

From Discourse Plans to Believable Behavior Generation

Berardina De Carolis

Dipartimento di Informatica
Università di Bari
Italy

decarolis@di.uniba.it

Valeria Carofiglio

Dipartimento di Informatica
Università di Bari
Italy

carofiglio@di.uniba.it

Catherine Pelachaud

Dipartimento Informatica e Sistemistica
Università di Roma "La Sapienza"
Italy

cath@dis.uniroma1.it

Abstract

Developing an embodied conversational agent that is able to exhibit a human-like behavior while communicating with other virtual or human agents requires enriching a typical NLG architecture. The purpose of this paper is to describe our efforts in this direction and to illustrate our approach to the generation of an Agent that shows a personality, a social intelligence and is able to react emotionally to events occurring in the environment, consistently with her goals and with the context in which the conversation takes place.

1 Introduction

Humans communicate combining signals of different nature. Body posture, gestures (pointing at something, describing object dimensions,...), facial expressions, gaze (making eye contact, looking down or up, at a particular object,...) may be combined with speech. The way in which people communicate, and therefore the employed signals, is influenced by their personality, goals and affective state and by the context in which the conversation takes place. Developing a "computer conversationalist" that is embedded, for instance, in a virtual human-like body and is able to exhibit these added dimensions of communication requires moving from natural language generation to multimodal behavior generation.

The purpose of this paper is to describe our efforts in this direction and to illustrate how a typical NLG architecture has been changed to generate context-adapted behavior in a Conversational Embodied Agent. Our Agent shows

a personality and a social intelligence and is able to react emotionally to events occurring in the environment, consistently with the context in which the conversation takes place and with her goals. To achieve such a context-adaptable multimodal behavior we employed a typical pipelined NLG architecture (Reiter and Dale, 2000): given a communicative goal to be achieved, the behavior generator plans the communication content at an abstract level ("what to say") and then realises it at the surface level according to expressive capabilities of the used "body" and to the conversational context ("how to say it"). These two steps may be more or less complex and may require an additional phase, aimed at "optimising" the communication from the content and style viewpoint ("sentence planner").

Planning the behaviour of our Agent may be approached in two alternative ways. In the first one, the planner may decide which verbal and non-verbal signal/s to employ in every conversational move; in the second one, it may define only the communicative function/s to attach to every move and leave, to the surface realizer, the task of deciding which signal/s to employ. The first perspective requires exploiting knowledge about mental and physical capabilities of the agent during planning. The planner must, as usual, establish the discourse steps that the agent will carry out to achieve the given communicative goal; in addition, it has to indicate the combination of signals through which every step of the planned discourse will be rendered. The main advantage of this solution is that the dialog move will be planned consistently with the agent's state, by establishing how verbal and non verbal components will be combined so as to avoid introducing signal redundancies or to decide to introduce them on purpose: for instance, by expressing a deictic act only with gaze or with both

gaze and words. Obviously, this solution is body-dependent. In the second perspective, the plan move will be signal-independent and the "body" generator will interpret what was produced by the planner to realise it according to the specific body capabilities. To avoid signal redundancy or conflicts, this second approach requires body-dependent rules at an intermediate *sentence planning* level.

This is the perspective we adopt. In this view, the Agent is seen as an entity made up of two main components, a 'Mind' and a 'Body', which are interfaced by a common I/O language, so as to overcome integration problems and to allow their independence and modularity. During the conversation, the Agent's Mind decides what to communicate, by considering the dialogue history, the conversational context and her own current cognitive state. The Body "reads" what the Mind decides to communicate and interprets and renders it at the surface level, according to the available communicative channels: different bodies may have different expressive capabilities and therefore may use different channels. To achieve a rich expressiveness, the output of the Agent's Mind cannot be just a combination of symbolic descriptions of communicative acts. It should include, as well, a specification of the 'meanings' that the Body will have to attach to each of them. The Mind of our believable conversationalist has to be able to perform the following functions: select an appropriate dialog move, decide whether, in correspondence of that move, an emotion is triggered and, finally, specify the meanings that have to be conveyed through the selected move. These meanings include the communicative functions that are typically used in human-human dialogs: topic-comment, affective, meta-cognitive, performative, deictic, adjectival and belief relation functions (Poggi et al., 2000).

To specify the format of the dialog move that should act as an interface between the Agent's Mind and her Body, we designed a Mind-Body interface that takes as input a specification of a discourse plan in an XML language (DPML: Discourse Plan Markup Language) and enriches this plan with the meanings that have to be attached to it, by producing an input to the Body in a new XML language (APML: Affective Presentation Markup Language). This approach is

also motivated by the fact that the Body we use in the MagiCster project¹ is a 3D realistic face called Greta (Pelachaud et al., 2001) and a synthetic voice (Festival homepage).

This paper is structured as follows. After describing our two-layered system architecture, we will describe the Mind-Body interface. To illustrate how this architecture works, we will use an example from the medical domain. Conclusions will be discussed in the last Section.

2 MagiCster Architecture

Let us start with two examples of dialog in the medical domain (Table 1).

<p>Greta0a: Good morning Doctor Eva. D1: Good morning Doctor Greta. Have you seen the tests of Mr. Smith? Greta1a: Yes, he has got a mild form of angina. D2: What are you prescribing him? Greta2a: Aspirin and Atenolol </p> <p style="text-align: center;">Example-1a</p> <p>Greta0b: Good morning Mr. Smith. U1: Good morning Doctor Greta. Have you seen my tests? Greta1b: Yes, ... and I'm sorry to tell you that you have been diagnosed as suffering from angina pectoris, which appears to be mild. U2: What is angina? Greta2b: Angina is a spasm of chest resulting from overexertion when heart is diseased. U3: Is it possible to cure it? Greta3b: Yes, a drug therapy does exist. To solve your problem, you should take two drugs. The first one is Aspirin and the second one is Atenolol. </p> <p style="text-align: center;">Example- 1b</p>
--

Table1. Examples of Dialog between (1a) two doctors and (1b) a doctor and a patient.

In both dialogs, the Agent (named Greta) takes the role of a doctor and the interlocutor is either a colleague (1a) or a patient (1b). In the first case, doctor Greta informs a colleague about Mr. Smith's health conditions, while in the second case the Interlocutor is the patient asking for information about his disease.

To show a believable behaviour, the agent has to act consistently with her role, mental state, goals, personality and social context; this is especially important in delicate conversational fields such as medical advice-giving. As we mentioned in the Introduction, this 'believable' behavior is related not only to the capability of

¹ IST Project IST-1999-29078, partners: University of Edinburgh, Division of Informatics; DFKI, Intelligent User Interfaces Department; SICS; University of Bari, Dipartimento di Informatica; University of Rome, Dipartimento di Informatica e Sistemistica; AvatarMe.

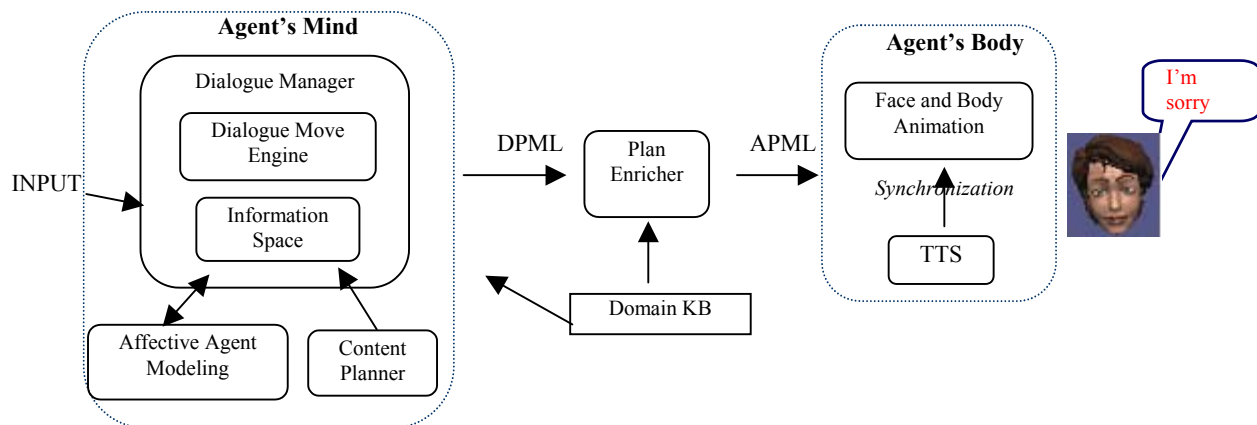


Figure 1. MagiCster Architecture.

using the communicative functions that are typical of human-human dialogs, but also the capability of deciding whether an emotion is felt and, according to the interactional context. When addressed to a colleague, the agent's dialog move may be expressed in a more direct and concise way (both doctors have a common knowledge). On the contrary, when the interlocutor is a patient, more information and particular attention to the way information is exposed and expressed is required. Moreover, in this last case, emotions are involved. For instance, while conversing with the patient (Example 1b), doctor Greta will coordinate her speech with various expressions:

- in Greta1b move, she manifests her empathy with the user. She does it not only verbally ("I'm sorry to tell you") but also nonverbally, by displaying the expression of "sorry-for". Expressing empathy will not be necessary if the conversational partner is a doctor or a nurse (Example 1a). To play down on the seriousness of the illness, Greta will emphasise both verbally and nonverbally the fact that it is still in a "mild" form.
- in Greta2b move, Greta indicates her chest while saying 'a spasm of chest' while, in turn Greta3b, she looks at the User while saying 'your problem'. The two expressions are realised through a particular gaze direction that plays a deictic function to indicate a given point in space.

Let us see now how the MagiCster's architecture supports the generation of dialogs of

this kind. As mentioned in the Introduction, the architecture includes two main components (a Mind and a Body), interfaced by a Plan Enricher. The **Agent's Mind** is responsible for deciding which dialog move to perform ("what to say") and includes a Dialogue Manager, a Content Planner and an Affective Agent Modelling module. Its result is a discourse plan formalized according to DPML. The **Body** is a 3D face, with a speech synthesiser (Festival homepage) for animated spoken delivery and accepts as input an APML specification.

We will briefly describe each module, to focus our description on the Mind-Body Interface that has the role of transforming the DPML output in the APML input.

2.1 The Affective Agent Modelling

This module is responsible for updating the Agent's mental state: it decides whether a particular affective state should be activated and with which intensity. The mental state is represented as a Dynamic Belief Network that is built automatically at every dialog turn from two main components: the network that represents the agent's state at the previous turn and the network(s) that represent the event(s) occurred in the interval between the two turns, with their possible causes and effects (de Rosis et al, in press).

2.2 The Content Planner

In NLG systems, planning is the step in which discourse coherence is ensured; a planner allows

one to flexibly generate the discourse structure that is appropriate in a given situation. In planning dialog moves, our planner works at two different levels. At a more abstract level, a plan representing the steps needed to achieve the initial communicative goal(s) is generated. These plan steps are then expressed as more or less complex dialog moves. At this lower level of abstraction, each step is expanded in a discourse plan representing the rhetorical organisation of its content. In both cases, we do not use a sophisticated planner but perform this task by retrieving an appropriate ‘recipe’ from a plan library. Recipes in the library are abstracted from a corpus of ‘natural’ presentations in the considered application domain. In the first case, the recipe is a set of dialog goals to be achieved during the dialog; in the second case, it is a discourse structure expressed according to DPML. DPML is based on XML and allows one to represent discourse plans according to Rhetorical Structure Theory (RST: Mann and Thompson, 1988) as evident from the DTD definition in Figure 2.

```

<!ELEMENT d-plan (node+)>
<!ATTLIST d-plan
  name CDATA #REQUIRED
>
<!ELEMENT node (node*, info*)>
<!ATTLIST node
  name CDATA #REQUIRED
  goal CDATA #REQUIRED
  role (root | nucleus | sat)
  #REQUIRED
  RR CDATA #REQUIRED
  focus CDATA #REQUIRED
  ~

```

Figure 2. DPML DTD

A discourse plan is a tree identified by its name; its main components are the nodes, that are identified, as well, by a name; nodes include mandatory attributes describing the communicative goal, the discourse focus and the rhetorical elements (role in the rhetorical relation (RR) of the father-node and RR). The XML-based annotation of discourse plans is justified by several reasons. The main ones concern the possibility of i) building a library of standard explanation plans, ii) using XML as a standard interface between the generator modules, so as to favour resources distribution and re-use, iii) enabling easy translation to other XML representations.

2.3 The Dialogue Manager

This module is built on top of the TRINDI architecture (Trindi homepage), which provides an engine (DM) for computing dialog moves and a space in which information relevant to the move selection and effect can be stored: for instance, the agent’s mental state and the current plan. After a dialog plan has been selected from the library of plan recipes, the first Agent move is generated according to the first step of this plan.

Let us consider Example 1b in Table 1. In this case, the planner sets the following dialog goals in the agenda in the Information Space (IS), that the agent will have to achieve during the conversation:

```

g0: Greet(A,U),
g1: Explain(A, U, has(U, angina)),
g2: Inform(A, U, description(angina)),
g3: Describe(A, U, therapy(angina)).

```

The dialog starts with the plan corresponding to the first goal. Then, the DM controls its flow by iterating the following steps, until the conversation ends:

1. the initiative is passed to the User, that can make questions on any of the topics under discussion;
2. the User move is translated into a symbolic communicative act (through a simplified interpretation process) and is passed to the DM;
3. the DM decides “what to say next” by selecting the sub-plan to execute.

During this cycle, the information space is used as follows: i) the symbolic representation of the user move is stored in the shared space of the IS; ii) this information is used by the Agent modelling component for deciding whether an emotion is triggered by this event and, if so, whether it has to be displayed. Information about the emotion to be displayed and its intensity are stored in the IS; iii) the DM uses this information to perform step 3) through an appropriate choice of a sub-plan and writes in the IS the next dialog move that the Agent has to perform.

In Example-1b, after greeting, the user request of *knowing about his health state* is stored in the IS as Ask(U,A, healthstate(U)) and is passed to the Affective Agent Modelling module that returns a “sorry-for” emotion. Then, DM selects the discourse plan allowing one to achieve g1. This is a complex dialog move whose DPML recipe is shown in Figure 3:

```

<node name="n1"
goal="Explain(Has(U,disease))"
role="sat"focus="disease"RR="ElabObjAttr"
>
  <node name="n2"
goal="Inform(Has(U,disease))"
role="nucleus"
  focus="Has(U,disease)"
RR="null"/>
  <node
name="n3"goal="Inform(Severity(dis
ease))" role="sat"
focus="Severity(disease)"
RR="null"/>
</node>

```

Figure 3. DPML representation of plan(g1).

2.4 The Plan Enricher

This module translates the symbolic representation of a dialog move into an Agent's behaviour specification. A dialog move may be a 'primitive' communicative act (for instance: a 'greet', a 'thanks', an 'inform', a 'request') or a more complex plan (as in Figure 3), annotated according to DPML. An algorithm translates this DPML-based tree-structure into another XML-based language (APML), through a set of transformation rules that depends on the information attached to nodes in the discourse plan: rhetorical relation name and type, communicative goal, discourse focus and so on. In instantiating plans and generating the verbal part of the communication, the domain model is consulted. Before showing how the transformation is performed, we will describe APML.

APML: A Markup Language for Behavior Specification

The use of Embodied Agents in human-computer interaction has increased the need of controlling their behaviour in an easier way than writing programs. A solution to this problem has been found in the use of XML-based languages, that include high-level primitives for specifying behaviour acts similar to those performed by humans. An effort in building a standard in this direction is represented by the Human Markup Language initiative (HML homepage). This language allows one to specify human communicative behaviors at a very high-level. The aim of HML is to "develop Internet tools and repository systems which will enhance the fidelity of human communications". It is therefore

designed to represent human characteristics through XML. Its specification modules include tags allowing the representation of physical, cultural, social, kinetic, psychological, and intentional features used by humans in communicating information. Envisaged applications of HumanML include agents of various types, AI systems, virtual reality, online negotiations, dialog and conflict resolution systems.

HML is a language at a very abstract level: using it for controlling specific agent bodies may be difficult and may require developing complex interpreters to translate a very abstract specification into low-level body actions. For this reason, researchers tend to develop their own languages, more suited to the type of embodied agent they wish to control. For instance, MPML (Multimodal Presentation Mark-up Language) has been developed with the aim of enabling authors of Web pages to add to them agents for improving human-computer interaction (Prendinger et al, in press). The design of this language has been driven by the choice of Microsoft Agent as a body. For instance, the tag for specifying a predefined animation sequence (<act>) takes, as a possible value, one of the MS-agent's animations. Therefore, this language is not body-independent.

Another XML language that was designed for generating embodied agent's behaviour is BEAT developed by (Cassell, et al, 2001). Here, the XML language is used for tagging both the agent's input and output. The input is an utterance that is parsed into a tree structure; this tree is manipulated to include information about non-verbal signals and then serialized again in XML. The output language, specifying the agent's behaviour, contains tags describing the type of animation to be performed and its duration.

APML aims also at describing the main communicative functions, and therefore the related expressions that are typical of an embodied conversational agent. As we said, this language is justified by the need of insuring independence of the Agent's Body from its Mind: this will enable an application to decide the behaviour to adopt in a particular interaction context, to select a Body suited to the Agent's personality and role, to the culture in which it will be employed and to the resources available: avatars, 3D or 2D characters

```

APML - Affective Presentation Markup Language DTD
<!ELEMENT APML (turn-allocation+, performative*, turn-allocation*)>
<!ENTITY %TA-TYPE "(take|give)">
<!ENTITY %P-TYPE "(inform|ask|greet|request|...)">
<!ENTITY %BR-TYPE "(adj|ElabObjAttr|ElabGenSpec|justification|motivation|...)">
<!ENTITY %A-TYPE "(joy|sorry-for|distress|...)">
...
<!ELEMENT turn-allocation (performative*)>
<!ATTLIST turn-allocation type %TA-TYPE #REQUIRED>
<!ELEMENT belief-relation (#PCDATA|performative)>
<!ATTLIST belief-relation type %BR-TYPE #REQUIRED>
<!ELEMENT performative ((adjectival|deictic)*, belief-relation*)>
<!ATTLIST performative type %P-TYPE #REQUIRED affect %A-TYPE #IMPLIED
certainty %C-TYPE #IMPLIED>
<!ELEMENT adjectival (#PCDATA)>
<!ATTLIST adjectival type %ADJ-TYPE #REQUIRED>
<!ELEMENT deictic (#PCDATA)>
<!ATTLIST deictic obj CDATA #REQUIRED>
...

```

Figure 4. APML DTD specification

with a context and culture-tailored physical aspect or even a cartoon when display, speed and memory capacities are more limited, are examples of possible bodies.

Poggi et al. (2000) defined a communicative function as a (meaning, signal) pair, where the meaning item corresponds to the communicative value of the signal item. For instance, a smile can be the signal of a “joy” emotion. This distinction between the meaning and the way in which the meaning can be communicated has driven the design of APML. Due to the architectural choice of Mind-Body separation, tags should not specify the signal(s) to be conveyed but only the meaning(s) associated with the act to be communicated. In addition, the Agent’s believability is strictly related, as we said, to features such as her personality, her role as well as the cultural and social context; therefore, abstracting, in the behaviour specification phase, the way in which the Body will render that meaning fosters adaptivity to these features.

The APML DTD is described in Figure 4: we decided to show here the DTD instead of the XML-Schema for space reasons, since Schemas have a less compact representation than DTDs. The first part of the DTD defines the enabled values for the tag attributes while the second part specifies tags and their nesting in the definition of a valid APML structure. Every dialog turn specified with this language starts with the root tag <APML>. To indicate that the agent is taking or giving the initiative, the turn-allocation tag may then be used: its type attribute may take, as values, “take” or “give”. Another tag is used to describe the

performative; its attributes allow one to attach to it the following information:

- type:** the type of performative, that may take one of the values specified in P-TYPE domain,
- affect:** an emotion in the A-TYPE set,
- certainty:** of what is being communicated.

This language also specifies rhetorical relations between message spans, through the <belief-relation> tag, whose type attribute is set with the name of the RR.

The <adjectival> tag may be attached to words denoting a ‘quantitative’ adjective, to which the body may associate an iconic expression; for instance, ‘big’ may be expressed by opening the eyes widely.

The <deictic> tag may be attached to words referring to objects having a specified position in the domain space: the agent may refer to them by using deictic gestures such as pointing, looking at etc.

So far we only defined the meanings that may be translated into facial expressions. In the future, we plan to extend our language to represent meanings that may be expressed with other nonverbal signals (for instance: gestures).

Compared to the languages described above, APML may be seen as a finer-grained language than HML: its schema definition could refer to HML types such as intentions, emotions and so on. Compared to the BEAT output language, APML appears to be at a higher level of abstraction, since it does not include any reference to the employed signal or animation. MPML is about at the same level then APML; however, our tags are derived by

a theory of human communication while MPML tags are defined after the actions that a specific Body (MS-Agent) is able to perform.

2.5 The Body Generator

This module interprets the APMML-tagged dialog move and decides how to convey every meaning (by which combination of signals: facial expression, gaze and/or head movement). As mentioned previously, the Body we use at present is a combination of a 3D face model, compliant with the MPEG-4 standard, and a speech synthesizer. Her name is “Greta” and she is capable of expressing the nonverbal communicative functions foreseen for our conversational agent.

3 An Example of translation from DPML to APMML

Given a DPML tree, the transformation algorithm (called Midas) reads it recursively down to the leaves. The root tag <APML> is introduced initially, followed by the <turn-allocation> tag whose type attribute is set to “take”, to indicate that the agent takes the initiative. After this step, the appropriate recursive schema is activated, according to the value of the ‘RR attribute’ attached to the node. Every RR attribute is transformed into a <belief-relation> whose type attribute is set with the name of the RR. The general rule is to put the <belief-relation> tag emphasis on the RR marker and the satellite, for nucleus-satellite RRs as shown in the example in Figure 6; only on the RR marker for multinuclear ones (i.e. Ordinal Sequence, Contrast, etc.).

An example of this case is the following:

```
The <belief-relation type="ordinaalseq">
first</belief-relation> <performative
type="inform" certainty="certain"> drug
is aspirin </performative> ...
```

Let us consider, again, the DPML structure in Figure 3. The root node is n1 and its RR is the ElaborationObjectAttribute; this includes a nucleus, which mentions an object, and a satellite, that describes a property of that object. In this case, the Midas transformation function may apply different recursive schemas.

One of them is shown in Figure 5: a recursive call is first made on the nucleus, then the

<belief-relation> tag is generated and a new recursive call is made on the satellite; finally, the <belief-relation> tag is closed. The generate_rr function allows one to generate the appropriate marker (which or that in this case).

```
Current_node=n1
Current_node.role=root
=>write("<APML><turn-allocation
type="take">")
Begin Midas(current_node)
Current_node.RR=ElabObjAttr =>
Midas(current_node.firstchild)
(n2 RR=null - leaf)
write("<belief-relation
type="ElabObjAttr"> ")
generate_rr(ElabObjAttr)
Midas(current_node.secondchild)
(n3 RR=null - leaf)
write("</belief-relation>")
end Midas
write("</turn-allocation></APML>")
```

Figure 5. An example of Translation Schema

An alternative schema for this RR opens first the <belief-relation> tag, then calls Midas on the satellite, closes the <belief-relation> tag and calls again Midas on the nucleus. This schema allows generating sentences of the type: attribute object.

When the algorithm reaches a leaf node, the generate_performative function is called and the recursion ends. This function is responsible for the surface realisation, in which the <performative> element is generated. If the Affective Agent Modelling component establishes that an emotion is felt by the Agent in correspondence with the current node and that this emotion has to be displayed, the affect attribute of the performative tag is set to that emotion’s name.

The generate_performative function, besides generating the <performative> tag, produces the verbal part of the speech act and includes, if needed, two more tags: the <adjectival> one (when the argument of the communicative goal is a quantitative attribute of the discourse focus) and the <deictic> one (when the argument of the communicative goal is described in the domain knowledge base as ‘referable through its coordinates’). In the example, “severity” is a quantitative property of angina, which is the discourse focus: therefore, the <adjectival> tag is generated around the attribute-word.

The following APMML string shows what is generated for the plan in Figure 3:

```

<APML><turn-allocation type="take">
<performative type="inform" affect="sorry-for"
certainty="certain"> I'm sorry to tell you that
you have been diagnosed as suffering from
angina pectoris,</performative> <belief-
relation type="eoa"> which <performative
type="inform" certainty="uncertain"> appears to
be <adjectival type="small">mild. </adjectival>
</performative> </belief-relation>
</turn-allocation></APML>

```

Figure 6. APMML representation of the plan in Figure 3.

4 Conclusions

In this paper, we have described the architecture of the behaviour generator of a believable conversational agent. In particular, we focused our discussion on the importance of Mind-Body separation and therefore on the need of an interface between the two modules.

This interface should be able to represent the communicative functions that can be potentially realized by different bodies with different expressive capabilities. We have defined two XML-based mark up languages to represent the Mind's output (DPML) and the Body's input (APML). We have also described how a plan enricher transforms DPML trees into APML trees.

There are still some open problems. The first one is related to tag repetition: some tags should not always be included in the annotated sentence. For instance: deictic gestures should not appear each time the agent refers to a given object. A possible solution is to include rules for deciding whether to tag them according to the interaction history and context as well as if the referent is in focus or not. Another problem is related to tags that depend on the dialog dynamics (i.e. from 'meta-cognitive' aspects); these tags that cannot be derived by interpreting DPML plans but require reasoning on the particular type of event under discussion.

Besides trying to solve these problems, we are also studying how to automatically integrate text annotations that enable us to improve the speech intonation of our believable agent (Hitzeman et al., 1999).

Acknowledgements

We wish to acknowledge Fiorella de Rosis for her useful suggestions.

References

- J. Cassell, H. Vilhjamsson, T. Bickmore. BEAT: the Behavior Expression Animation Toolkit. In Computer Graphics Proceedings, Annual Conference Series, ACM SIGGRAPH, 2001.
- B. De Carolis, F de Rosis, C. Pelachaud and I. Poggi. Verbal and nonverbal discourse planning. In Proceedings of IJCAI 2001 Seattle, August 2001.
- F. de Rosis, C. Pelachaud, I. Poggi, V. Carofiglio and B. De Carolis: From Greta's Mind to her Face: Modeling the Dynamics of Affective States in a Conversational Embodied Agent. International Journal of Human-Computer Studies. In press.
- FESTIVAL Home Page:
<http://www.cstr.ed.ac.uk/projects/festival/>
- J. Hitzeman, A. W. Black, C. Mellish, J. Oberlander, M. Poesio & P. Taylor (1999). An Annotation Scheme for Concept-to-Speech Synthesis. Proceedings of the European Workshop on Natural Language Generation, pp. 59-66, Toulouse, France.
- HML homepage: <http://www.hml.org>
- W. Mann and S Thompson. Rhetorical Structure Theory: towards a functional theory of text organization. Text, 3: 243-281. 1988.
- C. Pelachaud, E. Magno-Caldognetto, C. Zmarich, and P. Cosi. An approach to an Italian talking head. In Eurospeech'01, Aalborg, Denmark, September 2001.
- I. Poggi, C. Pelachaud, and F. de Rosis. Eye communication in a conversational 3D synthetic agent. Special Issue on Behavior Planning for Life-Like Characters and Avatars of AI Communications. 2000.
- H. Prendinger, S. Descamps, M. Ishizuka. Scripting Affective Communication with Life-like Characters in Web-based Interaction Systems. Applied AI, Special Issue on 'Merging Cognition and Affect in HCI. F.de Rosis (Guest Editor), in press.
- E. Reiter and R. Dale. Building Natural Language Generation Systems. Cambridge University Press. 2000.
- TRINDI Home Page:
<http://www.ling.gu.se/projekt/trindi/>