AMANDA - An Intelligent Dialog Coordination Environment

Marco A. Eleuterio[1]
PUC-PR/UTC
marcoa@hds.utc.fr

Jean-Paul Barthès
UTC, France
barthes@utc.fr

Flávio Bortolozzi
PUC-PR, Brazil
fborto@ppgia.pucpr.br

Celso A. Kaestner
PUC-PR, Brazil
kaestner@ppgia.pucpr.br

[1] Sponsored by CNPq, Brazil

Abstract

This paper describes AMANDA[2] - an intelligent system intended to coordinate collective dialog sessions in distance learning environments. The overall objective of AMANDA is to help tutors achieve better results from group discussions and improve knowledge transfer among the participants. This is done by integrating the collective dialog as a disciplined and well-coordinated activity in distance learning situations. For this purpose, the dialog is represented as an argumentation tree, a structured collection of questions, alternatives and arguments which evolves along sequential dialog cycles. The intelligent behavior of the system is due to its coordination actions taken in response to reasoning over the dialog. We describe how AMANDA coordinates the dialog process by generating a sequence of dialog cycles based on a set of coordination parameters. In this paper we briefly describe AMANDA's functional modules, internal structures and coordination algorithms. The knowledge models that support system reasoning are described, as well as our practical experience in domain modeling. We have tested the system in actual training situations, for which we chose a test course and modeled the corresponding domain knowledge. Although some modules of the system are still under development, specially those related to semantic reasoning, we discuss the application of semantic parameters and identify some techniques which may improve the coordination algorithm.

[2] AMANDA - Agent de Modélisation et ANalyse de Dialogues Argumentés - is a joint R&D effort between the Pontifical University of Paraná, Brazil (PUC-PR), the Technology University of Compiègne, France (UTC) and their respective partners Siemens Telecomunicações, Brazil and Cegos, France.

1. Introduction

Collaborative learning is about promoting knowledge transfer among the apprentices through a series of learning interactions. We recall a well-known knowledge management theory (Nonaka, 99) in which a knowledge transfer environment is composed of four knowledge-transfer spaces, namely the *socialization* space, the *dialoguing* space, the *systematization* space and the *internalization* space. In each of these spaces, a specific implicit↔explicit knowledge conversion occurs. By applying this approach to a collaborative learning environment, as detailed in (Eleuterio, 1999a), we categorize AMANDA as a *dialoguing* space in which the articulation of knowledge is the key for knowledge transfer. In traditional distance learning environments, this dialoguing space is normally implemented by discussion forums.

Our experience with discussion forums in Eureka (Eleuterio, 1999b), a web-based environment developed in partnership with Siemens and extensively used in academic and professional training contexts, shows that traditional discussions forums often fail to promote group learning. They either grow two much to be efficiently followed up by the tutor or suffer from the lack of participation and coordination. Similar problems are described in (Leary, 1998) when identifying common problems in discussion groups of knowledge management systems. From our observations, the two main reasons why discussion forums often fail are (i) the lack of discipline due to the poor integration of the discussion process into the regular activities of the course and (ii) the lack or articulation and coordination of the discussion.

With the purpose of overcoming the identified problems, we propose a dialog framework that covers both aspects, i.e. automatically coordinates the dialog while engaging the participants by generating dialog activities.

We identify three main differences between AMANDA and a traditional discussion forum. Firstly, the presence of domain models in AMANDA's architecture enables a certain degree of semantic reasoning over the dialog. Secondly, its coordination mechanism relieves the tutor from time-consuming coordination tasks, such as finding relations between users' inputs, measuring the degree of commitment of the participants, detecting disagreement topics and measuring the coverage of discussion topics. Thirdly, the system manages the dialog by generating discussion cycles, in which the participants express their supporting and opposing ideas in relation to another participant's input, thus creating a suitable context for the articulation and confrontation of ideas and points of view.

The proposed coordination mechanism allows various degrees of knowledge representation without impairing dialog control. It means that, if the system has no knowledge models, it can coordinate the dialog as well, gracefully degraded, by considering only structural parameters. This is possible due to the separation between structural and semantic aspects in the coordination mechanism (see section 4). This separation allows applying AMANDA to situations where knowledge modeling is neither feasible, e.g. open domain discussions, nor desirable, e.g. short-term courses.

Merging Two Complementary Approaches

Tutorial dialog has been subject of important research efforts, such as the CoLLeGE architecture (Ravenscroft & Pilkington, 2000) which analyzes dialog moves, conceptual changes and world models as the basis of the dialog process. Such work deeply inspects the tutor-apprentice interaction, but doesn't give much emphasis on the *collective* aspect of the dialog. On the other hand, the argumentative discourse environment (Karacapilidis, 1998) describes an argumentation framework applied to multi-agent decision making, which is fully devoted to formalize argumentative discourses. Our objective is to merge both approaches, which seem to be complementary, in a single dialog coordination system applied to collaborative distance learning environments.

2. System Overview

AMANDA is an autonomous domain-independent intelligent dialog coordination system applied to collective discussions. By *domain-independent* we mean that domain-dependent behavior is achieved by providing the corresponding domain knowledge models. By *intelligent coordination system* we mean that AMANDA takes coordination actions by reasoning over the structure and the semantics

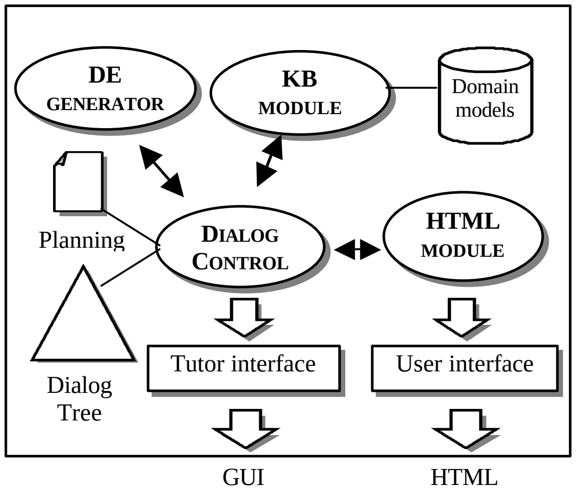


Figure 1a: System overview

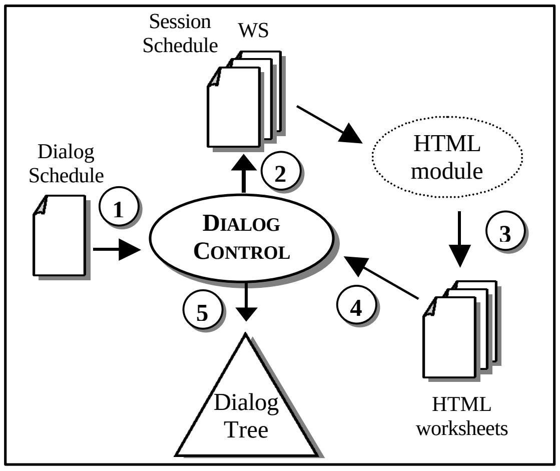


Figure 1b: Dialog control – simplified

of the dialog. The *autonomous* feature of AMANDA is due to its capability of coordinating the dialog without direct interference of the human tutor. Figure 1a shows the main modules of the system and the paragraphs below describe the modules, structures and processes that take part in the dialog coordination.

2.1. Dialog Control Module

This module is AMANDA's central coordination mechanism. Its principle is to organize the dialog in sequential periods called *sessions*, each one representing a time interval in which a certain number of discussions will be carried on. During each session, the system triggers a number of *dialog cycles* in order to update the dialog tree with input from the participants.

In the setup stage, the Dialog Control module reads the dialog schedule ①, where all sessions are described. It then repeatedly generates dialog cycles by producing worksheets ③ until a satisfactory degree of agreement is achieved. Each time the system receives input from the participants ④, the Dialog Control module analyzes and updates the dialog tree ⑤ and decides upon producing a new cycle or closing the dialog. The items below detail the structures handled by the Dialog Control module.

2.1.1. Dialog Planning

The dialog planning is represented by the *dialog schedule* and the *session schedule*.

Dialog Schedule

The dialog schedule is the overall planning of the dialog. It specifies the dialog sessions, the corresponding start/end dates and the respective domain of discourse (Figure 2).

Session	SD	ED	DS (domain of discourse)
S-1	Sd	Ed	(c1 … cm)
S-2	Sd	Ed	(c1 … cn)
			:
S-n	Sd	Ed	(c1 … cp)

S-n: the nth session of the dialog	DS: a set of concepts from the domain ontology
SD: start date; ED: end date	

Figure 2: The dialog schedule Session

Session Schedule

The *session schedule*, on the other hand, is a dynamic structure automatically produced and updated by the system during a given session (Figure 3). Each entry of the session schedule is a *dialog cycle* which specifies a *dialog task* to each participant. A dialog task is the set of all nodes from the dialog tree (see item 2.1.2) which are assigned to the same participant at a certain dialog cycle.

A dialog task is represented by a *worksheet assignment* of the type (id, list-of-WEs), in which a list of worksheet elements (we) is assigned to a particular participant (id). Worksheet elements map directly to specific nodes of the dialog tree.

Cycle	SD	ED	WS assignment
C-1-x	Sd	Ed	((id (we-y-1 … we-y-n)) …)
C-2-x	Sd	Ed	((id (we-y-1 … we-y-n)) …)
			: :
C-n-x	Sd	Ed	((id (we-y-1 … we-y-n)) …)

C-n-x:	the nth dialog cycle of session x	WS:	worksheet, a set of ordered pairs of the type (id-x we-y)
SD:	start date	id:	the ID of the participant
ED:	end date	we:	worksheet element

Figure 3: The session schedule

2.1.2. Dialog Tree

The dialog tree, shown in Figure 4 is the structure that represents the dialog. Its internal nodes can be of five types: DIALOG, SESSION, DE, ALT and ARG. Its internal structure was adapted from the argumentation model (Karacapilidis, 1998). The paragraphs below describe each type of node and their corresponding relations to the dialog process.

DIALOG node

The DIALOG node is the uppermost node of the tree. It contains a reference to a number of dialog sessions. When a dialog is created, this node is initialized with the information contained in the dialog schedule (Figure 2.a).

SESSION node

The SESSION node is the uppermost node of a dialog session. Dialog sessions are intended to organize the discussion into separate time periods, each one assigned to a certain *domain*

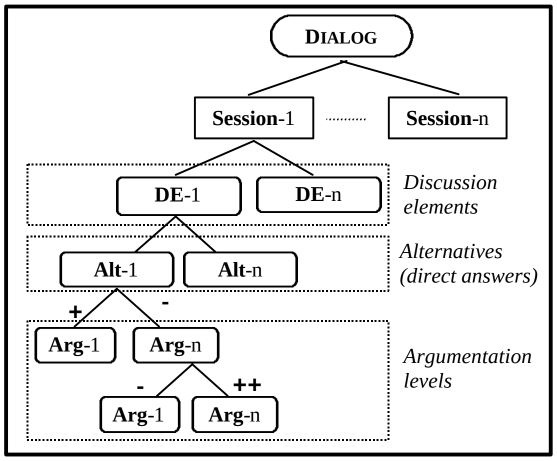


Figure 4: The dialog tree

of discourse. The SESSION node contains a reference to all discussion elements (DEs) which are scheduled for discussion within this session.

DE node
The DE node represents a discussion element, i.e. a natural language question that will originate a specific discussion. Examples of DEs are: "which are the elements of a training budget?" or "what types of connection elements exist in a computer network?".

A DE can be classified as a *content-expected interrogative speech act* (Porayska-Pompa, 2000), for which we expect an answer with a certain "content" as response. According to the argumentation model of (Karacapilidis, 1998), a DE node is an *issue* to be debated.

ALT node
The ALT node is an answer to a question. It is an *alternative* response to a certain DE. The answer contained in an ALT node is the "content" expected by its corresponding DE node. In Karacapilidis' model, an ALT node is a *position* over an *issue*.

ARG node
The ARG node, or *argumentation* node, represents a *supporting* or *opposing* reaction from a given participant over a dialog element placed by another participant. An ARG node can either refer to an ALT node or to another ARG node.

Argumentation nodes are key elements of the dialog. When analyzed as a whole, they represent the level of collective agreement over a given position. Each ARG node conveys a supporting or opposing intention, or *polarity*. This intention is expressed by four levels: *total agreement* (++), *partial agreement* (+), *partial disagreement* (-) and *total disagreement* (--).

A substantial coordination effort of AMANDA is concentrated in analyzing the effects of the ARG nodes over the dialog tree (more details in item 4).

2.1.3. Dialog Control Interface

The Dialog Control module has a graphical interface which allows us to view the dialog tree and perform editing and follow-up functions over the dialog. This interface, primarily designed to follow up the dialog, can also be used to simulate dialog situations and evaluate the coordination algorithms.

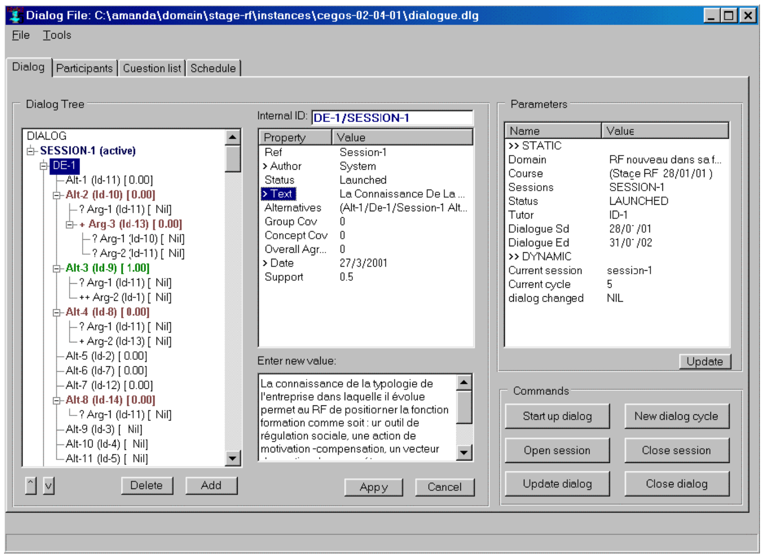
Figure 5 shows the Dialog Control interface. It allows to (i) view the dialog tree, (ii) edit its nodes, (iii) view the internal parameters of the dialog and (iv) simulate a dialog by means of control buttons.

2.2. KB Module

This module is responsible for managing the knowledge model and providing semantic parameters to the Dialog Control module. The central knowledge representation is the *domain ontology*, but other structures may be added, such as the domain task structure. The KB module evaluates the dialog from the semantic point of view, by calculating a certain number of parameters, such as the semantic proximity between two text-based messages, the conceptual distance between ontology concepts or the conceptual coverage of a certain dialog session.

2.2.1. Domain Models

AMANDA requires domain models to perform semantic reasoning over the dialog. In order to enable different types of domain models to be "plugged" into the KB module, we decided to use an ontology-centered approach. This allows to build various models, such as conceptual maps and task structures, which refer to the ontology concepts when applicable.



Dialog File: C:\amanda\domain\stage-rf\instances\cegos-02-04-01\dialogue.dlg

File Tools

Dialog | Participants | Cuestion list | Schedule

Dialog Tree

```
DIALOG
SESSION-1 (active)
  DE-1
    Alt-1 (Id-11) [0.00]
    Alt-2 (Id-10) [0.00]
      ? Arg-1 (Id-11) [ Nil]
      + Arg-3 (Id-13) [0.00]
        ? Arg-1 (Id-10) [ Nil]
        ? Arg-2 (Id-11) [ Nil]
    Alt-3 (Id-9) [1.00]
      ? Arg-1 (Id-11) [ Nil]
      ++ Arg-2 (Id-1) [ Nil]
    Alt-4 (Id-8) [0.00]
      ? Arg-1 (Id-11) [ Nil]
      + Arg-2 (Id-13) [ Nil]
    Alt-5 (Id-2) [0.00]
    Alt-6 (Id-7) [0.00]
    Alt-7 (Id-12) [0.00]
    Alt-8 (Id-14) [0.00]
      ? Arg-1 (Id-11) [ Nil]
    Alt-9 (Id-3) [ Nil]
    Alt-10 (Id-4) [ Nil]
    Alt-11 (Id-5) [ Nil]
```

Internal ID: DE-1/SESSION-1

Property	Value
Ref	Session-1
> Author	System
Status	Launched
> Text	La Connaissance De La ...
Alternatives	(Alt-1/De-1/Session-1 Alt...
Group Cov	0
Concept Cov	0
Overall Agr...	0
> Date	27/3/2001
Support	0.5

Enter new value:

La connaissance de la typologie de l'entreprise dans laquelle il évolue permet au RF de positionner la fonction formation comme soit : ur outil de régulation sociale, une action de motivation -compensation, un vecteur

Parameters

Name	Value
>> STATIC	
Domain	RF nouveau dans sa f...
Course	(Stage RF 28/01/01)
Sessions	SESSION-1
Status	LAUNCHED
Tutor	ID-1
Dialogue Sd	28/0˙/01
Dialogue Ed	31/0˙/02
>> DYNAMIC	
Current session	session-1
Current cycle	5
dialog changed	NIL

Figure 5: The Dialog Control interface

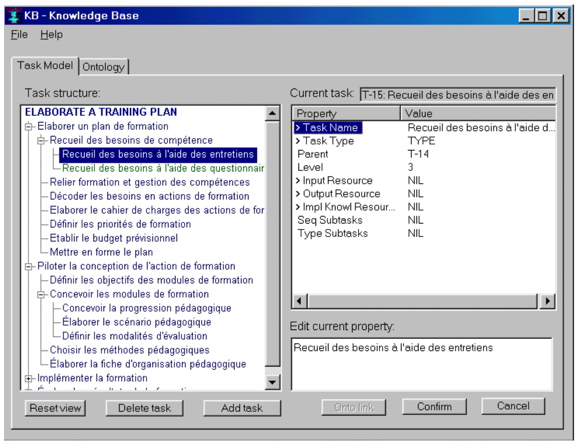
Domain Ontology

The domain ontology is AMANDA's central knowledge representation. Its role is to organize domain concepts so as to enable reasoning. Apart the various definitions found in the knowledge representation literature, it is a consensus that ontologies are conceptual models that explicit the nature of the concepts. The basic type of ontology, the "terminological ontology", or "level 1" ontology (Mizoguchi, 2000), contains primarily is-a links. In some cases, formal definitions are needed to completely reason over a concept. In such cases, more powerful ontologies, like interpretable or executable ontologies, are required.

In our system, since the ontology is used mainly for terminological purposes, we adopted a simple structure which organizes concepts by means of *is-a* and *part-of* links. We decided to merge is-a and part-of links for practical reasons. We were faced with situations in which a concept would be better represented by a part-of decomposition than by a taxonomy relation. In fact, in certain domains, the use of part-of links is the only way to construct ontologies, as in the case of the PLINIUS project (Van der Vet & Mars, 1998). However, depending on the rigor demanded by the ontology application, merging is-a and part-of links may result in tangled hierarchies and confuse the reasoning mechanisms. A formal approach to this problem is described in (Guarino, 2000).

Task Structure

Due to the inherent task-oriented nature of the test course, we used a *task structure* as a complementary knowledge model. It represents the decomposition of a task by means of two types of links: the *sequence* link and the *type* link. *Sequence links* decompose a complex task in a sequence of more detailed sequential subtasks, while *type links* specify different methods of performing a certain task. A detailed description of task structures can be found in (Chandrasekaran, 1992) and (Decker, 1995). Figure 6 shows the Task Model section of the KB interface.



KB - Knowledge Base

File Help

Task Model | Ontology

Task structure:

```
ELABORATE A TRAINING PLAN
Elaborer un plan de formation
  Recueil des besoins de compétence
    Recueil des besoins à l'aide des entretiens
    Recueil des besoins à l'aide des questionnair
  Relier formation et gestion des compétences
  Décoder les besoins en actions de formation
  Elaborer le cahier de charges des actions de for
  Définir les priorités de formation
  Etablir le budget prévisionnel
  Mettre en forme le plan
Piloter la conception de l'action de formation
  Définir les objectifs des modules de formation
  Concevoir les modules de formation
    Concevoir la progression pédagogique
    Élaborer le scénario pédagogique
    Définir les modalités d'évaluation
  Choisir les méthodes pédagogiques
  Élaborer la fiche d'organisation pédagogique
Implémenter la formation
```

Current task: T-15: Recueil des besoins à l'aide des en

Property	Value
> Task Name	Recueil des besoins à l'aide d...
> Task Type	TYPE
Parent	T-14
Level	3
> Input Resource	NIL
> Output Resource	NIL
> Impl Knowl Resour...	NIL
Seq Subtasks	NIL
Type Subtasks	NIL

Edit current property:

Recueil des besoins à l'aide des entretiens

Figure 6: KB interface – *Task Model* section

2.3. DE Generator

This module produces natural language questions based on the available knowledge models. Questions are generated by the system in order to include a given domain topic to the dialog. In practice, this is done to move the focus of the dialog to a desired sub-domain. The content of the questions are based on the links and concepts available in the knowledge models.

Suppose an ontology in the "computer network" domain containing an *is-a* link from the concept <*connection element*> toward the concepts <*hub*> and <*router*>. The semantics of this relation is: *"hubs and routers are types of connection elements"*. For the DE Generator, this link would produce a sentence of the type "what differs *hubs* from *routers* since both are *connection elements?*". This sentence conveys a pre-defined intention to find out the *identity criteria*, or a distinguishing property, between two concepts belonging to the same parent. We could generalize this principle by stating: *"if there is a taxonomic distinction between two concepts, there must be a set of properties capable to distinguish them"* (Guarino, 2000). For each type of semantic relation contained in the knowledge models, we can define a set of generic principles that can be used for sentence generation.

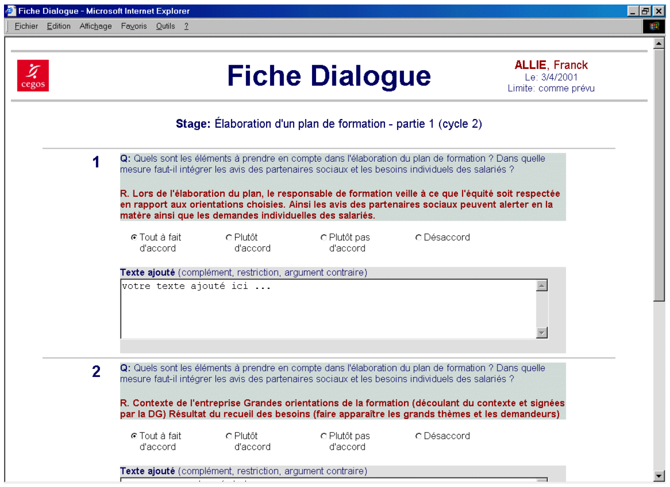
As in the propositions of (Ravenscroft, 2000), the sentences produced by the DE Generator carry a specific intention in the discourse. In our case, they are meant to investigate the domain along five different axes, each one assigned to a specific semantic link of the knowledge model. The ontology contributes with two axes: (i) the nature of the concepts (*is-a* links) and (ii) the elements of a composed concept (*part-of* links). The task model contributes with the remaining three axes: (i) the use of the concepts by a certain task (*resource* link); (ii) the decomposition of a complex task into sub-tasks (*sequence* link); and (iii) different ways of performing a task (*type* link). Each of these axes maps to a set of *sentence structures* of the type shown in the example above. The DE Generator can thus be considered the *linguistic level* of the knowledge models.

2.4. The HTML Module

This module is responsible for the interface between AMANDA and the participants of the dialog. This is done by the dynamic generation of *worksheets* in HTML format (see figure 7).

These worksheets are accessed by the participants, filled in and sent back to the system. Once the worksheets are returned, the Dialog Control module updates the dialog tree.

The HTML module was implemented by a PHP script running on an HTTP server. The communication between the HTML module and the Dialog Control module (see Figure 1.b ④) is done by intermediate files.



Fiche Dialogue

ALLIE, Franck
Le: 3/4/2001
Limite: comme prévu

Stage: Élaboration d'un plan de formation - partie 1 (cycle 2)

1 **Q:** Quels sont les éléments à prendre en compte dans l'élaboration du plan de formation ? Dans quelle mesure faut-il intégrer les avis des partenaires sociaux et les besoins individuels des salariés ?

R. Lors de l'élaboration du plan, le responsable de formation veille à ce que l'équité soit respectée en rapport aux orientations choisies. Ainsi les avis des partenaires sociaux peuvent alerter en la matère ainsi que les demandes individuelles des salariés.

[✓ Tout à fait d'accord] Plutôt d'accord Plutôt pas d'accord Désaccord

Texte ajouté (complément, restriction, argument contraire)

votre texte ajouté ici ...

2 **Q:** Quels sont les éléments à prendre en compte dans l'élaboration du plan de formation ? Dans quelle mesure faut-il intégrer les avis des partenaires sociaux et les besoins individuels des salariés ?

R. Contexte de l'entreprise Grandes orientations de la formation (découlant du contexte et signées par la DG) Résultat du recueil des besoins (faire apparaître les grands thèmes et les demandeurs)

[✓ Tout à fait d'accord] Plutôt d'accord Plutôt pas d'accord Désaccord

Texte ajouté (complément, restriction, argument contraire)

Figure 7: Worksheets in HTML format

3. The Dialog Process

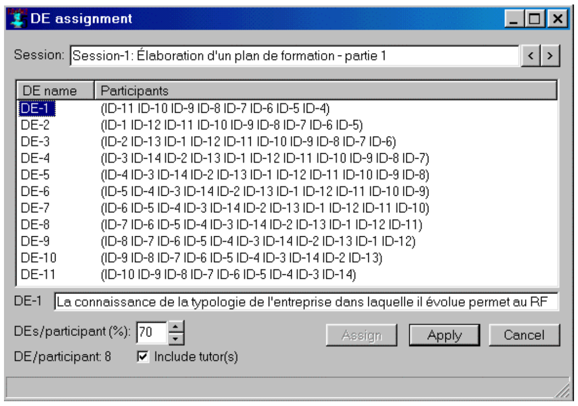
This item explains how AMANDA starts and conducts the dialog process, as well as the related algorithms.

3.1. Dialog Setup

The dialog starts with the creation of a *session schedule* based on the available *dialog schedule* (Figures 2 and 3). Once the dialog session is established, i.e. the SESSION node and the related DE nodes are created, the system can trigger the first dialog cycle.

3.2. First Dialog Cycle

The first dialog cycle, identified as the ALT level in the dialog tree, is intended to distribute the DEs among the participants. To do so, AMANDA takes the set of DEs, as well as the set of participants, and executes the DE-assignment algorithm. This algorithm generates DE assignments of the type (DE, list-of-ids) and can be parameterized according to the desired load of DE/participant and the presence/absence of the tutor(s) in the discussion (see Figure 8).



Session: Session-1: Élaboration d'un plan de formation - partie 1

DE name	Participants
DE-1	(ID-11 ID-10 ID-9 ID-8 ID-7 ID-6 ID-5 ID-4)
DE-2	(ID-1 ID-12 ID-11 ID-10 ID-9 ID-8 ID-7 ID-6 ID-5)
DE-3	(ID-2 ID-13 ID-1 ID-12 ID-11 ID-10 ID-9 ID-8 ID-7 ID-6)
DE-4	(ID-3 ID-14 ID-2 ID-13 ID-1 ID-12 ID-11 ID-10 ID-9 ID-8 ID-7)
DE-5	(ID-4 ID-3 ID-14 ID-2 ID-13 ID-1 ID-12 ID-11 ID-10 ID-9 ID-8)
DE-6	(ID-5 ID-4 ID-3 ID-14 ID-2 ID-13 ID-1 ID-12 ID-11 ID-10 ID-9)
DE-7	(ID-6 ID-5 ID-4 ID-3 ID-14 ID-2 ID-13 ID-1 ID-12 ID-11 ID-10)
DE-8	(ID-7 ID-6 ID-5 ID-4 ID-3 ID-14 ID-2 ID-13 ID-1 ID-12 ID-11)
DE-9	(ID-8 ID-7 ID-6 ID-5 ID-4 ID-3 ID-14 ID-2 ID-13 ID-1 ID-12)
DE-10	(ID-9 ID-8 ID-7 ID-6 ID-5 ID-4 ID-3 ID-14 ID-2 ID-13)
DE-11	(ID-10 ID-9 ID-8 ID-7 ID-6 ID-5 ID-4 ID-3 ID-14)

DE-1 La connaissance de la typologie de l'entreprise dans laquelle il évolue permet au RF

DEs/participant (%): 70

DE/participant: 8 ☑ Include tutor(s)

Figure 8: The DE-assignment interface

3.3. Argumentation Cycles

As a result of the first cycle, the system receives a number of answers to the proposed DEs, or so called *alternatives*. These alternatives, represented by ALT nodes in the dialog tree, will be subject of analysis in the argumentation cycles. From this moment on, AMANDA will generate a sequence of dialog cycles in order to expand the tree either in depth or in breadth, until a satisfactory degree of agreement is reached. At this point we distinguish two key concepts: the *dialog level* and the *dialog cycle*.

Dialog level

The dialog level is the *depth* level of the dialog tree, i.e. the distance from a certain node to the root. A large number of dialog levels means that the dialog has grown in depth, i.e. an original answer of a given DE has been subject of many subsequent *argumentation* cycles.

High dialog levels indicate that either the answer has been repeatedly *opposed* or progressively *clarified*, depending on the polarity of the ARG nodes. Certain typical behaviors in argumentative discourse, such as belief change, can only be detected with high dialog levels.

In practice, however, high dialog levels lead to interpretation difficulties that must be handled by the interface design. For example, suppose that a participant receives a discussion element of argumentation level 3, i.e. an Arg-3 node. It means that he is supposed to analyze his parent node (argument Arg-2) that refers to another argument (Arg-1), which in turns refers to an answer (Alt) to a given question (DE). If the user interface is not carefully designed, it's likely that we misinterpret the participant's contribution due to the large number of previous elements. On the other hand, we must present the whole history of the discussion so that the user can trace the ideas and place his contribution. This problem opens a design issue which must not be overlooked.

Dialog cycle

The dialog cycle, on the other hand, is a time period in which the dialog tree expands, possibly in depth but not necessarily. A large number of dialog cycles means that the dialog has evolved through a large number of interactions, but not necessarily that it has grown in depth. This is the distinction between the dialog level and the dialog cycle.

To exemplify, suppose that a certain answer (ALT node) exhibits low local support level (typically negative values) and low participation level (i.e. few lower level ARG nodes). This is the case, for example, when an answer is opposed by some counter-arguments, but has not been broadly discussed within the group. In this case, the system may decide to create a specific dialog cycle to re-launch this answer to be analyzed by other participants. This new dialog cycle will only increase the breadth of the tree, keeping the dialog depth unchanged.

4. Reasoning Over the Dialog

The coordination actions taken by the system are based on a certain degree of reasoning over the dialog tree. Two types of reasoning are proposed: *structural* and *semantic* reasoning.

Structural reasoning concerns to the structural aspect of dialog tree, specially the distribution of the ARG nodes and their corresponding polarities. Semantic reasoning, on the other hand, analyzes the content of the textual information in order to find semantic relations among the nodes.

The separation between structural and semantic reasoning allows AMANDA to coordinate the dialog even in the absence of domain models. The following paragraphs identify and propose some of the parameters to be evaluated in each type of reasoning.

4.1. Structural reasoning

Structural reasoning analyses the structure of the dialog tree, mainly the distribution of ARG nodes and their embedded supporting/opposing intentions, to decide which nodes will be re-launched and to which participants they will be assigned. The main structural parameter is the *support level* of a node in respect to its lower level sub-tree. The items below detail the implementation of this reasoning.

4.1.1. Evaluating the support level

Before initiating a new dialog cycle, AMANDA evaluates the overall agreement level of each DE and decides upon creating a new cycle or closing the discussion tree for the corresponding DE. This decision takes into consideration the concepts of *local* and *transmitted support level*.

Local support level

Each "supportable" node of the dialog tree (i.e. nodes of the type ALT or ARG) can be assigned a local support level (LS). This level represents the degree of consensus of this node regarding its lower level sub-tree. The support levels are calculated by traversing the dialog tree from the leaves to the root and assigning support levels to each ALT or ARG node. The local support level is a real number ranging from –1.0 to +1.0, respectively meaning total disagreement and total agreement. This number is the average level of *transmitted support* from all its direct descendant nodes (see Eq. 1). If the node has no direct child nodes, i.e. in the case of *leaf nodes*, the local support level is assigned the maximum value of +1.0.

The local support level of a node N, LS(N), is expressed by Eq. 1 and exemplified in Figure 9.

$$\text{LS(N)} = \begin{cases} \Sigma\,(\text{TS(child(N))}/n & \text{if } n > 0 \\ +1.0 & \text{if } n = 0 \end{cases}$$

Where:

- TS is the transmitted support level (Eq. 2),
- child(N) returns the next child of node N
- "n" is the number of child nodes.

Eq. 1: The local support level (LS)

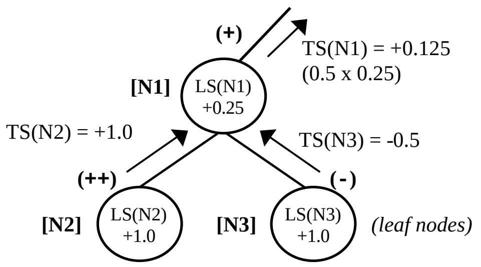


Figure 9 Local and transmitted support levels

The transmitted support level

The principle is that each descendant ARG node transmits to its direct parent a certain level of support – the *transmitted support level*. This level depends on the type of argument (++, +, -, --) and the local support level of the transmitting node itself. The nominal level that a node of type ++/+/-/-- transmits to its parent is respectively +1.0/+0.5/-0.5/-1.0.

For example, an ARG++ transmits to its direct parent a support level of +1.0 multiplied by its own local support level. Analogously, an ARG- node transmits to its direct parent a support level of -0.5 multiplied by its own local support level. In other words, the local support level acts as a "damping" parameter that tends to reduce the transmitted support level if the node does not exhibit total support from its lower levels. The support level TS(N) transmitted by a node N to its direct parent is expressed by Eq.2.

$$\text{TS(N)} = \begin{cases} +1.0 \times \text{LS'(N)} & \text{if arg-type(N) = "++"} \\ +0.5 \times \text{LS'(N)} & \text{if arg-type(N) = "+"} \\ -0.5 \times \text{LS'(N)} & \text{if arg-type(N) = "-"} \\ -1.0 \times \text{LS'(N)} & \text{if arg-type(N) = "- -"} \end{cases}$$

Where LS'(N) = min(0, LS(N))

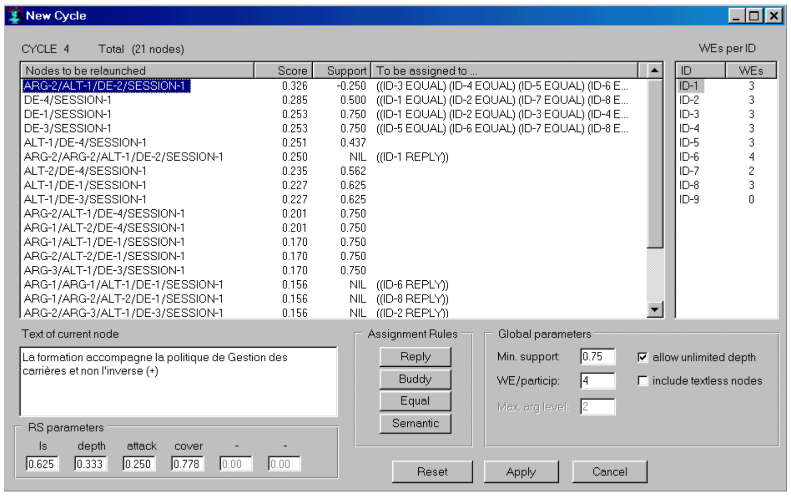
Eq. 2: The transmitted support level (TS)

An important assumption of the algorithm is that nodes with negative LS are disabled to transmit TS level to their parent by being excluded from the set of children in LS calculation. This is done to prevent highly opposed nodes from influencing their respective ascendants. In addition, this is necessary to avoid undesirable situations in which the original polarity of a node (*supporting* or *opposing*) is inverted by its negative LS.

The algorithm starts the evaluation by assigning LS values of +1.0 to all leaf nodes and then “climbs” up the tree by calculating the corresponding LS values for all nodes up to the DE node.

Tests performed in actual dialog situations show that the support levels obtained by this algorithm reflect the collective agreement of a dialog contribution within the discussion. They are used to compute a *priority value* that defines which nodes are to be re-launched in the next dialog cycle.

Figure 10 shows the interface for opening a new dialog cycle. It shows the nodes to be re-launched, their corresponding re-launch score and support levels and the assignment proposed by the system.



CYCLE 4 Total (21 nodes)

Nodes to be relaunched	Score	Support	To be assigned to ...
ARG-2/ALT-1/DE-2/SESSION-1	0.326	-0.250	((ID-3 EQUAL) (ID-4 EQUAL) (ID-5 EQUAL) (ID-6 E...
DE-4/SESSION-1	0.285	0.500	((ID-1 EQUAL) (ID-2 EQUAL) (ID-7 EQUAL) (ID-8 E...
DE-1/SESSION-1	0.253	0.750	((ID-1 EQUAL) (ID-2 EQUAL) (ID-3 EQUAL) (ID-4 E...
DE-3/SESSION-1	0.253	0.750	((ID-5 EQUAL) (ID-6 EQUAL) (ID-7 EQUAL) (ID-8 E...
ALT-1/DE-4/SESSION-1	0.251	0.437	
ARG-2/ARG-2/ALT-1/DE-2/SESSION-1	0.250	NIL	((ID-1 REPLY))
ALT-2/DE-4/SESSION-1	0.235	0.562	
ALT-1/DE-1/SESSION-1	0.227	0.625	
ALT-1/DE-3/SESSION-1	0.227	0.625	
ARG-2/ALT-1/DE-4/SESSION-1	0.201	0.750	
ARG-1/ALT-2/DE-4/SESSION-1	0.201	0.750	
ARG-1/ALT-1/DE-1/SESSION-1	0.170	0.750	
ARG-2/ALT-2/DE-1/SESSION-1	0.170	0.750	
ARG-3/ALT-1/DE-3/SESSION-1	0.170	0.750	
ARG-1/ARG-1/ALT-1/DE-1/SESSION-1	0.156	NIL	((ID-6 REPLY))
ARG-1/ARG-2/ALT-2/DE-1/SESSION-1	0.156	NIL	((ID-8 REPLY))
ARG-2/ARG-3/ALT-1/DE-3/SESSION-1	0.156	NIL	((ID-2 REPLY))

WEs per ID

ID	WEs
ID-1	3
ID-2	3
ID-3	3
ID-4	3
ID-5	3
ID-6	4
ID-7	2
ID-8	3
ID-9	0

Text of current node

La formation accompagne la politique de Gestion des carrières et non l'inverse (+)

RS parameters

ls	depth	attack	cover	-	-
0.625	0.333	0.250	0.778	0.00	0.00

Assignment Rules

Global parameters

Min. support: 0.75
WE/particip: 4
Max. arg level: 2
[✓] allow unlimited depth
[ ] include textless nodes

Figure 10: Opening a new dialog cycle

4.2. Semantic Reasoning

Due to the text-based nature of the dialog contributions and the domain dependency of the dialog, it seems reasonable to apply semantic matching techniques to improve the coordination mechanism. We identify two semantic parameters with large potential for this purpose.

The first parameter is the *semantic proximity* between textual inputs, such as direct answers or arguments. This may be useful to discover hidden relations among users’ input, specially in extensive dialog trees with large amounts of textual information. The availability of a domain ontology may extend the traditional word-matching by adding concept-based matching, as described in (Honkela, 1995).

The second parameter is the *conceptual coverage*, which aims to detect missing or insufficiently covered topics in dialog sessions. Such topics can be identified by analyzing the occurrence of certain words of domain in a given dialog sub-tree. As a response, specific DEs can be generated with the objective of bringing such subjects back to the dialog (see section 2.3).

Other text techniques, such as ontology-based information retrieval, can be applied for finding related concepts among textual information (Guarino, 1999).

One of the difficulties to apply semantic reasoning is the need for comprehensive and well constructed knowledge models, which are difficult to achieve even by experienced knowledge experts. In addition, lexical diversity may impose difficulties in relating similar concepts from different user inputs. This suggests that semantic reasoning might give better results when applied to very specific domains with low terminology diversity.

5. Conclusion

Our system was developed under empirical observations over distance learning environments, specially over the poor results achieved in traditional discussion forums. The large potential in terms of knowledge transfer of such environments encouraged us to go beyond traditional approaches and to design an environment that takes advantage of the collective discussions.

The real problem that we aim to solve is that successful distance discussion sessions require participants to be highly committed and represent a very time-consuming effort from the tutors. As a result, very few discussion forums end up satisfactorily.

We created a dialog framework that attempts to keep up the commitment of the participants by generating regular dialog activities and relieve the tutor from the dialog coordination task. This framework has been applied in actual distance training situations and has been the test-bed for various algorithms and coordination strategies.

A modular approach for the coordination mechanism, which separates structural from semantic parameters, allows it to be applied to situations where domain models are not available.

The next steps of this work are to implement the semantic reasoning over the dialog and to consolidate the results obtained in actual training situations.

Acknowledgements

We are grateful to CEGOS for its financial support and active participation in the context of the CEGOS-UTC partnership and to Siemens Telecomunicações Brasil for its financial support of the LAMI laboratory in the context of the PUCPR-Siemens partnership.

6. References

Chandrasekaran B., Johnson T., Smith J. Task-structure analysis for knowledge modeling. Communications of the ACM (CACM 35) no.9, 1992, pp. 1124-137.

Decker K. Environment Centered Analysis and Design of Coordination Mechanisms; Ph.D. thesis. Department of Computer Science; University of Massachusetts, Amherst, 1995.

Eleuterio M., Eberspächer H. A Knowledge Management Approach to Virtual Learning Environments. International Workshop on Virtual Education (WISE), 1999.

Eleuterio M., Eberspächer H.,Vasconcelos C., Jamur J. Eureka: um ambiente de aprendizagem cooperativa baseado na Web para Educação à Distância. In: Brazilian Symposium on Informatics in Education (SBIE), 1999.

Guarino N., Masolo C., and Vetere G., OntoSeek: Content-Based Access to the Web, IEEE Intelligent Systems 14(3), May/June 1999.

Guarino N., Welty, C. Ontological Analysis of Taxonomic Relationships. In, Laender, A. and Storey, V., eds, Proceedings of ER-2000: The 19th International Conference on Conceptual Modeling. Springer-Verlag LNCS. 2000.

Honkela T. Self-organizing maps in natural language processing. Ph.D. thesis. Helsinki University of Technology, 1995.

Karacapilidis N., Papadias D. A computational approach for argumentative discourse in multi-agent decision making environments. AI communications 11 (1998) 21-23.

Leary D. Using AI in Knowledge Management: Knowledge Bases and Ontologies; IEEE Intelligent Systems, May/June, 1998.

Mizoguchi R., Bourdeau J. Using Ontological Engineering to Overcome Common AI-ED Problems; International Journal of Artificial Intelligence in Education 2000.

Nonaka I., Toyama R., Konno N. SECI, Ba, and Leadership: A Unified Model of Dynamic Knowledge Creation; D.J. Teece and I. Nonaka (Eds). Oxford University Press, 1999.

Porayska-Pompa K, Pain H. Aspects of Speech Act Categorisation: Towards Generating Teacher’s Language. International Journal of Artificial Intelligence in Education, 2000.

Ravenscroft A., Pilkington R. Investigation by design: developing dialog models to support reasoning and conceptual change. International Journal of Artificial Intelligence in Education, 2000.

Van der Vet P., Mars N. Bottom-up Construction of Ontologies. IEEE Transactions on Knowledge Engineering, vol. 10 no. 4, 1998.