

# Using Dialogue Games to Maintain Diagnostic Interactions

Vania Dimitrova

School of Computing, Leeds University, UK  
vania@comp.leeds.ac.uk

**Abstract.** This paper presents an approach to dynamically extract individual user models by engaging users 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 student modelling system. The framework is validated in an evaluative study of STyLE-OLM. Potential improvements are sketched out.

## 1 Introduction

It has been acknowledged that user models have to represent not canonical users but to capture specific aspects of each individual (Rich, 1999). One of the challenges to individual user modelling is to accommodate the dynamics of users' behaviour, especially when the users are students whose cognitive models gradually develop (Self, 1990). A possible approach to dynamically extract individual user models is discussed in this paper. We consider *intelligent computer diagnosers that interact with human users and extract models of the users' cognition*. Such extended diagnostic interactions are increasingly needed in advanced computer applications that require sophisticated user models to provide effective personalisation as well as aim at helping users understand their own needs and problems.

This paper discusses the application of an approach known as dialogue games to the maintenance of diagnostic interactions that extract models of the users' cognition. The computational framework proposed here is related to the work of Grasso et al. (2000) who apply dialogue games to model argumentative dialogues. Differently from this work, our approach adapts dialogue games to simulate diagnostic dialogues that have specific characteristics, as discussed below. A recent stream in user and student modelling stresses the importance of involving users in diagnosis and considers approaches such as open and collaboratively constructed user models (Kay, 1995; Paiva & Self, 1995; Bull et al., 1995; McCalla et al., 2000; Morales et al., 2000; Zpata-River & Greer, 2002). The framework discussed here contributes to this stream proposing an original approach for involving users in diagnosis where the communication between a computer diagnoser and a human user is the focal point.

Next in the paper, we will outline distinctive characteristics of the phenomena to be modelled, namely diagnostic dialogues (Section 2). We will then introduce dialogue games focusing on their use for user modelling (Section 3). Section 4 will present a framework for maintaining diagnostic dialogues, while Section 5 will illustrate the

framework with an example from a student modelling system. Finally, we will discuss advantages of the proposed architectures and will sketch out possible improvements.

## 2 Diagnostic Interactions Modelled Here

We will outline distinctive characteristics of user modelling interactions and will use these characteristics as a basis for developing a framework for interactive diagnosis.

1. *Extracting user models.* This entails the detection of users' intentions from their utterances and the extraction of the beliefs the users have committed to. The interaction may advance the user modelling process enabling the detection of users' misconceptions and the discovery of possible causes for some misconceptions.
2. *User's active involvement.* Although diagnostic interactions are usually guided by a diagnoser, the diagnosee's active involvement is crucial. User modelling dialogues should be highly interactive providing possibilities for agents to change their commitments and to influence others' cognitive processes. Moreover, the users should have the power to influence the models the systems build of them.
3. *Different views about the user model.* The computer and the user can have different views about the user 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 (1977) 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 users, 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 (Walton, 1984). 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 user modelling, which is presented next.

## 4 A Dialogue Framework for Interactive User Modelling

This section presents a computational framework for maintaining the interaction between a computer (denoted with *c*) and a user (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 (Dimitrova, 2001).

### 4.1 User Model

We consider a user model structured as enumerative bug model (Holt et al., 1993) that incorporates user's *beliefs* (domain propositions) which can be correct, erroneous, and incomplete. The user model includes also some possible *explanations* of causes for the erroneous beliefs, which are based on erroneous reasoning such as *misclassification* and *misattribution* (McCoy, 1989). The system's assumptions about the user'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  $(move_1, p_1)? (move_2, p_2)$  to postulate that a CA with  $move_2$  and proposition  $p_2$  is permitted if the previous turn has included  $move_1$  and proposition  $p_1$ . For instance,  $(inform, p)? (challenge, p)$  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 (Walton, 1984), we consider that the user 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 user'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 users' cognition.

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

**Commitment rules** define the effects of moves upon the agents' commitment stores (see Table 1). There are two operations *add* - appends a belief to a commitment store - and *delete* - removes a belief from a store. For example, *add p* to UCS will be executed as appending  $B_u(p)$ , while *add not p* to UCS will result in adding  $\neg B_u(p)$ . Delete enforces the removal of all *positive* beliefs about  $p$  in an agent's commitment store, e.g. *delete p* from CCS will remove  $B_c(B_u(p))$ , if present. The commitment stores' consistency is maintained with belief revision following (Paiva & Self, 1995).

The commitment rules do not impose many changes in the hearer's store, however, a reasoning mechanism for maintaining commitment stores described in (Dimitrova et al., 2000) will add more beliefs in both commitment stores by applying commonsense reasoners and assuming that agents accept beliefs that do not contradict their beliefs.

**Table 1.** Commitment rules for diagnostic dialogues.

Dialogue move	Commitment	Store
	Speaker	Hearer
<i>(inform,p)</i>	add <i>p</i>	add <i>p</i>
<i>(inquire,p)</i>	add <i>not p</i>	
<i>(challenge,p)</i>	delete <i>p</i>	
<i>(disagree,p)</i>	delete <i>p</i> ; add <i>not p</i>	
<i>(justify,p)</i> (after <i>(challenge,q)</i> or <i>(disagree,q)</i> )	add ( <i>p</i> $\mathcal{P}$ <i>q</i> )	
<i>(agree,p)</i>	add <i>p</i>	
<i>(suggest,p)</i> , <i>(deny,p)</i> , <i>(accept,p)</i> , <i>(skip,p)</i>		

**Dialogue history** stores all performed CAs and is used by some of the commitment rules (see Table 1). 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 (1977), the DGs are defined as triples  $\langle Parameters, Specification, Components \rangle$ , where

? *Parameters* represent values specific for the game. We consider:

*Goal* - the goal of the dialogue game;

*Focus\_Space* - a 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 user's knowledge of a specific topic and can be initiated by both agents.

? *Explanatory games* aim at discovering a possible reason for the user's erroneous beliefs and can be initiated by the computer when there is a pattern of the user'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):

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 user model* which may be accumulated from previous interactions or by using other diagnostic methods, assumptions about the participant's *domain reasons* 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* (UCS 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.

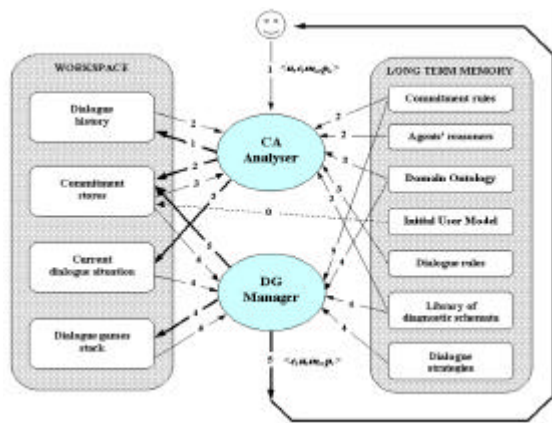


Figure 1. Architecture of dialogue processing for interactive user modelling.

The five main processes of dialogue management are:

1. The *CA analyser* reads the student's CA and updates the *dialogue history*.

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

The interaction cycle consists of analysing the user's CA  $\langle u, c, m_u, p_u \rangle$  and generating a computer's CA  $\langle c, u, m_c, p_c \rangle$ .

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 user's CA with the dialogue rules, calls domain inference mechanisms to assess the correctness of the user's claim, and searches for diagnostic schema when an erroneous user'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 user. Follows a user'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 user model. This mechanism is described elsewhere, see (Dimitrova et al., 2000).

## 5 An Example

This section will illustrate the work of the dialogue management framework utilised in a student modelling system - STyLE-OLM. STyLE-OLM is 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 (Dimitrova et al., 2002) 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) (Sowa, 1984). STyLE-OLM applies CG reasoning to extract *Relevant Propositions*, *Focus Space*, and *Tactics*, see (Dimitrova, 2001).

We will assume that 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} U[inform] I think that VISUAL BASIC is an OBJECT-ORIENTED
LANGUAGE.
```

The user makes an erroneous statement  $p_1$ ="Visual Basic is an object-oriented language". Applying CG specification and generalisation operations (Sowa, 1984), STyLE-OLM identifies the following propositions to be discussed with the user:  $q_1$ ="Object-oriented languages contain objects";  $q_2$ ="Objects pass messages between themselves";  $q_3$ ="The main characteristics of object-oriented languages are inheritance and encapsulation";  $q_4$ ="Visual Basic contains objects";  $q_5$ ="Visual Basic has inheritance and encapsulation";  $q_6$ ="Visual Basic is a visual language";  $q_7$ ="Visual C++ is a visual languag.";  $q_8$ ="Visual C++ is an object-oriented language";  $q_9$ ="Visual C++ has inheritance and encapsulation". Note that there may

be much more propositions to be discussed, we have selected few for a clarity of the example.

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 user believes a proposition and an *inform* CA when it assumes that the user believes the proposition.

The *DG manager* initiates *negotiate\_game(p<sub>1</sub>)*, shown in the top left corner in Figure 2, following a strategy rule suggesting that the diagnoser shall challenge the user's erroneous claims.

<p><b>explore_game(p<sub>1</sub>)</b>  <u>Parameters:</u>            Goal = [explore, p<sub>1</sub>]            Focus_Space = [OBJECT-ORIENTED LANGUAGE, OBJECT, MESSAGE, INHERITANCE, ENCAPSULATION, VISUAL BASIC, VISUAL LANGUAGE]  <u>Specifications:</u> B<sub>c</sub>(B<sub>u</sub>(p<sub>1</sub>))  <u>Components:</u>            Relevant_Propositions = [q<sub>1</sub>, q<sub>2</sub>, q<sub>3</sub>, q<sub>4</sub>]            List_of_Communicative_Acts = [&lt;c, u, inform, q<sub>1</sub>&gt;, &lt;c, u, inquire, q<sub>2</sub>&gt;, &lt;c, u, inquire, q<sub>3</sub>&gt;, &lt;c, u, inform, q<sub>4</sub>&gt;]</p> <p><b>negotiate_game(p<sub>1</sub>)</b>  <u>Parameters:</u> Goal = [negotiate, p<sub>1</sub>]            Focus_Space = [OBJECT-ORIENTED LANGUAGE, VISUAL BASIC]  <u>Specifications:</u> B<sub>c</sub>(p<sub>1</sub>) and ¬B<sub>u</sub>(p<sub>1</sub>)  <u>Components:</u> Relevant_Propositions = [p<sub>1</sub>]            List_of_Communicative_Acts = [&lt;c, u, challenge, p<sub>1</sub>&gt;]</p>	<p><b>explain_game(p<sub>1</sub>)</b>  <u>Parameters:</u>            Goal = [explain, p<sub>1</sub>]            Focus_Space = [OBJECT-ORIENTED LANGUAGE, OBJECT, VISUAL BASIC, INHERITANCE, ENCAPSULATION]  <u>Specifications:</u> (B<sub>c</sub>(B<sub>u</sub>(p<sub>1</sub>)) and ¬B<sub>u</sub>(p<sub>1</sub>)) or (B<sub>u</sub>(p<sub>1</sub>) and ¬B<sub>c</sub>(p<sub>1</sub>))  <u>Components:</u>            Relevant_Propositions = [q<sub>1</sub>, q<sub>2</sub>, q<sub>3</sub>]            List_of_Communicative_Acts = [&lt;c, u, inform, q<sub>1</sub>&gt;, &lt;c, u, inform, q<sub>2</sub>&gt;, &lt;c, u, inquire, q<sub>3</sub>&gt;]</p> <p><b>explain_game(p<sub>1</sub>)</b>  <u>Parameters:</u> Goal = [explain, p<sub>1</sub>]            Focus_Space = [VISUAL BASIC, VISUAL LANGUAGE, VISUAL C++, OBJECT-ORIENTED LANGUAGE, INHERITANCE, ENCAPSULATION]  <u>Specifications:</u> (B<sub>c</sub>(B<sub>u</sub>(p<sub>1</sub>)) and ¬B<sub>u</sub>(p<sub>1</sub>)) or (B<sub>u</sub>(p<sub>1</sub>) and ¬B<sub>c</sub>(p<sub>1</sub>))  <u>Components:</u> Relevant_Propositions = [q<sub>1</sub>, q<sub>2</sub>, q<sub>3</sub>, q<sub>4</sub>, q<sub>5</sub>]            List_of_Communicative_Acts = [&lt;c, u, inform, q<sub>1</sub>&gt;, &lt;c, u, inquire, q<sub>2</sub>&gt;, &lt;c, u, inquire, q<sub>3</sub>&gt;, &lt;c, u, inquire, q<sub>4</sub>&gt;, &lt;c, u, inquire, q<sub>5</sub>&gt;]</p>
---	---

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

{2} C<sub>[challenge]</sub> 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 (Dimitrova et al., 2001). Suppose that the learner answers:

{3a} U<sub>[justify]</sub> I think that VISUAL BASIC is an OBJECT-ORIENTED LANGUAGE because it contains OBJECTS.

Applying commitment rules and commonsense reasoners, STyLE-OLM will update UCS. At this stage, the *workspace* will contain:

Dialogue history: [<u, c, inform, p<sub>1</sub>>, <c, u, challenge, p<sub>1</sub>>, <u, c, justify, q<sub>4</sub>>]

UCS: [B<sub>u</sub>(p<sub>1</sub>), B<sub>u</sub>(q<sub>4</sub>⇒p<sub>1</sub>), B<sub>u</sub>(q<sub>4</sub>), B<sub>u</sub>(q<sub>1</sub>)]

CCS: [B<sub>c</sub>(B<sub>u</sub>(p<sub>1</sub>))]

DG stack: [explore\_game(p<sub>1</sub>)],

Active DG: explore\_game(p<sub>1</sub>)

Current situation: the user has justified with a proposition that correspond to a possible misclassification pattern.

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 user 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}**  $U_{[justify]}$  I think that VISUAL BASIC is an OBJECT-ORIENTED LANGAUGE because it is a VISUAL LANGUAGE.

Applying commitment rules and commonsense reasoners, STyLE-OLM will update UCS. At this stage, the *workspace* will contain:

Dialogue history: [ $\langle u, c, inform, p_1 \rangle$ ,  $\langle c, u, challenge, p_1 \rangle$ ,  $\langle u, c, justify, q_6 \rangle$ ]

UCS: [ $B_u(p_1)$ ,  $B_u(q_6 \Rightarrow p_1)$ ,  $B_u(q_6)$ ]

CCS: [ $B_c(B_u(p_1))$ ]

DG stack: [ $explore\_game(p_1)$ ],

Active DG:  $explore\_game(p_1)$

Current situation: the user has justified with a proposition that correspond to a possible misclassification pattern.

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 user 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.

After the interaction is terminated, STyLE-OLM applies a mechanism based on modal logic that elicits a resultant user model from the commitment stores. The mechanism is presented in (Dimitrova et al., 2000).

## 6 Evaluation and Future Improvements

A small evaluative study of STyLE-OLM was conducted (Dimitrova, 2001) to

examine the advantages of interactive open learner modelling (Dimitrova et al., 2001) and to assess the robustness of the framework the system is based on. In the context of this paper, the evaluation of the dialogue maintenance mechanism in STyLE-OLM is essential and will be discussed here.

The study involved seven post graduate students from the author's department who interacted with a STyLE-OLM instantiation in a Finance domain (Dimitrova, 2002). The dialogue management was examined by analysing log files and asking the users and a teacher to identify interaction problems by inspecting the dialogue transcripts.

The interactions resulted in building more elaborated models of the learners' cognition. The resultant user models contained more learners' beliefs, less invalid beliefs, and some explanations of learners' misconceptions (Dimitrova et al., 2001). We ought to mention though 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. 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 users 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 users 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).

The evaluation of STyLE-OLM outlined aspects that need further investigation and ought to be addressed in an enhanced framework for interactive diagnosis:

**Using shallow domain expertise.** Often 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 (e.g. Horvitz & Paek, 2001).

**Maintaining coherent diagnostic dialogue.** 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 (e.g. Pulman, 2002). More types of DG types need to be included, such as adaptive explanations (e.g. Bontcheva, 2001), comparison (Milosavljevic, 1997), and error repair (VanLehn et al., 1998).

## 7 Conclusions

This paper presents a framework for maintaining diagnostic dialogues that complies with distinctive characteristics of these dialogues: (1) the major aim of diagnostic interactions is extracting extended models of the users' cognition; (2) the users are actively involved in the interactions influencing the flow of dialogue and the content of the user models; (3) the views of both diagnosticians and diagnosees about the user models are accumulated; (4) mixed initiative is maintained with possibilities for users to interrupt the dialogue and initiate new discussions. We have combined two approaches known as dialogue games, namely, linguistic DGs (Levin & Moore, 1977) to maintain dialogue episodes and logical DGs (Walton, 1984) to enable the participants' symmetrical involvement and to collect their commitments. The framework has been applied in the STyLE-OLM system which simulates interactive open learner modelling. An evaluative study of STyLE-OLM has showed that the interactions meet the requirements of interactive user modelling and has identified two major directions for improving the dialogue framework - employing a shallow domain expertise and improving the dialogue coherence. The author intends to address these aspects in future studies.

Interactive user modelling supports dynamic individual modelling and can be applied in various applications. Possible examples are (to mention a few) new generation e-learning systems capable of adapting to individual learners (e.g. Brusilovsky & Peylo, 2001/2002) novel e-training systems that provide personal e-mentors to conform to the needs of each individual, or personalised e-commerce systems that employ personal e-consultants to enable effective service tailored to the customers' needs (e.g. Aberg & Shahmehri, 2001).

## 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 encouragement and constructive criticism are greatly acknowledged.

## References

- Aberg, J. & Shahmehri, N. (2001). User modelling as an aid for human web assistants. In M. Bauer, P. Gmytrasiewicz & J. Vassileva, *Proceedings of User Modelling 2001* (pp. 201-203). Springer-Verlag, LNCS.
- Bontcheva, K. (2001). Tailoring the content of dynamically generated explanations. In M. Bauer, P. Gmytrasiewicz & J. Vassileva, *Proceedings of User Modelling 2001* (pp. 213-215). Springer-Verlag, LNCS.
- Brusilovsky, P., & Peylo, C. (2001/2002). Special Issue on Adaptive and Intelligent Web-Based Systems. *International Journal of AI in Education*. 12/13.
- Bull, S., Brna, P., & Pain, H. (1995). Extending the scope of student models. *User Modeling and User-Adapted Interaction*, 5(1), 45-65.
- Dimitrova, V. (2001) Interactive open learner modelling. PhD thesis, University of Leeds.
- Dimitrova, V. (2002). STyLE-OLM: Interactive open learner modelling. *International Journal of AI in Education*, vol 13, to appear.

- Dimitrova, V., Brna, P., & Self, J. (2002). The design and implementation of a graphical communication medium for interactive open learner modelling. In S. Cerri, G. Gourderes & F. Paraguacu (Eds.), *Intelligent Tutoring Systems* (pp. 432-441). Springer-Verlag, LNCS.
- Dimitrova, V., Self, J., & Brna, P. (2001). Applying interactive open learner models to learning technical terminology. In M. Bauer, P. Gmytrasiewicz & J. Vassileva, *Proceedings of User Modelling 2001* (pp. 148-157). Springer-Verlag, LNCS.
- Dimitrova, V., Self, J.A., & Brna, P. (2000). Maintaining a jointly constructed student model. In S.A. Cerri & D. Dochev (Eds.), *Artificial intelligence: Methodology, systems, and applications* (pp. 221-231). Berlin: Springer, Lecture Notes of Artificial Intelligence.
- 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.
- 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.
- Horvitz, E., & Paek, T. (2001). Harnessing models of users' 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.
- Kay, J. (1995). The UM toolkit for cooperative user modelling. *User Modeling and User-Adapted Interaction*, 4, 149-196.
- Levin, J., & Moore, J. (1977). Dialogue games: meta-communication structures for natural language interaction. *Cognitive Science*, 1 (4), 395-420.
- 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.
- 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.
- 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*. Available on-line from <http://um.org>.
- 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.
- Paiva, A., & Self, J.A. (1995). TAGUS - a user and learner modelling workbench. *User Modeling and User-Adapted Interaction*, 4, 197-226.
- Pilkington, R. (1999). Analysing educational discourse: the DISCOUNT scheme. *Technical Report #99/2*, Computer Based Learning Unit, University of Leeds.
- Pulman, S. (2002). Relating dialogue games to information states. *Speech Communication*, 36, 15-30.
- Rich, E. (1999). Users are individuals: individualizing user models. *International Journal of Human-Computer Studies*, 51, 323-338.
- Self, J. A. (1990). Bypassing the intractable problem of student modelling. In C. Frasson & G. Gauthier (Eds.), *Intelligent tutoring systems: At the crossroad of artificial intelligence and education*. Norwood, New Jersey: Ablex Publishing Corporation.
- Sowa, J. (1984). *Conceptual structures: Information processing in mind and machine*. MA: Addison-Wesley.
- 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.
- Walton, D. (1984). *Logical dialogue games and fallacies*. University Press of America, Lanham.

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.