

Maintaining Diagnostic Interactions that Promote Learners' Reflection

Vania DIMITROVA
School of Computing, Leeds University, UK

Abstract. This paper presents an approach to dynamically extract individual learner models by engaging learners in diagnostic interactions. A framework for maintaining diagnostic dialogues based on approaches known as dialogue games is outlined and illustrated in STyLE-OLM - an interactive open learner modelling system. The potential for encouraging learners' reflection is discussed based on an example dialogue in STyLE-OLM. Possible improvements of the dialogue framework are sketched out.

1 Introduction

This paper discusses the application of an approach known as dialogue games to the maintenance of diagnostic interactions that extract models of the learners' cognition. The research presented here contributes to a recent stream in user and student modelling that stresses the importance of involving learners in diagnosis and considers approaches such as open and collaboratively constructed user/learner models [1, 2, 3, 4, 5, 6]. A distinct characteristic of our work is that the communication between a computer diagnoser and a human learner is considered to be the focal point when involving the learner in diagnosis. Hence, advanced dialogue framework is needed to maintain student modelling interactions. Such a framework is presented here. Similarly to Grasso et al. [7] we employ dialogue games to maintain user-computer dialogues. Distinctly from Grasso et al., who apply dialogue games to model argumentative dialogues, our work adapts dialogue games to simulate diagnostic dialogues that have specific characteristics, as discussed below. It is through these dialogues, originally aimed at extracting models of the learners' cognition, that the learners are encouraged to reflect on their understanding of the domain being studied.

Next in the paper, we outline characteristics of student modelling interactions (section 2); introduce dialogue games (section 3); outline a framework for maintaining diagnostic dialogues (section 4); illustrate the framework with an example and discuss potential of the approach for promoting learners' reflection (section 5); and point out future work (section 7).

2 Diagnostic Interactions Modelled Here

We will sketch out distinctive features of learner modelling interactions. The features are catered for in the framework for interactive diagnosis outlined in Section 3.

1. *Extracting learner models.* This entails the detection of learners' intentions from their utterances and the extraction of the beliefs the learners have committed to. The interaction may advance the learner modelling process enabling the detection of learners' misconceptions and the discovery of possible causes for some misconceptions.

2. *Learner's active involvement.* Although diagnostic interactions are usually guided by a diagnoser, the diagnosee's active involvement is crucial. Learner modelling dialogues should be highly interactive providing possibilities for agents to change their commitments and to influence others' cognitive processes. Moreover, the learners should have the power to influence the models the systems build of them.

3. *Different views about the learner model.* The computer and the learner can have different views about the learner model. These views may need to be maintained separately and have to be taken into account in the resultant model. Each party has to be provided the means both to manifest disagreements with the other's views and to demand/compose justifications that may change the participants' commitments.

4. *Mixed initiative.* Diagnostic interactions are mixed initiative interactions where each participant should be allowed to change the direction of the topic of the conversation or take the lead in discussing the current topic. At the same time, the diagnoser should steer the conversation in a way that allows specific diagnostic goals to be achieved and a coherent dialogue produced.

Driven by these characteristics, we have utilised dialogue games for maintaining diagnostic interactions.

3 Dialogue Games

Dialogue games (DGs) have been proposed by Levin and Moore [8] to model regularities in natural language dialogues and are defined as knowledge structures that represent multiple turn dialogue patterns organised around specific dialogue goals. A DG is represented by three parts:

1. Parameters that define specific values of the game, e.g. the participants involved in the DG and its goal;

2. Specifications that describe a particular situation in the world which triggers a DG and holds whilst the game is conducted;

3. Components that represent a set of subgoals to be addressed in the game and determine a sequence of utterances the participants produce in a dialogue game.

Levin and Moore propose that a DG model of interaction comprises:

1. Long term memory, which includes the knowledge the dialogue participants possess about the world before the dialogue starts;

2. Workspace, which contains all the partial and temporary results of processing;

3. Dialogue processors, which modify entities in the workspace. The processors are responsible for initiating a game when the specifications are satisfied and terminating it whenever any condition in the specifications ceases to hold.

While Levin and Moore's theory allows inferring the goals the people might have for generating their utterances, which is critical in modelling learners, their theory does not define how people can influence others' cognitive states and does not ensure equal involvement of all participants. One possible approach to deal with these issues is the logical dialogue game theory developed by philosophers to enable the argument in rational human debates to proceed fairly and the participants' to have

equal power [9]. A DG in this theory consists of a set of dialogue moves to represent the interaction turns, players' commitment stores that include statements the players have committed to in the debate, commitment rules that define changes to the commitment stores as results of the dialogue moves, and game rules, which define allowed sequences of moves. A distinction is made between structural rules that declare syntactical conventions in the game and strategic rules that a player can follow in order to win the game based on assumptions the agents make about the reasoning of their opponents. Logical DGs distinguish between publicly proclaimed commitments and de facto commitments which agents hold applying their reasoners.

Both DG approaches discussed above will be combined in a dialogue framework for interactive learner modelling, which is presented next.

4 A Dialogue Framework for Interactive Diagnosis

This section presents a computational framework for maintaining the interaction between a computer (denoted with c) and a learner (denoted with u) discussing the beliefs of the latter. We assume that c has an appropriate domain expertise represented in a domain ontology and employs appropriate techniques to extract knowledge needed for maintaining dialogue focus and reasoning about the student's beliefs. The use of domain ontology for interactive diagnosis is outside the scope of this paper and is discussed in detail in [10].

4.1 Learner Model

We consider a learner model structured as enumerative bug model [11] that incorporates learner's beliefs (domain propositions) which can be correct, erroneous, and incomplete. The learner model includes also some possible explanations of causes for the erroneous beliefs, which are based on erroneous reasoning such as misclassification and misattribution [12]. The system's assumptions about the learner's reasoning are used in the dialogue management.

4.2 Main Components of the Dialogue Framework

The dialogue framework described here follows Levin and Moore's DG theory to organise the dialogue episodes and utilises logical DGs to enable the participants' symmetrical involvement and to collect their commitments.

Communicative acts (CAs) are defined as quadruples $\langle S, H, M, P \rangle$, where S is the Speaker, H - Hearer, M - Move (a performative verb that describes the illocutionary force of the CA), and P - domain Proposition. The moves are taken from (Pilkington, 1999) and adapted for diagnostic dialogues. They include: *Inform* (S believes a proposition and informs H about this), *Inquire* (S asks about a proposition), *Challenge* (S doubts a proposition), *Disagree* (S disagrees with a proposition), *Justify* (S explains why a proposition is correct), *Agree* (S agrees with a proposition), *Suggest* (S suggests a new topic for discussion), *Deny* (S does not accept a topic for discussion), *Accept* (S accepts a topic for discussion), *Skip* (S skips its turn and passes the initiative to H).

Dialogue rules are defined as $(\text{move1}, p1) \rightarrow (\text{move2}, p2)$ to postulate that a CA with move2 and proposition p2 is permitted if the previous turn has included move1 and proposition p1. For instance, $(\text{inform}, p) \rightarrow (\text{challenge}, p)$ allows a statement made

by one of the agents to be challenged by the other. The dialogue rules are valid throughout the whole interaction to allow participants to make statements, ask questions, suggest discussion topics, or skip their turns at any time of the dialogue. Discarding the win-lose principle of the logical DGs [9], we consider that the learner shall not be forced to follow the game rules, instead, these rules are used by the computer to maintain dialogue coherence and to understand the learner's contributions. In the same line of argument, the participants in diagnostic interactions may change their minds challenging, withdrawing, or contradicting statements they have made earlier, which enables capturing the dynamics of learners' cognition.

Commitment stores accumulate the commitments of the agents disclosed throughout the interaction. The Student's Commitment Store (SCS) contains his beliefs about the domain, while the Computer's Commitment Store (CCS) includes its beliefs about the beliefs of the learner. As mentioned earlier, the learner and the system may have different views about the learner model. These views are represented in the commitment stores and are combined when the interaction finishes to extract a resultant learner model. SCS includes two types of belief expressions $Bs(p)$ (s believes the proposition p) or $\neg Bs(p)$ (s does not believe p). CCS includes $Bc(Bs(p))$ (c believes that s believes p) and $\neg Bc(Bs(p))$ (c does not believe that s believes p). Likewise, we will denote the computer's domain beliefs, extracted from its knowledge base, with $Bc(p)$ (the computer domain expertise supports p) or $\neg Bc(p)$ (the computer domain expertise does not support p).

Commitment rules define the effects of moves upon the agents' commitment stores. There are two operations add - appends a belief to a commitment store - and delete - removes a belief from a store. For example, when the student makes a statement p, $Bs(p)$ will be added to SCS and $Bc(Bs(p))$ will be added to CSS. A disagreement with p expressed by the student will enforce the removal of all positive beliefs about p in SCS and adding $\neg Bs(p)$. The commitment stores' consistency is maintained with belief revision similar to [2]. Note that the commitment rules do not impose many changes in the hearer's store but a reasoning mechanism adds more beliefs applying commonsense reasoners and assuming that agents accept beliefs that do not contradict their beliefs.

Dialogue history stores all performed CAs and is used by some of the commitment rules. It also enables the dialogue focus maintenance.

Dialogue games define interaction episodes, which correspond to certain diagnostic goals and discussion topics. Following Levin and Moore [8], the DGs are defined as triples (Parameters, Specification, Components), where

- *Parameters* represent values specific for the game and include: *Goal* (the goal of the dialogue game) and *Focus_Space* (list of focus concepts relevant to this game).
- *Specifications* define conditions necessary for the game to take place. Each condition is an expression that includes beliefs of the dialogue participants.
- *Components* determine a sequence of utterances the game generates and are represented in: *Schema* (algorithm for extracting Relevant_Propositions to be discussed in the game); *Game_Tactic* (plan, obtained by using the *Relevant_Propositions* and the current state of the commitment stores, comprising CAs the diagnoser will address in the dialogue).

We do not aim to classify all types of interactive diagnostic patterns, which would require studies of many naturalistic diagnostic situations. Instead, we follow a generic model of interactive diagnostic dialogues capable of capturing a variety of

patterns related to this type of interaction and have experimented with three types of DG:

- *Exploratory games* aim at collecting more information about the learner's knowledge of a specific topic and can be initiated by both agents.
- *Explanatory games* aim at discovering a possible reason for the learner's erroneous beliefs and can be initiated by the computer when there is a pattern of the learner's misconception for which a schema can be applied to define dialogue tactics.
- *Negotiative games* aim at clarifying agents' positions when discrepancies in their views are discovered and can be initiated by both parties asking for justifications.

The interaction in STyLE-OLM is organised as a series of DGs. At any moment of the dialogue several dialogue games can be open, while one of them is active and represents the current dialogue episode. The open DGs can become active in future interactions if the participants shift back to uncompleted issues. A DG stack has on its top the active DG and contains all DGs that are open at the current point. An active game can be terminated and deleted from the stack or suspended and left in the stack while its top place is taken by another active game, which can be either a newly initiated DG or a re-activated DG that complies with the dialogue focus.

4.3 Dialogue processing

We follow Levin and Moore's DG processing model (see Figure 1):

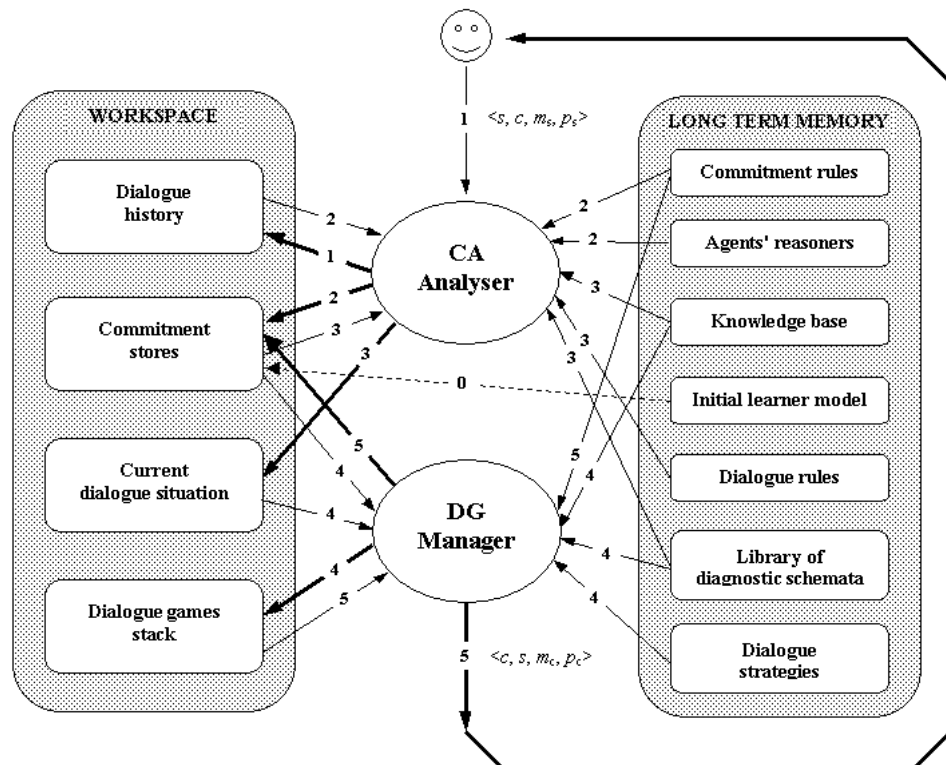


Figure 1: Architecture of dialogue processing for interactive learner modelling.

1. *Long term memory* is the knowledge the participants bring to dialogue before it starts and includes: domain ontology with the domain expertise of the diagnoser, initial learner model which may be accumulated from previous interactions or by using other diagnostic methods, assumptions about the participant's domain reasoners

to infer consequences of their commitments expressed in the dialogue, and discourse knowledge in terms of dialogue rules and commitment rules. In addition, the computer uses a library of diagnostic schemata to collect relevant propositions to discuss and a set of dialogue strategies to define which DG to open/re-activate when there are several possibilities.

2. *Workspace* accumulates all temporary results of the dialogue and comprises of: commitment stores (SCS and CCS), dialogue history, DG stack, and a structure that represents the current dialogue situation (see below).

3. *Dialogue processors* are algorithms that modify the entities in the workspace and include: CA analyser that examines the student's input and fills in parameters of the current dialogue situation on the workspace and DG manager that undertakes operations over the DG stack based on the current dialogue situation, the beliefs in the commitment stores, and the DG stack's state.

Current dialogue situation is defined as $s = \langle s_1, s_2, \dots, s_n \rangle$, where s_1, s_2, \dots, s_n represent constraints on dialogue features and taken in conjunction define the state of the dialogue at that moment.

There are five processes invoked in turn and an initialisation process (marked with 0 in Figure 1) where an initial learner model is used for initialising SCS and CCS (note that the initial learner model may be empty).

The interaction cycle consists of analysing the learner's CA $\langle s, c, m_s, p_s \rangle$ and generating a computer's CA $\langle c, s, m_c, p_c \rangle$.

The five main processes of dialogue management are:

1. The CA analyser reads the student's CA and updates the dialogue history.
2. The CA analyser updates the participants' commitment stores applying commitment rules upon the current CA. Using agents' reasoners, de facto commitments are also inferred. The collected participants' beliefs are deposited in the corresponding commitment stores.

3. The CA analyser examines the current dialogue situation and fills in corresponding features. It considers the participants' beliefs, compares the learner's CA with the dialogue rules, calls domain inference mechanisms to assess the correctness of the learner's claim, and searches for diagnostic schema when an erroneous learner's belief is discovered.

4. The DG manager examines the commitment stores and the current dialogue situation and, applying dialogue strategies, performs changes in the DG stack. When a new DG has to be initiated, the DG manager selects a schema in order to generate an appropriate dialogue tactic.

5. Finally, the DG manager selects the corresponding CA from the active DG and sends this act as a system's response to the learner. Follows a learner's turn.

When the interaction is terminated, a formal mechanism based on modal logic combines the beliefs in the commitment stores and elicits a resultant learner model [10].

5 Promoting Learners Reflection

This section will illustrate the work of the dialogue management framework utilised in STyLE-OLM - an interactive open learner modelling system where a learner is enabled to discuss and inspect the content of the learner model. The communication is conducted in a graphical manner but is presented below in a textual form due to space constraints. The illustration is from a discussion in a Computing domain (topic -

Programming Languages). The domain propositions are represented in STyLE-OLM with conceptual graphs (CGs) [13]. STyLE-OLM applies CG reasoning to extract Relevant_Propositions, Focus_Space, and Tactics.

The interaction starts with an initial learner model that does not contain information about propositions relevant to this example, which will allow us to better follow the effect of the dialogue on the commitment stores.

```
{1} S[inform] I think that VISUAL BASIC is an OBJECT-ORIENTED
LANGUAGE.
```

The learner makes an erroneous statement $p1$ ="Visual Basic is an object-oriented language". Applying CG specification and generalisation operations [13], STyLE-OLM identifies the following propositions to be discussed with the learner: $q1$ ="Object-oriented languages contain objects"; $q2$ ="Objects pass messages between themselves"; $q3$ ="The main characteristics of object-oriented languages are inheritance and encapsulation"; $q4$ ="Visual Basic contains objects"; $q5$ ="Visual Basic has inheritance and encapsulation"; $q6$ ="Visual Basic is a visual language"; $q7$ ="Visual C++ is a visual language."; $q8$ ="Visual C++ is an object-oriented language"; $q9$ ="Visual C++ has inheritance and encapsulation". Note that there may be much more propositions to be discussed, only few are listed here for simplicity.

There are four possible DGs (see Figure 2) to be initiated as the specifications for them are satisfied. Note that the dialogue tactics assign an inquire CA when the system does not have information whether the learner believes a proposition and an inform CA when it assumes that the learner believes the proposition.

The DG manager initiates *negotiate_game*($p1$), see Figure 2 (left), following a strategy rule suggesting that the diagnoser shall challenge the learner's erroneous claims.

<p>explore_game(p_1)</p> <p><u>Parameters:</u></p> <p>Goal = [explore, p_1]</p> <p>Focus_Space = [OBJECT-ORIENTED LANGUAGE, OBJECT, MESSAGE, INHERITANCE, ENCAPSULATION, VISUAL BASIC, VISUAL LANGAUGE]</p> <p><u>Specifications:</u> $B_c(B_u(p_1))$</p> <p><u>Components:</u></p> <p>Relevant_Propositions = [q_1, q_2, q_3, q_4]</p> <p>List_of_Communicative_Acts = [$\langle c, u, inform, q_1 \rangle, \langle c, u, inquire, q_2 \rangle, \langle c, u, inquire, q_3 \rangle, \langle c, u, inform, q_4 \rangle$]</p> <p>negotiate_game(p_1)</p> <p><u>Parameters:</u> Goal = [negotiate, p_1]</p> <p>Focus_Space = [OBJECT-ORIENTED LANGUAGE, VISUAL BASIC]</p> <p><u>Specifications:</u> $B_u(p_1)$ and $\neg B_c(p_1)$</p> <p><u>Components:</u> Relevant_Propositions = [p_1]</p> <p>List_of_Communicative_Acts = [$\langle c, u, challenge, p_1 \rangle$]</p>	<p>explain_game(p_1)</p> <p><u>Parameters:</u></p> <p>Goal = [explain, p_1]</p> <p>Focus_Space = [OBJECT-ORIENTED LANGUAGE, OBJECT, VISUAL BASIC, INHERITANCE, ENCAPSULATION]</p> <p><u>Specifications:</u> ($B_c(B_u(p_1))$ and $\neg B_c(p_1)$) or ($B_u(p_1)$ and $\neg B_c(p_1)$)</p> <p><u>Components:</u></p> <p>Relevant_Propositions = [q_1, q_4, q_5]</p> <p>List_of_Communicative_Acts = [$\langle c, u, inform, q_1 \rangle, \langle c, u, inform, q_4 \rangle, \langle c, u, inquire, q_5 \rangle$]</p> <p>explain_game(p_1)</p> <p><u>Parameters:</u> Goal = [explain, p_1]</p> <p>Focus_Space = [VISUAL BASIC, VISUAL LANGUAGE, VISUAL C++, OBJECT-ORIENTED LANGAUGE, INHERITANCE, ENCAPSULATION]</p> <p><u>Specifications:</u> ($B_c(B_u(p_1))$ and $\neg B_c(p_1)$) or ($B_u(p_1)$ and $\neg B_c(p_1)$)</p> <p><u>Components:</u> Relevant_Propositions = [$q_6, q_7, q_8, q_9, q_5, q_3$]</p> <p>List_of_Communicative_Acts = [$\langle c, u, inform, q_6 \rangle, \langle c, u, inquire, q_7 \rangle, \langle c, u, inquire, q_8 \rangle, \langle c, u, inquire, q_5 \rangle, \langle c, u, inquire, q_3 \rangle$]</p>
--	--

Figure 2: Dialogue games that can be initiated after the learner's statement in CA {1}.

```
{2} C[challenge] Why do you think that VISUAL BASIC is an OBJECT-
ORIENTED LANGUAGE?
```

The learner's domain knowledge is challenged and he searches for grounds for his domain beliefs, which is identified as a reflective activity [14]. The learner answers:

```
{3a} S[justify] I think that VISUAL BASIC is an OBJECT-ORIENTED
LANGAUGE because it contains OBJECTS.
```

Following a strategy rule to pursue explanations every time the learner makes a statement that relates to a misconception pattern and a corresponding schema can be found, the system would initiate an explain DG as the one shown in the top right corner in Figure 2 (explaining that the learner may wrongly believe that Visual Basic is an object-oriented language because Visual Basic has objects) and will direct the dialogue to discussing main characteristics of object-oriented languages:

```
{4a} C[inquire] Do you know what are the main characteristics of  
OBJECT-ORIENTED LANGUAGES?
```

Let us assume that searching for grounds for his domain beliefs the learner answers differently, instead of {3a} we now have {3b}:

```
{3b} S[justify] I think that VISUAL BASIC is an OBJECT-ORIENTED LANGAUGE because it is a  
VISUAL LANGUAGE.
```

Following the same strategy rule as in {4a}, the system will initiate an explain DG as the one shown in the bottom right corner in Figure 2 (explaining that the learner may wrongly believe that Visual Basic is an object-oriented language because Visual Basic is a visual language and is similar to Visual C++ which is a visual language but also an object-oriented language). The system will pursue this explanation:

```
{4b} C[inquire] Do you think that VISUAL C++ is a VISUAL LANGUAGE.
```

It may well happen that the learner does not make a statement that directs to a possible explanation, for example he may discover that his knowledge about object-oriented languages is limited and may suggest this topic for further discussion, i.e. instead of {3a} or {3b} we may have:

```
{3c} U[suggest] Let us talk about OBJECT-ORIENTED LANGAUGES.
```

In this case the learner initiates an explore DG as the one shown in the bottom left corner in Figure 2. This game will be made active and the system will continue the dialogue discussing properties of object-oriented languages.

As shown in the example, the dialogue maintenance mechanism encourages learners to explore aspects of their domain knowledge coherently revisiting and recalling related concepts, which is a main reflection activity [15].

The presence of reflection was monitored in a small evaluative study of STyLE-OLM [10]. The study involved seven post graduate students from the author's department who interacted with a STyLE-OLM instantiation in a Finance domain. The dialogue management was examined by analysing log files. The presence of several reflective activities was found, as discussed in detail in [16]. In short, the students made claims about their beliefs, went back to claims they made and (sometimes) changed their beliefs. Thus, the students recalled and reconsidered domain aspects [15]. Examples of such activities included inform, agree, challenge, or disagree moves from the learners to confirm or alter their beliefs. The students also investigated arguments to support their beliefs, i.e. they searched for grounds of their beliefs [14]. These activities were commonly initiated by a system's challenge followed by either a learner's justification (uttered as an inform or justify move) or a learner's question (uttered as an inquire move) to clarify domain aspects.

An important feature of the interaction with STyLE-OLM was that learners were allowed to take the initiative in maintaining the dialogue by changing the focus of conversation or initiating new dialogue games. The learners found the dialogue moves useful and did not experience major problems to participate in the interactions. The focus maintenance (based on CG inference) was relatively robust and allowed discussing connected terms and elaborating more aspects of the learners' domain

knowledge. The interactions with STyLE-OLM were classified by most of the learners as discussions about their domain knowledge. The participants were not aware that two different views of the student models were maintained but felt that they could influence the system's diagnosis (in fact, they did). We ought to mention that opening the learner model helped to overcome some deficiencies in dialogue management - at every time the learners could look at their models and initiate dialogue to correct beliefs that were assigned wrongly.

6 Future Work

The evaluation of STyLE-OLM confirmed that learners should be given responsibility for the maintenance of the models the system builds of them and that this involvement should be done through carefully planned dialogue, which may not only lead to a better quality learner model but may also promote learners' reflection on the domain being studied. While the study showed the feasibility of the dialogue framework presented here, it also revealed some aspects that need further investigation. For instance, at times the interactions with STyLE-OLM included peculiar episodes due to flaws in the domain ontology, e.g. the system challenged because its domain expertise did not confirm a proposition due to incompleteness of the domain ontology or plausible DGs were not followed because misconception patterns were not confirmed due to vagueness in defining misconceptions. A possible improvement may consider using clarification dialogues and uncertainty management methods [17]. A major deficiency of the dialogue in STyLE-OLM was the lack of semantic structure in rendering the system's CAs. As a result, some learners found that the system jumped from sentence to sentence without any obvious reason and they hardly followed what was going on because there was a lack of explanation of the purpose of the system's CAs. An enhanced DG mechanism is needed which may consider complex rhetorical relations between DGs and DG nesting [18]. More types of DG types need to be included, such as adaptive explanations [19], comparison [20], and error repair [21].

Finally, the dialogue maintenance mechanism in STyLE-OLM did not consider explicitly CAs aimed at promoting reflection. Although the evidence gathered of reflective activities is encouraging, these activities were somehow a side effect of the interaction (in fact, interaction is a necessary condition for reflection in open learner modelling [22]). It still remains an open question whether the dialogue maintenance mechanism should embed goals that explicitly aim at promoting students' reflection, and how such goals might be incorporated in dialogue strategies in line with the main objective of diagnostic dialogues - extracting a model of the student's cognition.

Acknowledgements. Many of the ideas presented here are result of the discussions the author had with John Self and Paul Brna who supervised her PhD studies. Their constant encouragement and constructive criticism are greatly acknowledged.

References

1. Kay, J. (1995). The UM toolkit for cooperative user modelling. *User Modeling and User-Adapted Interaction*, 4, 149-196.
2. Paiva, A., & Self, J.A. (1995). TAGUS - a user and learner modelling workbench. *User Modeling and User-Adapted Interaction*, 4, 197-226.

3. Bull, S., Brna, P., & Pain, H. (1995). Extending the scope of student models. *User Modeling and User-Adapted Interaction*, 5(1), 45-65.
4. McCalla, G., Vassileva, J., Greer, J., & Bull, S. (2000). Active learner modelling. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Intelligent tutoring systems* (pp. 53-62). Berlin Heidelberg: Springer-Verlag.
5. Morales, R., Pain, H., & Conlon, T. (2000). Understandable learner models for a sensorimotor control task. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Proceedings of the 5th International conference on Intelligent Tutoring Systems* (pp. 222-231). Springer, LNCS.
6. Zapata-Rivera, J., & Greer, J. (2002) Inspecting and visualizing distributed Bayesian student models. In G. Gauthier, C. Frasson & K. VanLehn (Eds.), *Intelligent Tutoring Systems, Proceedings of ITS2002* (pp. 544-553). Berlin Heidelberg: Springer Verlag.
7. Grasso, F., Cawsey, A., & Jones, R. (2000). Dialectical argumentation to solve conflicts in advice giving: a case study in the promotion of healthy nutrition. *International Journal Human-Computer Studies*, 53, 1077-1115.
8. Levin, J., & Moore, J. (1977). Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science*, 1 (4), 395-420.
9. Walton, D. (1984). *Logical dialogue games and fallacies*. University Press of America, Lanham.
10. Dimitrova, V. (2001) *Interactive open learner modelling*. PhD thesis, University of Leeds.
11. Holt, P., Dubs, S., Jones, M., & Greer, J. (1993). The state of student modelling. In J. Greer & G. McCalla (Eds.), *Student Modelling: The Key to Individualized Knowledge-Based Instruction* (pp. 3-38). Springer-Verlag.
12. McCoy, K. (1989). Highlighting a user model to respond to misconceptions. In A. Kobsa & W. Wahlster (Eds.), *User models in dialogue systems* (pp. 233-253). Springer-Verlag, Berlin.
13. Sowa, J. (1984). *Conceptual structures: Information processing in mind and machine*. MA: Addison-Wesley.
14. Dewey, J. (1960). *How we think - a restatement of the relation of reflective thinking to the educational process*. D.C. Heath and Company.
15. Boud, D., Keogh, R., & Walker, D. (1996). What is reflection in learning. In D. Boud, R. Keogh & D. Walker (Eds.), *Reflection: turning experience into learning* (pp. 7-17). London: Kogan Page.
16. Dimitrova, V. (2002). STyLE-OLM: Interactive open learner modelling. *International Journal of AI in Education*, vol 13, to appear.
17. Horvitz, E., & Paek, T. (2001). Harnessing models of learners' goals to mediate clarification dialog in spoken language systems. In M. Bauer, P. Gmytrasiewicz & J. Vassileva, *Proceedings of User Modelling 2001* (pp. 3-13). Springer-Verlag, LNCS.
18. Pulman, S. (2002). Relating dialogue games to information states. *Speech Communication*, 36, 15-30.
19. Dimitrova, V., & Bontcheva, K. (2003). The use of conceptual graphs for interactive student modelling and adaptive web explanations, 7th international conference on Knowledge-Based Intelligent Information and Engineering Systems, KES2003, Oxford, UK.
20. Milosavljevic, M. (1997). Augmenting the user's knowledge via comparison. In A. Jameson, C. Paris & C. Tasso (Eds.), *Proceedings of the 6th International conference on User Modeling*.
21. VanLehn, K., Siler, S., Murray, C., & Bagget, W. (1998). What makes a tutorial event effective? In: M.A. Gernsbacher & S. Derry (Eds.), *Proceedings of the 21st Annual Conference of the Cognitive Science Society* (pp. 1084-1089). Hillsdale, NJ: Erlbaum.
22. Zapata-Rivera, J. (2003). *Learning Environments based on Inspectable Student Models*. PhD Thesis, University of Saskatchewan, Canada.