Algorithms for Controlling Cooperation between Output Modalities in 2D Embodied Conversational Agents

Sarkis Abrilian*, Jean-Claude Martin*† and Stéphanie Buisine*
* LIMSI-CNRS, BP 133, 91403 Orsay Cedex, France, +33.1.69.85.81.04
† LINC-Univ Paris 8, IUT de Montreuil, 140 rue de la Nouvelle France, 93100 Montreuil, France
{sarkis,martin,buisine}@limsi.fr

ABSTRACT

Recent advances in the specification of the multimodal behavior of Embodied Conversational Agents (ECA) have proposed a direct and deterministic one-step mapping from high-level specifications of dialog state or agent emotion onto low-level specifications of the multimodal behavior to be displayed by the agent (e.g. facial expression, gestures, vocal utterance). The difference of abstraction between these two levels of specification makes difficult the definition of such a complex mapping. In this paper we propose an intermediate level of specification based on combinations between modalities (e.g. redundancy, complementarity). We explain how such intermediate level specifications can be described using XML in the case of deictic expressions. We define algorithms for parsing such descriptions and generating the corresponding multimodal behavior of 2D cartoon-like conversational agents. Some random selection has been introduced in these algorithms in order to induce some "natural variations" in the agent's behavior. We conclude on the usefulness of this approach for the design of ECA.

Categories and Subject Descriptors

H.5.2-H.5.1 [**Information Interfaces and Presentation**]: User Interface – *interaction styles, standardization, ergonomics, user interface management systems.* Multimedia Information Systems.

General Terms

Algorithms, Human Factors, Languages.

Keywords

Multimodal output, Embodied Conversational Agent, Specification, redundancy.

1. INTRODUCTION

Amongst multimodal output interfaces, Embodied Conversational Agents (ECA) seem to be promising for the intuitiveness and richness of Human-Computer Interaction. Advances in the specification of the multimodal behavior of ECA have mostly proposed direct one-step mappings from high-level specifications of dialog state or agent emotion onto low-level specifications of the multimodal behavior to be displayed by the agent (e.g. facial expression, gestures, vocal utterance). For example, the SAFIRA project [1] proposes a dual top-down approach via the Character Mark-up Language (from personality, emotion and behavior to animation) and bottom-up approach via the Avatar Mark-up Language (selection and synchronized merging of animations). The NECA system [6] generates the interaction between two or more characters in a number of steps, with the information flow proceeding from a Scene Generator to a Multi-modal Natural Language Generator, to a Speech Synthesis component, to a Gesture Assignment component, and finally to a media player. Thus a representation language was defined as a means for representing the various kinds of expert knowledge required at the different interfaces between the components. Other XML based specification language for ECA include VHML[4], MPML[7], APML[3]. All these languages propose mappings between a rather "high level" of abstraction and a "low level" of abstraction (e.g. translating a "happy" tag into corresponding animations of facial expressions and prosodic parameters for speech synthesis). The difference of abstraction between these two levels of specification makes difficult the definition of such a complex mapping which is indeed a key issue in the design of "believable" ECA. One potential dimension of cooperation between modalities, which is not considered in such specification languages is the degree of redundancy vs. complementarity between signals conveyed by several modalities for rendering different emotional states or communicative act strengths. Moreover, in most systems this mapping is deterministic. That makes the agent always react exactly in the same way to a given situation. Such a behavior might appear consistent but quite unnatural when compared to the complexity of human communication and reactions.

Section 2 describes the 2D agent technology we use. In section 3, we define algorithms for parsing such "intermediate level" descriptions and generating the corresponding multimodal behavior of 2D cartoon-like conversational agents in the case of classical referring expressions. Some random selection has been introduced in these algorithms in order to induce some "natural variations" in the agent's behavior.

2. LOW-LEVEL SPECIFICATION

We use 2D cartoon-like agents developed in Java. A catalogue of images representing several configurations of each body part has been designed. We present below a low-level specification of a configuration of the agent and the corresponding display in Figure 1 (text is both displayed and rendered using IBM ViaVoice):

```
<configuration>
  <speech>Hello</speech>
  <body>front</body>
  <head>front</head>
  <eyebrows>up</eyebrows>
  <gaze>middle</gaze>
  <mouth>open</mouth>
  <leftArm>hip</leftArm>
  <rightArm>forearmUpPalmFront</rightArm>
</configuration>
```

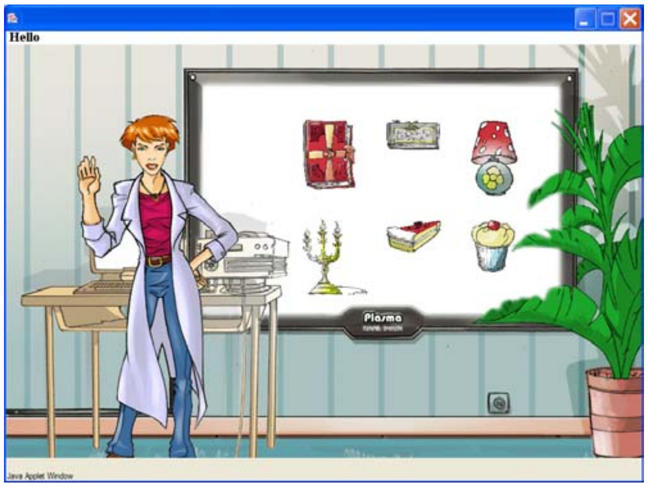

Figure 1. Display generated by the low-level specification above (the spoken utterance: "Hello" is also displayed in the upper left corner).

The illustrative examples that we use consist in generating multimodal references to graphical objects displayed besides the agent. There are 18 objects of different types, size and color. The objects attributes are declared in a XML file (the id attribute is used for referring to objects' declarations in the agent's behavior):

```
<objectset>
  <object>
    <id>1</id>
    <name>kitchen_book</name>
    <size>small</size>
    <shape>rectangular</shape>
    <color>gray</color>
    <position>
      <x>565</x>
      <y>130</y>
    </position>
    <characteristic>light</characteristic>
  </object>
  …
</objectset>
```

3. CONTROLLING COOPERATION BETWEEN MODALITIES

We focus on two dimensions involved in the reference to an object: the attributes of the object the agent has to refer to (e.g. its name, its type, its color) and the output modalities that can be used by the agent (e.g. speech, pointing gesture, iconic gesture). In the following sub-sections, we illustrate various ways of specifying the combinations between modalities as well as the corresponding low-level description which is generated by our software and the corresponding display. An on-line demonstration is available on the web[1].

3.1 Fully-specified cooperation

In this case, the intermediate level specification includes both the attributes that the agent has to communicate and the set of modalities that it should use:

```
<complemredundancy>
  <idObj>1</idObj> // id of the book
  <attributeset>
    <attribute>name</attribute>
    <attribute>size</attribute>
    <attribute>shape</attribute>
    <attribute>position</attribute>
  </attributeset>
  <modalityset>
    <modality>speech</modality>
    <modality>bothArms</modality>
  </modalityset>
</complemredundancy>
```

The specification is parsed by the following algorithm:

```
For each specified modality m
  For each specified attribute a
    If the value of a can be referred in m,
    Then generate a reference to a in m
```

The following low-level specification is generated by our software and the corresponding display is provided in Figure 2.

```
<configuration>
  <speech>small rectangular kitchen book to
the up left</speech>
  <body>front</body>
  <head>front</head>
  <eyebrows>up</eyebrows>
  <gaze>middle</gaze>
  <mouth>open</mouth>
  <bothArms>rectangularShape</bothArms>
</configuration>
```

3.2 Controlling the degree of complementarity vs. redundancy

Instead of providing the full specification of both the attributes and the modalities to be involved in the agent's behavior as in the previous section, we introduce in this section means for controlling the degree of cooperation between modalities on two dimensions: object's attributes and output modalities.

3.2.1 Controlling the selection of attributes

The specification below makes use of a random factor in the selection of the attributes for display via a fully-specified set of modalities:

```
<complemredundancy>
  <idObj>3</idObj>
  <attsalience>70</attsalience>
  <modalityset>
    <modality>leftArm</modality>
    <modality>gaze</modality>
    <modality>speech</modality>
```

[1] http://www.limsi.fr/Individu/martin

```
      </modalityset>
   </complemredundancy>
```

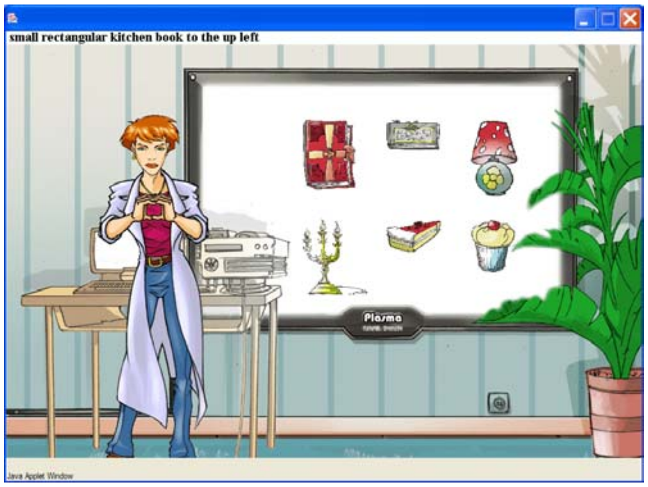

Figure 2. Display generated from the fully-specified redundancy/complementarity (spoken utterance: "small rectangular kitchen book to the up left").

In the above specification, the object referred to is:

```
   <object>
      <id>3</id>
      <name>candlestick lamp</name>
      <size>medium</size>
      <shape>fork</shape>
      <color>silver</color>
      <position>
         <x>440</x>
         <y>280</y>
      </position>
   <characteristic>shining</characteristic>
   </object>
```

The algorithm we use to parse such a specification is:

```
 For each specified modality m
   let A the attributes displayable by m
   let nbAtt = round(attributesalience*|A|)
   Select randomly nbAtt attributes in A
```

The specified modalities in the above specification were *leftArm*, *gaze*, *speech*. For each modality, 70% of the object attributes which can be displayed are selected. In this example:

```
Speech  :
   A = {name, size, shape, color, position,
 characteristic}
   nbAtt = round(70 * 6 / 100) = 4
   Selected attributes : name, size,
 position, characteristic

LeftArm:   A = {position}
   nbAtt = round(70 * 1 / 100) = 1
   Selected attributes : position

 Gaze:   A = {position}
   nbAtt = round(70 * 1 / 100) = 1
   Selected attributes : position
```

The following low-level specification and display (Figure 3) are generated by our software implementing the algorithm above:

```
<configuration>
   <speech>medium shining candlestick lamp
 to the middle left</speech>
   <body>front</body>
   <head>front</head>
   <eyebrows>up</eyebrows>
   <gaze>left</gaze>
   <mouth>open</mouth>
   <leftArm>pointingMiddleLeft</leftArm>
   <rightArm>drop</rightArm>
   </configuration>
```

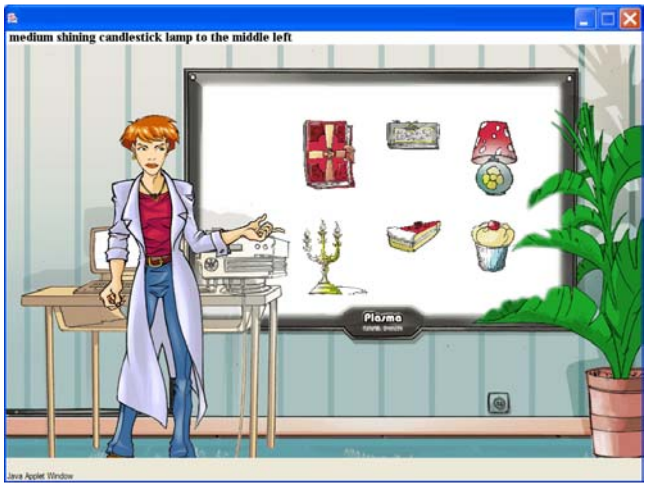

Figure 3. Display generated from a random selection of attributes via a specified set of modalities (spoken utterance: "medium shining candlestick lamp to the middle left").

Another possibility that we will investigate, and which could produce different results, is to first select 70% of the attributes and then display each of them on each specified modality.

3.2.2 Controlling the selection of modalities

We have developed the symmetrical specification and algorithm in which the attributes are fully specified but the modalities are randomly selected:

```
   <complemredundancy>
   <idObj>0</idObj>
   <modalitysalience>50</modalitysalience>
      <attributeset>
         <attribute>name</attribute>
         <attribute>size</attribute>
         <attribute>shape</attribute>
         <attribute>color</attribute>
      </attributeset>
   </complemredundancy>
```

3.2.3 Random selection of attributes and modalities

In the last version of the specification and algorithm, both the random selection of attributes and modalities can be controlled:

```
<complemredundancy>
  <idObj>2</idObj>
  <attributesalience>60</attributesalience>
  <modalitysalience>50</modalitysalience>
</complemredundancy>
```
Yet, the lack of constraints in the selection of attributes and modalities may produce behaviors which may look weird (such as a purely non-verbal behavior in Figure 4).

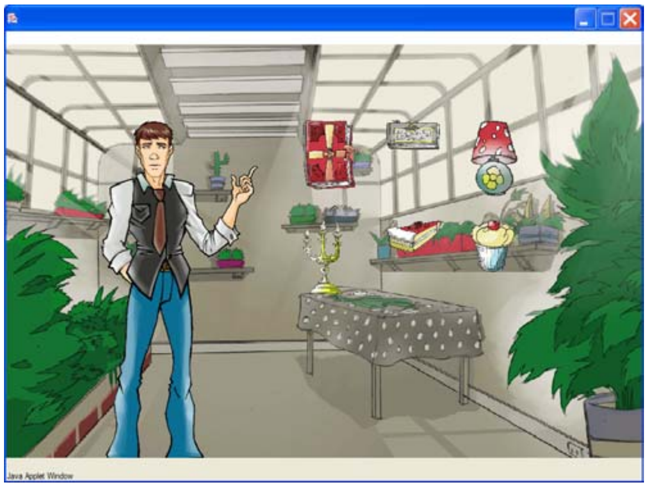

Figure 4. Display generated by the specification defined in section 3.2.3. Due to the lack of constraints in the specification of modalities and attributes, the randomly generated behavior is only non-verbal.

4. CONCLUSIONS AND FUTURE DIRECTIONS

We have introduced a new XML language for specifying the cooperations between output modalities to be used by an ECA as well as the algorithms used to parse such descriptions and generate the corresponding low-level animations for 2D cartoon agents. We believe that such an intermediate level of specification can be useful to ease the design of "believable" agents in which a direct mapping between high-level of abstraction and low-level is too complex to achieve with a single step. In some projects, some random behavior is brought into play in ECA design for example to solve conflicts in the mapping between emotional state and facial expressions [5] but not for selecting both the information to communicate and the modalities to be used based on values for controlling redundancy and complementarity rates. Evaluation of manually specified similar cooperation have been achieved for more complex examples than references to objects (e.g. technical presentations) [2]. In these experiments, we compared the effect of a redundant vs. a complementary individual agent's behavior on the user. Two other male cartoon-like agents have been designed for this purpose. Yet, the automatic generation (and its evaluation) of low-level specification from the specification of cooperation including randomised modality allocation remains to be done for these complex examples, as well as the generation of intermediate representations from high-level specifications related to emotion and communicative acts.

We will continue to investigate this specification and processing of cooperations between output modalities in several directions including their use for 3D agents, the temporal organization of several generated configurations, the specification of other kinds of cooperation than redundancy and complementarity such as the possibility for the agent to switch between several modalities (equivalence), the addition of some constraints on the required use of some modalities for some attributes, and finally their possible integration within a multimodal natural language generator for higher interactive situations such as chatting with the agent.

5. ACKNOWLEDGMENTS

This work was developed at LIMSI-CNRS and supported by the EU / HLT funded project NICE (IST-2001-35293) http://www.niceproject.com/.

6. REFERENCES

[1] Arafa, Y., Kamyab, K., Mamdani, E., Kshirsagar, S., Magnenat-Thalmann, N., Guye-Vuillème, A., and Thalmann, D. Two approaches to Scripting Character Animation in [8].

[2] Buisine, S., Abrilian, S., and Martin, J.-C. Evaluation of individual multimodal behavior of 2D embodied agents in presentation tasks in proc. of Workshop "Embodied conversational characters as individuals", Marriot, A., Pelachaud, C., Ruttkay, Z. (Eds), 2nd Int. Joint Conf. on Autonomous Agents & Multiagent Systems (AAMAS'03), Melbourne, Australia, 2003.

[3] De Carolis, B., Carofiglio, V., Bilvi, M., and Pelachaud, C. APML, a Markup Language for Believable Behavior Generation in [8].

[4] Marriot, A. and Stallo, J. VHML - Uncertainties and problems... A discussion in [8].

[5] Pelachaud, C. and Poggi, I. Subtleties of facial expressions in embodied agents. The Journal of Visualization and Computer Animation. Special Issue: Graphical Autonomous Virtual Humans. Issue Edited by Daniel Ballin, Jeff Rickel, Daniel Thalmann., vol. 13 (5), pp. 301-312, 2002.

[6] Piwek, P., Krenn, B., Schröder, M., Grice, M., Baumann, S., and Pirker, H. RRL: A Rich Representation Language for the Description of Agent Behaviour in NECA in [8].

[7] Prendinger, H., Descamps, S., and Ishizuka, M. Scripting affective communication with life-like characters in web-based interaction systems. Applied Artificial Intelligence, vol. 16 519-553, 2002.

[8] Proc. of Workshop on "Embodied conversational agents - let's specify and evaluate them!", 1st Int. Joint Conf. on "Autonomous Agents & Multi-Agent Systems" (AAMAS'02), Bologna, Italy, 2002