Dialectic Approach for Using Viewpoint Discrepancies in Learning

Christiana Panayiotou and Vania Dimitrova

School of Computing,
University of Leeds,
UK
{cpaa,vania}@comp.leeds.ac.uk
http://www.comp.leeds.ac.uk

Abstract. The paper proposes a dialectic approach to exploit discrepancies of viewpoints for learning. The approach is illustrated with an elaborated example. A computational framework of a pedagogical agent capable of interacting with a learner for discussing different viewpoints in the same domain is outlined. The framework employs AI technologies, such as argumentation for defeasible reasoning, situation calculus for contextualized reasoning and dialogue management. The approach can be applied in interactive learning environments to promote awareness, reflection, and conceptual change.

Key words: viewpoints, conceptual discrepancies, argumentation, context, e-learning

1 Introduction

The Semantic Web enables the representation of different conceptualisations in the form of ontologies. However, studies show that conceptualisations may differ between tutors and the resources they prepare [1], as well as between a learner and a tutor [2]. Reasons for this variation can be the intented use of each conceptualisation, the background knowledge of tutors and learners, or the incompleteness of domain ontologies. The awareness of alternative views can bring educational benefits by broadening the perspective of learners.

This paper argues that discrepancies in conceptualisations can be handled constructively to enrich the learning experience in educational systems. We outline an approach where a software agent detects discrepancies in conceptual viewpoints of a learner, tutor, and learning resources, and engages in a dialogue to explore similarities and differences between different viewpoints.

The paper reviews existing approaches for dealing with different viewpoints in learning systems and proposes a dialectic approach for handling viewpoints in educational semantic web applications. The proposed approach will be introduced with the help of an example. Then, the architecture of a dialogue agent that explores different viewpoints in a conversation with a learner will be outlined. We will illustrate the use of AI technologies, such as argumentation for defeasible reasoning, situation calculus for contextualized reasoning and dialogue management, to exploit viewpoints discrepancies in learning.

2 Using Viewpoints in Learning Systems

The first attempts to deal with viewpoints in learning can be traced back to some of the early Intelligent Tutoring Systems (ITSs). Among these, two notable uses of viewpoints are shown in the systems VIPER [3] and DENISE [4]. In VIPER, viewpoints represent different ways of decomposing a domain and provide different interpretations of domain knowledge. However, viewpoints are fixed in advance and refer only to the domain expertise. In contrast, DENISE [4] focuses on student modelling, and considers that the domain model and the learner model may represent different viewpointsA formal way for representing viewpoints in ITS is given in [5] where the viewpoint of an agent a is defined as a triple $Va =< Ba, La, Ma >$ with each element being a subset of the agent's complete belief, logic and meta-logic space, respectively.

While the early ITS research on viewpoints considers different perspectives of the domain and offers representations that distinguish between the tutor's and learner's viewpoints, these projects suffer from two key limitations. Firstly, although the students are considered to have alternative views upon the domain, any deviation from the view of the tutor is considered as a *bug* that needs to be fixed. Secondly, the early ITS systems adopt rather static approaches for dealing with viewpoints, e.g. transmitting the tutor's viewpoint by telling it to the student and assuming that it will overwrite the student's own.

More recent approaches followed in collaborative learning systems which enable the discussion and exchange of different points of view among peers. Based on research in Education which advocates the use of argumentation for constructive learning, collaborative learning systems were implemented to enable and encourage the use of argumentation for joined decision making and sharing of knowledge, e.g. [6, 7]. Empirical evidence from the use of these systems suggests that the exchange and challenge of different viewpoints via argumentation motivates the processes of reflection, articulation and conceptual change. Although these systems aim to sharpen the learner's critical skills, they typically provide very limited analysis of the discussion. They do not model the learners' beliefs during the interaction and do not provide any automatic support to facilitate articulation and clarification of different views about the domain.

Collaborative learning environments have influenced the design of computational approaches for developing intelligent pedagogical agents that support viewpoint clarification. Despite the notable successes, the existing computational approaches do not fully address the problem of identifying and clarifying viewpoints because they do not explore the context in which the views have been formed and ignore what arguments have led the learner/tutor to form a particular position. Moreover, none of these approaches is SW-compliant, so additional work is required to make them ontology-based and to integrate them in educational SW applications, as illustrated in [2].

Proposed Approach. Building on research in dialogue pedagogical agents, Semantic Web, and argumentation, we propose a dialectic approach for exploiting viewpoint discrepancies for learning. The proposed approach caters for the representation of multiple viewpoints of the same domain, treats discrepancies in conceptualisations between the learner and the agent as triggers for dialogue games that clarify different viewpoints and enables the participants to justify their positions via the use of argumentation thus promoting the processes of reflection and articulation.

3 Illustrative Example

We will illustrate our approach with the help of an example that shows how a pedagogical agent A can be integrated in an educational semantic web system, e.g. an adaptive recommender system like OntoAIMS [2]. The agent A has a domain ontology Ω_A representing the main concepts and relations in the domain (we will use here the ontology about Programming Languages from one of the instantiations of STyLE-OLM [8]). The agent recognises that its knowledge can be incomplete and engages in a dialogue to explore different viewpoints.

A **viewpoint** $\mathcal{V}$ is defined as a structure $\mathcal{V} \equiv < s, p, B, \Gamma >$ where: s denotes the source of the viewpoint (e.g. the learner, the agent, or an existing learning resource), p denotes the position of the viewpoint represented with a proposition (e.g. *'VB is OOL'*), B is a set of beliefs of s related to p (e.g. *'All OOL have objects'*), and Γ is a set of arguments of s supporting the position p.

Assume that the learner L is a distance student attending an introductory course on programming languages. She receives most of her learning material online through a list of online resources aiming to enable her to enhance her knowledge on the topic. L comes across a learning resource that states *'Visual Basic is an object-oriented language'*. She believes that an object-oriented language (OOL) needs to satisfy the property of encapsulation but she does not know whether Visual Basic (VB) has this property. Further, she is almost sure that she has read somewhere else that Visual Basic is not an object-oriented language. The learner is confused and invokes a discussion with the pedagogical agent A in order to clarify the position she encountered in a resource R with url URI_1. The agent A accepts the initiated discussion and starts a *viewpoint elicitation game* to find out more about the viewpoint expressed in the resource.

(1) **L:** Let us Discuss the statement in Resource: URI_1: *VB is an OOL*.
(2) **A:** What is the explanation given in the Resource URI_1 for *VB is an OOL*?
(3) **L:** The resource URI_1 states that *VB is an OOL* because *VB contains Objects*.

Based on the above dialogue, the viewpoint $\mathcal{V}_R$ of the resource R is derived:

$\mathcal{V}_R \equiv$ **Viewpoint of resource** R	
Source s_R	R
Position p_R	`instance_of(VB,OOL)`
Belief set B_R:	
	`contain(VB, objects)`
Argument set Γ_R:	
$claim_1$	`instance_of(VB,OOL)`
$datum_1$	`contain(VB, objects)`
$warrant_1$	$\forall X$: `contain(X, objects)` $\Rightarrow$ `instance_of(X,OOL)`

To find out whether the learner agrees with the viewpoint of the resource, the agent A initiates an *explore viewpoint game*. The interaction continues with:

(4) **A:** So do you agree that *VB includes objects*?
(5) **L:** Yes..I agree that *VB includes objects*
(6) **A:** Do you agree that a *PL is OOL* because it *contains objects*?
(7) **L:** I disagree with this.
(8) **A:** OK, do you agree that *contains objects* is necessary for a PL to be OOL?
(9) **L:** Yes.
(10) **A:** What other condition is needed for a PL to be OOL?
(11) **L:** OOL have the *property of 'encapsulation'*
(12) **A:** So, you think that *IF a PL contains objects AND PL has the property of encapsulation THEN PL is an OOL*
(13) **L:** Yes, that's it!
(14) **A:** Therefore, you think that *IF VB has the property of encapsulation THEN VB is an OO Language*
(15) **L:** Yes.

Initially, A has no information about the viewpoint $\mathcal{V}_L$ of the learner apart from assuming that L is confused about p_R. In the above dialogue, move **(4)** checks whether L supports $\Gamma_R : datum_1$, i.e. the datum in the only argument in $\mathcal{V}_L$, while moves **(6)** and **(8)** clarify whether L supports $\Gamma_R : warrant_1$. Following L's dialogue moves, the agent discovers that there is a discrepancy between viewpoints $\mathcal{V}_L$ and $\mathcal{V}_R$, and attempts to explore the difference between these viewpoints. Move **(10)** asks for an additional belief, based on which a rule is abducted and then checked in move **(12)**. Finally, in move **(14)** the agent confirms the claim of the learner's argument. Hence, the agent derives the following viewpoint of the learner:

$\mathcal{V}_L \equiv$ **Viewpoint of Learner** L	
Source s_L	L
Position p_L	`? instance_of(VB,OOL)`
Belief set B_L**:**	

	`contain(VB, objects)`
	`contain(OOL, objects)`
	`has_property(OOL, encapsulation)`

Argument set Γ_L**:**	
$claim_1$	`has_property(VB,encapsulation)` $\longrightarrow$ `instance_of(VB,OOL)`

$datum_1$	`contain(VB,objects)`
	`contain(OOL,objects)`
	`has_property(OOL,encapsulation)`

$warrant_1$	$\forall X$: `contain(X,objects)` $\wedge$ `has_property(X,encapsulation)` $\Rightarrow$ `instance_of(X,OOL)`

The question mark in p_L shows that L is undecided about this position, and $\Gamma_L : claim_1$ indicates the condition needed for the learner to support p_L.

To continue the dialogue, the agent compares its own viewpoint with that of the leaner. The agent's viewpoint $\mathcal{V}_A$ is derived from the ontology Ω_A that can be incomplete. Ω_A represents that *'All OOL have the property of inheritance'* and *'Java is an OOL'* but does not have information about VB apart from it being a programming language. Hence, the viewpoint of the agent A is:

$\mathcal{V}_A \equiv$ **Viewpoint of agent** A	
Source s_A	A
Position p_A	`? instance_of(VB,OOL)`
Belief set B_A**:**	

	`instance_of(VB, Programming Language)`
	`instance_of(Java, OOL)`
	`contain(VB, objects)`
	`contain(OOL, objects)`
	`has_property(OOL, inheritance)`
	`class(OOL)`

Argument set Γ_A**:**	
claim	`has_property(VB,inheritance)` $\rightarrow$ `instance_of(VB,OOL)`

datum	`contain(VB,objects)`
	`contain(OOL,objects)`
	`has_property(OOL,inheritance)`

warrant	$\forall X$: `contain(X,objects)` $\wedge$ `has_property(X,inheritance)` $\Rightarrow$ `instance_of(X,OOL)`

Similarly to $\mathcal{V}_L$, $\mathcal{V}_A$ has a question mark in p_A to indicate that A is undecided about this position. $\Gamma_A : claim_1$ indicates the condition needed for the agent to support p_A and $\Gamma_A : warrant_1$ shows the rule which A holds in order to derive that a programming language is object-oriented. The agent notes that the viewpoints $\mathcal{V}_L$ and $\mathcal{V}_A$ have some similarities and differences. Both A and L are undecided about the position *'VB is OOL'* and share some common beliefs. However, A and L have built arguments that support different claims: $\Gamma_L : claim_1 \neq \Gamma_A : claim_1$. The agent cannot accept $\Gamma_L : warrant_1$ because it contradicts with $\Gamma_A : warrant_1$ and A believes that the property of inheritance is necessary. Thus, A initiates an *explore-inform viewpoint game* to find out whether L is aware of the property of inheritance, and if this is not the case, A employs didactic tactics to give guidance to L. This is illustrated in the following dialogue:

(16) **A:** I disagree that IF a PL contains objects AND PL has the property of encapsulation THEN PL is an OOL.
(17) **A:** Let us discuss this.
(18) **A:** Do you think that Java is an OOL?
(19) **L:** Yes, I think that Java is an OOL.
(20) **A:** Can you tell me a property of Java other than encapsulation?
(21) **L:** I can't think of anything else...
(22) **A:** Are you aware of the property of inheritance?
(23) **L:** No.
(24) **A:** Inheritance means that a class obtains the characteristics of it superclasses. All OOL have the property of inheritance. Java has the property of inheritance.

The agent expresses its disagreement with $\Gamma_L : warrant_1$ in move **(16)**. Instead of simply informing the learner that inheritance is a necessary property, A introduces a specialisation of the rule in $\Gamma_A : warrant_1$ and checks whether L the learner can derive the need for inheritance from the specialisations with moves **(18)** and **(20)**. When A discovers that L is unaware that *'Java has inheritance'*, it checks in move **(22)** whether L knows anything about inheritance. Because L is unaware of inheritance, A composes an explanation by combining the statements about inheritance extracted from Ω_A. This ends the dialogue game and updates $\mathcal{V}_L$ to include the new belief about Java. If the answer in move **(23)** was positive, the discussion would have continued with checking whether *'VB has the property of inheritance'* and might have led to updating $\mathcal{V}_L$ or $\mathcal{V}_A$ accordingly.

The above example illustrates a dialogue that explores three viewpoints from a learning resource, the learner, and the tutoring agent. It also illustrate how disagreement in viewpoints can be exploited for learning about the domain. Although the agent is equipped with the domain knowledge to form a viewpoint, this viewpoint is not imposed on the learner and she is only informed of the agent's opinion about inheritance after the viewpoint discrepancies are explored. The agent allows the learner to form her own opinion making her aware of all the relevant knowledge, and the point of view of the agent.

4 Proposed Framework

The goal of our research is to develop a computational framework for the design of a tutoring agent A capable of engaging in discussions to elicit viewpoints and explore similarities and differences between them, as illustrated in the above example. We will outline here the main architecture of our framework and will define its main components. The proposed architecture is illustrated in figure 1.

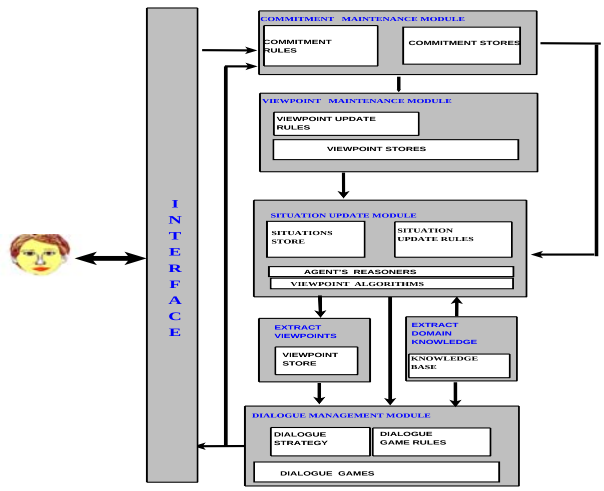

Fig. 1. Proposed Architecture of a Framework for Dialectic Viewpoints Handling

Interface. We assume that both the learner and the tutoring agent are provided with an appropriate interface to compose their utterances that express dialogue moves. In line with existing computational approaches, e.g. [9, 10, 8, 2], we assume that the interaction is restricted to the use of predefined moves where each move is associated with several possible sentence openers. A set of moves and their corresponding sentence openers are illustrated in Section 3.In addition, we assume that the interface provides an appropriate way for the dialogue participants to compose the propositions of their dialogue moves, e.g.

by using structured sentences or graphical statements [2]. Hence, a move m is defined as a tuple $m =< n, a, t, \varphi >$, representing its unique identifier which is a number n, the agent a who produces the move, the move type t that is linked to possible sentence openers, and the statement φ. To make a statement that a proposition p is valid in a particular context C we will use the predicate $ist(C, p)$ [11]. For instance, the first two moves in the example above express statements about the resource R and are defined as follows:

$m_1 =< 1, L, informDiscuss, ist(R, instance\_of(VB, OOL) >$
$m_2 =< 2, A, questionExplore, ist(R, instance\_of(VB, OOL) >$

Commitment maintenance. The beliefs of both participants derived from the dialogue are stored in commitment stores, and are used to compose the viewpoints or to plan the dialogue. Similarly to [8, 2], we employ commitment rules to establish the beliefs to which the participants of the dialogue are committed by taking into account the current dialogue move and the dialogue history. The agent's commitments are also derived from its ontology Ω_A, see Section 3.

Viewpoint maintenance. The viewpoints derived from the dialogue are stored in viewpoint stores. The definition of viewpoints given in Section 3 enables us to compare two viewpoints and identify similarities and differences between them, as shown in the illustrative example above. In addition, the maintenance of viewpoints includes a set of operations over the viewpoint stores to add, delete, update, and revise viewpoints.

Situation update. Based on the commitment stores and the viewpoint stores, the agent obtains information about the current situation which is used for planning the dialogue and update of the existing viewpoints. Situation update is performed after each dialogue move to encounter the changes it brings. For instance, a situation can present that there are discrepancies in two viewpoints (e.g. the situation after move **(15)** in the example will represent that $\mathcal{V}_L \neq \mathcal{V}_A$) or that there is insufficient information about a particular viewpoint (e.g. the situation after move **(3)** in the example will represent that $\mathcal{V}_L$ is still empty).

Dialogue management. The dialogue is organised as a sequence of dialogue games which in turn are sequences of dialogue moves. Each game pursues a particular goal and is initiated and terminated when certain situations occur. For example, the dialogue game in moves **(4)-(15)** in Section 3 has the goal to *explore the viewpoint* $\mathcal{V}_L$ by following viewpoint $\mathcal{V}_R$, and is initiated when there is sufficient information about $\mathcal{V}_R$ and no information about $\mathcal{V}_L$. The dialogue management checks the current situation and initiates or terminates dialogue games, accordingly.

5 Conclusion

The paper proposed a dialectic approach for exploiting viewpoint discrepancies in interactive learning environments. The key characteristics of our approach are that: (a) viewpoints are composed of positions, relevant beliefs and supporting arguments; (b) incompleteness of or discrepancies between viewpoints are used as triggers for argumentative dialogue games; (c) viewpoints represent statements valid in particular contexts, which is explored during the interaction; (d) while discrepancies and similarities between viewpoints are explored, changes in viewpoints are not imposed; (e) viewpoints are accumulated in viewpoint stores and can be shown to a learner to promote domain awareness or to a human tutor to highlight problems with learning resources or existing ontologies.

Currently, we are working on the formal description of our framework by employing argumentation dialogue frameworks based on situational calculus and dialogue games. At the same time, we are developing a Prolog-based proof of concept prototype to illustrate and validate the main definitions. The prototype uses a sample domain ontology about Programming Languages and takes as input Prolog-based definitions of dialogue moves (i.e. it assumes that the moves have been recognised). Once the framework is developed and tuned by using the prototype, we plan to deploy it in an existing educational semantic web system, e.g. OntoAIMS [2], to help learners make links between learning resources and become aware of different perspectives of content and ontologies.

References

1. Dicheva, D., Dichev, C.: Confronting some ontology-building problems in educational topic map authoring. In: Proceedings of Workshop on Semantic Web for Educational Adaptive Hipermedia at AH2006. (2006)
2. Denaux, R., Dimitrova, V., Aroyo, L.: Integrating open user modeling and learning content management for the semanticweb. In: 10th Int. Conference on User Modeling. (2005) 18–22
3. Moyse, R.: A structure and design method for multiple viewpoints. Journal of Artificial Intelligence in Education **3(2)** (1992) 207–233
4. Nichols, D.: Intelligent Student Systems: an Application of Viewpoints to IntelligentLearningEnvironments. PhD thesis, Computing, Lancaster University (1993)
5. Self, J.: Formal approaches to student modelling. In Greer, J., McCalla, G., eds.: Student Modelling: The Key to Individualised Knowledge-based Instruction, Springer-Verlag (1994) 295–352
6. Suthers, D., Weiner, A., Connelly, J., Paolucci, M.: Belvedere: Engaging students in critical discussion of science and publicpolicyissues. In: AIED. (1995)
7. Ravenscroft, A., McAlister, S.: Digital games and learning in cyberspace: a dialogical approach. E-Learning **3**(1) (2006) 37–50
8. Dimitrova, V.: Style-olm: Interactive open learner modelling. International Journal of Artificial Intelligence in Education **13** (2003) 35–78
9. Soller, A.: Computational modeling and analysis of knowledge sharing in collaborative distance learning. User Modeling and User-Adapted Interaction: The Journal of Personalization Research **14** (2004) 351–381
10. Tedesco, P.: Marco: Building an artificial conflict mediator to support group planning interactions. International Journal of Artificial Intelligence in Education **13** (2003) 117–155
11. Guha, R., McCool, R., Fikes, R.: Contexts for the semantic web. In McIlraith, S.A., Plexoudakis, D., van Harmelen, F., eds.: Proceedings of the International Semantic web Conference, Lecture Notes in Computer Science (2004)