The vocabulary represents the possible values of each element of the CD, as specified in the generic framework (Table 6).

The set of rules defines some constraints amongst the values of the CD components within a given pedagogical context: the intra-rules. In some cases, the rules also define some constraints amongst the values of the CD components across pedagogical contexts: the inter-rules (Table 7). In Table 7 for example, the first inter-rule tells that the system will allow the designer to specify that he wants to observe the performance or the capacity of association of the learner – which are typical of an empiricist/behaviorist context – even in a pragmatist context of learning, provided that the goal of CD is to determine the learner's change or evolution of understanding with respect to some knowledge element.

The database of CD process patterns contains a set of skeletons (or specifications) of CD scenarios. Since the metadata are associated with a specifica-

Table 6
Making the CD Elements Explicit in a Pragmatic View of Learning

<table>
<tr><th colspan="2">CD_SPECIES_Knowledge_Base.Vocabulary.Pragmatist.InquiryLearning</th></tr>
<tr><th>CD ELEMENT</th><th>VOCABULARY</th></tr>
<tr><td>Observation and Evaluation</td><td>- GOAL_OF_OBSERVATION:
<skill_inquiry>| <skill_drawConclusion> |
<skill_changeOfUnderstanding>|<skill_planning>,
<skill_integrateCultureIntoLearning>|
<skill_knowledgeConstructionFromInteraction>
- TYPE_OF_OBSERVATION
<skill_inquiry>
<skill_informationCollection>
[number_of_hypothesis_made]
[number_conclusion_drawed]
[number_consistent_conclusion_drawed]</td></tr>
<tr><td>Generation and Inference of Hypotheses</td><td>- METHOD FOR INFERENCE
<Relate_Resources_Inquired>
<Infer_Inquiry_Method>
<Human_Analysis>{Explanation_of_Drawed_Concept}</td></tr>
<tr><td>Reflection</td><td><Tool_To_Relate_Activities_In_Inquiry_Process>
<Tool_To_Trace_Interactions_Frequency></td></tr>
</table>

Table 7
An Illustration of the Rules of the CD-SPECIES Knowledge Base

CD_SPECIES_Knowledge_Base.Rules
INTRA_RULES
1. IF <type_of_observation> is {performance} AND <learning_context> contains {pragmatist:socio-constructivist} ⇒ inconsistency_message("In Socio-constructivist contexts, learners actions and interactions are more relevant for Cognitive Diagnosis than performance")
2. IF <learning_context> contains {pragmatist:socio-constructivist} AND IF <goal_of_observation> is {skill_knowledgeConstructInterActions} AND IF <type_of_observation> is {skill_informationCollection} ⇒ irrelevance_message("Information collection will generally happen prior to the interaction with peers. It is more appropriated in the individual inquiry process")
INTER_RULES{Pragmatist, Empiricist}
1. IF <learning_context> contains {pragmatist:*:*} AND IF <goal_of_observation> is { skill_changeOfUnderstanding } ⇒ accept_values(<Observation> in {Association, Performance})

tion, it could be searched for, according to a given criterion: for example, a user could search for CD specifications for an epistemic CD process (the search criterion in this example is the conceptual characteristic of the CD process).

An Overview of the CD Design Process with CD-SPECIES

CD-SPECIES methodology for the specification of the CD process is similar to the IMS learning design methodology (Figure 4). An instructional designer *uses* the tool to *produce a specification* of the diagnosis process according to the pedagogical context in which it will evolve. The tool supports the instructional designer in order to ensure two qualities of the design process: the designer's explicit articulation of the CD components in the ILE and the comprehensiveness of the knowledge base used for the specification. Indeed, CD-SPECIES allows the instructional designer to become conscious of the CD components process in an ILE and of the relationships amongst them. This is relevant as sometimes, the elements of instructional expertise are implicit in the mind of the instructional designer. When it is necessary to use this expertise or to make it operational in a context, the instructional designer may have difficulties articulating these elements. CD-SPECIES is intended to help instructional designers to formulate their CD knowledge, by presenting a skeleton of the features of that knowledge. CD-SPECIES allows for an evolution towards a more and more exhaustive framework for CD since as a computer tool, it can be continuously updated, adding new CD components and features.

The only way for the instructional designer to take advantage of these properties is through an appropriate interface in CD-SPECIES. The functions offered in that interface allow for:

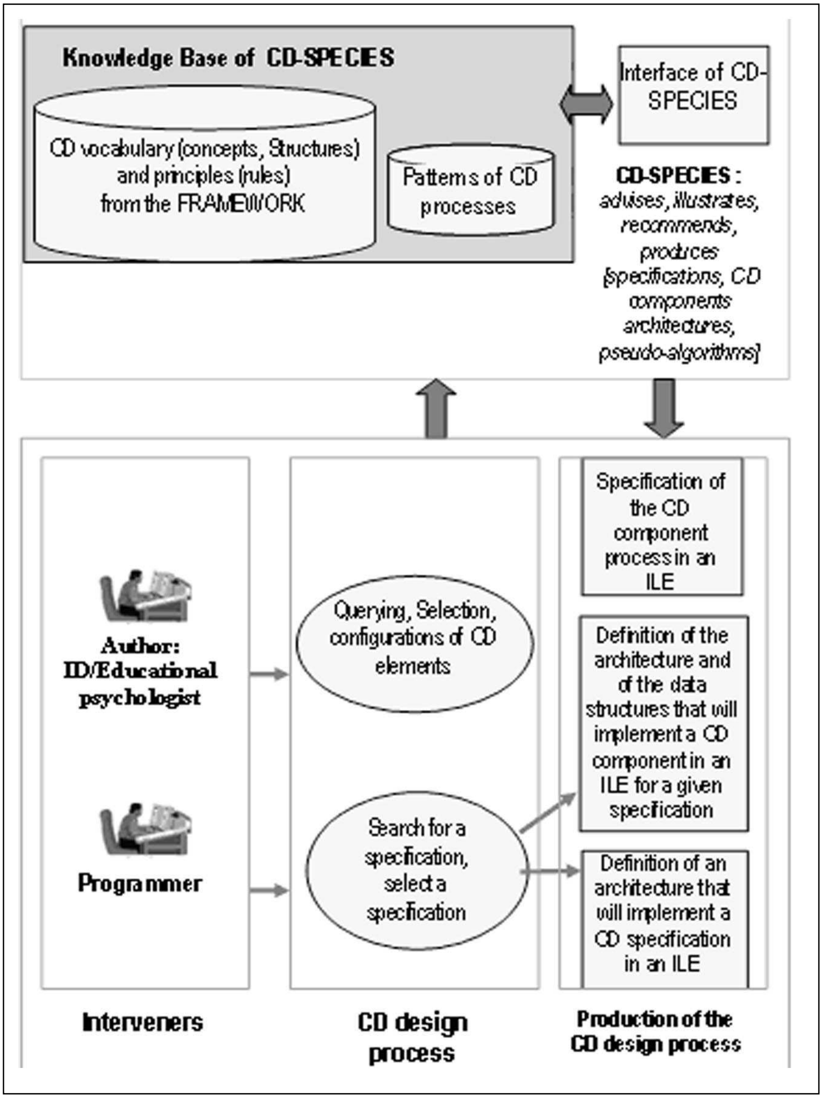

Figure 4. Conceptual model of CD-SPECIES

(1) The definition of the pedagogical context of the CD process;

(2) The definition of the corresponding values for the CD process components (Figure 5);

(3) The request to visualize a specification or an instance of specification.

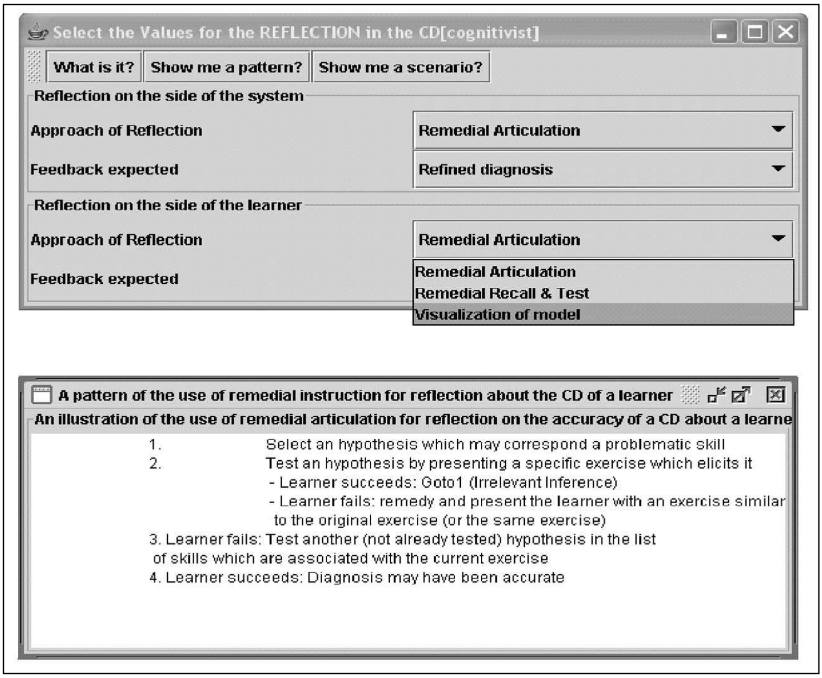

Figure 5. Use of the authoring tool by the designer

How does CD-SPECIES Assist the Instructional Designer?

Besides providing a knowledge base for the content of a specification, CD-SPECIES also assists the instructional designer. This assistance consists in three main functions (Table 5): (1) it provides the ID with information concerning the nature of a CD component, (2) it checks the consistency of a specification while it is being defined and (3) it generates specifications.

While constructing specifications, the instructional designer can request information regarding the nature of a CD process component. In this case, CD-SPECIES provides a general definition of the corresponding component. If the request is made after a pedagogical context has already been defined, CD-SPECIES may provide a component description with respect to the pedagogical context (Table 2, Table 3). For example, CD-SPECIES can provide a description of the appropriate types of learners' reflection for a specific context with a pragmatist view of the nature of knowledge.

Specifications attribute values to the CD process components in a specific pedagogical context. These values must be consistent with the pedagogical context. For example, the first Intra-Rule in Table 7 shows that it is inconsistent to observe the performance of the learner in a pedagogical context defined by a pragmatist view of cognition and a socio-constructivist view of learning. The function of consistency is based on the CD rules in the CD-SPECIES knowledge base. These rules allow the verification of consistent values selected by the designer for each CD component. If there is a constraint on these values, then the system sends a message to the designer (Figure 6).

When a specification is sufficiently well defined, CD-SPECIES generates a s*pecification object*. A specification object can take two forms: a raw text file or a formatted text file that can be read by another program. Moreover, to allow for subsequent referencing, a specification should be associated with a set of metadata so that it could be referenced for future searches if need be.

How CD-Species Informs the Educational Software Programmer

A specification produced by CD-SPECIES can be further processed in the following ways:

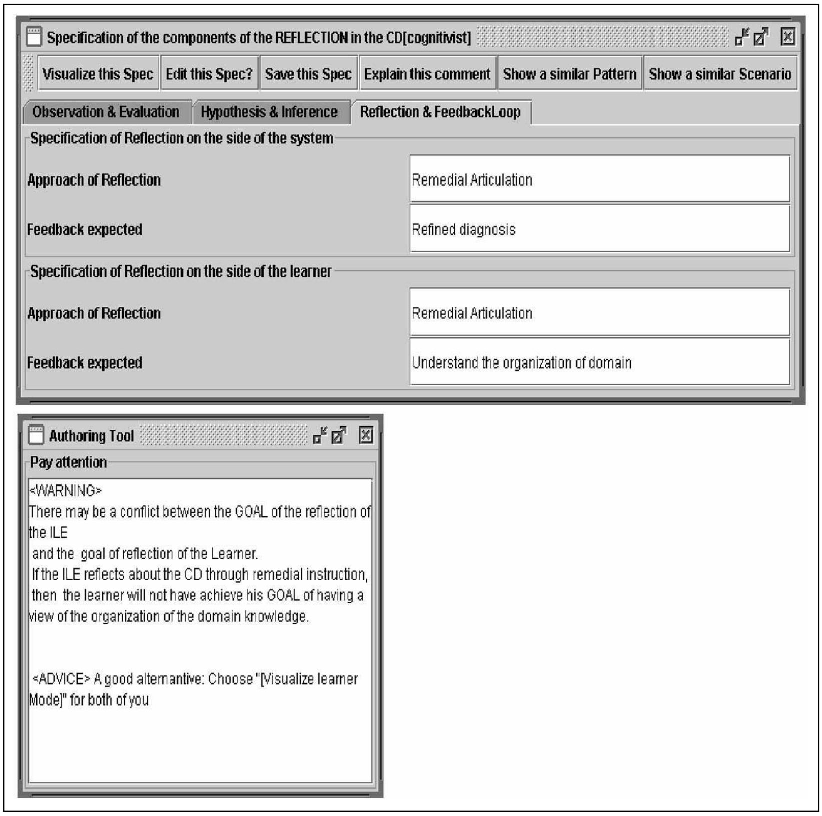

Figure 6. Verification of consistency in CD-SPECIES

(1) A semi-computational version of the algorithms that implement a specification may be produced. "Semi-computational" refers to a pseudo-algorithm which defines a behavior in an abstract manner. Computer platform and programming language details that will be used are ignored in the pseudo algorithm;

(2) A conceptual architecture of the CD component of an ILE which corresponds to a specification may be generated. This architecture is envisioned as a hierarchy of files that correspond to the computer program's *packages* that will implement the specification. Each file is intended to contain a description of a computer program or a set of data structures that will contribute to the implementation. The programmer would fill in each one of these files accordingly;

(3) CD-SPECIES may inform the programmer about the AI techniques that are best suited to implement a given specification in an ILE. For example, with a behavioral CD, in a context that integrates a cognitive view of learning, model tracing techniques based on task graphs or production rule systems are best suited.

CONCLUSION

There is currently a large gap between the instructional designer intentions with respect to the CD component of ILEs and its effective implementation. Besides, ILE instructional designers have mainly focused on presenting and organizing courseware, rather than on the explicit design of pedagogical functions. The first assumption of this article is that an explicit design of instruction in general, and of the CD in particular, will contribute to bridging this gap. The second assumption of this article is that beyond an explicit definition through a framework, the CD process in an ILE should integrate pedagogical interactions that foster a better recognition of the learner's knowledge by the system. Thus, two goals emerged with respect to these assumptions.

The first goal was to consider an approach to the design of a CD component for an ILE which takes into account the issue of reflection. Reflection is a very important issue in the current research trends in learner modeling. However, the impacts of its effect on learning have yet to be well formalized in the process of learner modeling (or learners' CDs). In this respect, the first contribution of this article concerns the integration of reflection in the CD process, in the form of a feedback loop in a global framework for the CD of learners' errors. This feedback loop is a conceptualization of the different ways in which interactions between an ILE and a learner could enhance the results of the CD process as well as the pedagogical use of these results. This study illustrated, instantiated and implemented this generic re-conceptualization of the CD process in Prolog-Tutor, an ILE for Logic Programming.

The second goal of this article was to provide instructional designers with the means to specify this particular view of the CD design for a specific pedagogical context. We believe that allowing instructional designers to explicitly specify the manner in which they want the ILE to diagnose learners' errors will fill the gap between them and the ILE programmers. This article illustrated how a method of instructional design could use this framework to implement the knowledge base of CD-SPECIES, a computer tool intended to support the specification of the CD component for an ILE. Indeed, research in the design of the learners' cognitive state diagnosis has acknowledged the need to take into account learning, instructional and instructional design theories. Therefore, the design method that was proposed contributes to integrating this practice in the particular case of designing the CD component of ILEs. Moreover, an important issue in the design of pedagogically sound learning environments is the translation of educational science principles into the constraints of a computational system. According to the model of CD-SPECIES, it is intended to translate CD specifications into semi-computational specifications or into a generic architecture for a CD component in an ILE. In this respect, CD-SPECIES model contributes to bridging the gap between researchers in the domain of education and those from the field of AI in education.

The proposed framework is currently being evaluated by instructional designers and students experienced with Logic Programming. Instructional designers are involved to evaluate (1) the content of CD-SPECIES with respect to the properties of explicitness and exhaustiveness and (2) the relevance and the usefulness of the support functions provided by CD-SPECIES.

Students are involved in evaluating the effectiveness of Prolog-Tutor CD component, implemented on the basis of a CD-SPECIES specification. Given a CD specification, four questions are addressed in this evaluation:

(1) Does the way that reflection is implemented make the intention of the system to go through a process of reflection obvious?

(2) Does the way that reflection is implemented actually make the learners reflect?

(3) Does the way that reflection is implemented trigger a feedback loop that allows the learners to review their understanding of the domain learned?

(4) Does the way that reflection is implemented trigger a feedback loop that allows the system to adjust its assessment of the learners' cognitive state?

Results from this evaluation will provide a validation of the framework, as well as guidelines to improve the conceptualization and the implementation of CD-SPECIES and Prolog-Tutor.

References

Ainsworth, S., Major, N., Grimshaw, S., Hayes, M., Underwood, J., Williams, B., et al. (2003). REDEEM: Simple intelligent tutoring systems from usable tools. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 205-232, Dordrecht: Kluwer Academic Publishers.

Anderson, J., Corbett, A., Koedinger, K., & Pelletier, R. (1995). Cognitive tutors: Lessons learned. *The Journal of the Learning Sciences 4*(2), 167-207.

Arruarte, A., Ferrero, F., FernÃndez-Castro, I., Urretavizcaya, A., Ãlvarez, A., & Greer, J. (2003). The IRIS authoring tool. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning environments: Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 233-268, Dordrecht: Kluwer Academic Publisher.

Blessing, S. (2003). A programming by demonstration authoring tool for model-tracing tutors. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 93-120, Dordrecht.: Kluwer Academic Publishers.

Brusilovsky, P., & Peylo, C. (2003). Adaptive and intelligent web-based educational systems. *International Journal of Artificial Intelligence in Education, 13*(2-4), 159-172.

Chi, M. T., De Leeuv, N., Chiu, M., H., & Lavancher, C. (1994). Eliciting self-explanations improves understanding. *Cognitive Sciences, 18*(3), 439-477.

Conati, C., Gertner, A., & VanLehn, K. (2002). Using Bayesian networks to manage uncertainty in student modeling. *Journal of User Modeling and User-Adapted Interaction, 12*(4), 371-417.

Dewey, J. (1933). *How we think: A restatement of the relation of reflective thinking to the educative process*. Boston: Heath & Company.

Dimitrova, V., Self, J., & Brna, P. (2000). *Involving the Learner in Diagnosis: Potentials and Problems.* Tutorial presented at the Conference on Web Information Technologies: Research, Education and Commerce, MontPellier, France.

Greeno, J. G., Collins, A. M., & Resnick, L. B. (1996). Cognition and learning. In D. Berliner & R. Calfee (Eds.), pp. 15-45, *Handbook of educational psychology*. New York: Macmillan.

Heift, T., & Nicholson, D. (2001). Web delivery of adaptive and interactive language tutoring. *International Journal of Artificial Intelligence in Education, 12*(4), 310-324.

Katz, S., Lesgold, A., Hughes, E., Peters, D., Eggan, G., Gordin, M., et al. (1998). Sherlock 2: An intelligent tutoring system built on the LRDC framework. *International Journal of Artificial Intelligence in Education, 3*(4), 495-518.

Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., et al. (2001). ActiveMath: A web-based learning environment. *International Journal of Artificial Intelligence in Education, 12*(4), 385-407.

Merrill, M., D., & ID2-Research-Group. (1998). ID Expert: A second generation instructional development system. *Instructional Science, 26*, 243-262.

Mislevy, R. (1994). Evidence and inference in educational assessment. *Psychometrika, 59*, 439-483.

Mizoguchi, R., & Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems. *Journal of Artificial Intelligence and Education, Special Issue on AIED 2010, 11*, 107-121.

Morrison, G., Ross, S., & Kemp, J., E. (2004). *Designing effective instruction* (4th ed.). New York: John Wiley & Sons.

Murray, T. (2003). Eon: Authoring tools for content, instructional strategy, student model, and interface design. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 309-340, Dordrecht.: Kluwer Academic Publishers.

Nkambou, R., Frasson, C., & Gauthier, G. (1997). Un modèle de représentation des connaissances relatives au contenu dans un système tutoriel intelligent. *Sciences et Techniques Éducatives, 4*(3), 299-330.

Nkambou, R., Frasson, C., & Gauthier, G. (2003). CREAM-Tools: An authoring environment for knowledge engineering in intelligent tutoring systems. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software* (pp. 93-138). Dordrecht: Kluwer Academic Publisher.

Paquette, G., Bourdeau, J., Henri, F., Basque, J., Leonard, M., & Maina, M. (2003). Construction d,une base de connaissances et de ressources sur le téléapprentissage. *STICEF, 10*, Numéro Spécial Formation à distance, 29-56. http://sticef.org.

Reigeluth, C., M. (1999). *Instructional design theories and models: A new paradigm of instructional theory, Volume II.* Mahwah NJ: Lawrence Erlbaum Associates, Inc.

Self, J. (1994). Formal approaches to student modeling. In *Student Models: The Key to Individualized Educational Systems.* New York: Springer Verlag.

Tchetagni, J., Nkambou, R., & Bourdeau, J. (2005). *Supporting Student Reflection in an Intelligent Tutoring System for Logic Programming.* Proceedings of Lemore Workhop 2005 at the International Conference on Artificial Intelligence in Education, pp. 42-52.

Tchetagni, J., Nkambou, R., & Kabanza, F. (2004). *Epistemological remediation in intelligent tutoring systems.* Proceedings of IEA/AIE2004. pp. 955-966

Towne, D. (2003). Automated knowledge acquisition for intelligent support of diagnostic reasoning. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology Learning Environments Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 121-148, Dordrecht.: Kluwer Academic Publishers.

Towne Shute, V., & Torreano, L. (2003). Formative evaluation of an automated knowledge elicitation and organization tool. In T. Murray, S. Blessing & S. Ainsworth (Eds.), *Authoring Tools for Advanced Technology learning Environments: Toward cost-effective adaptive, interactive, and intelligent educational software*, pp. 149-180, Dordrecht.: Kluwer Academic Publishers.

Van Marcke, K. (1998). GTE: An epistemological approach to instructional modeling. *Instructional Science, 26*, 147-191.

Weber, G., & Brusilovsky, P. (2001). ELM-ART: An adaptive versatile system for web-based instruction. *International Journal of Artificial Intelligence in Education, 12*(4), 351-384.

Wenger, E. (1987). *Artificial intelligence and tutoring systems: Computational and cognitive approaches to the communication of knowledge*: Morgan Kaufman Press.