

# Synthetic Characters as Multichannel Interfaces

Elena Not, Koray Balci, Fabio Pianesi and Massimo Zancanaro

ITC-Irst

Via Sommarive

38050 Povo-Trento, Italy

tel. +39-0461314567

{not,balci,pianesi,zancana}@itc.it

## ABSTRACT

Synthetic characters are an effective modality to convey messages to the user, provide visual feedback about the system internal understanding of the communication, and engage the user in the dialogue through emotional involvement. In this paper we argue for a fine-grain distinction of the expressive capabilities of synthetic agents: avatars should not be considered as an indivisible modality but as the synergic contribution of different communication channels that, properly synchronized, generate an overall communication performance. In this view, we propose SMIL-AGENT as a representation and scripting language for synthetic characters, which abstracts away from the specific implementation and context of use of the character. SMIL-AGENT has been defined starting from SMIL 0.1 standard specification and aims at providing a high-level standardized language for presentations by different synthetic agents within diverse communication and application contexts.

## Categories and Subject Descriptors

H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems – *animations*.

## General Terms

Management, Standardization, Languages.

## Keywords

Synthetic Characters, Multimodal Presentations, SMIL.

## 1. INTRODUCTION

Synthetic characters are often integrated in multimodal interfaces as an effective modality to convey messages to the user. They provide visual feedback about the system internal understanding of the communication and engage the user in the dialogue through emotional involvement. However, avatars should not be

considered as an indivisible modality but as the synergic contribution of different communication channels that, properly synchronized, generate an overall communication performance: characters can emit voice and sounds, animate speech with lips and facial expressions, move eyes and body parts to realize gestures, express emotions, perform actions, sign a message for a deaf companion, display listening or thinking postures and so on.

If we interpret metaphorically a synthetic character as a multimodal output interface, we clearly get the picture of a set of modalities or *performance abilities* (e.g. speech, speech animation, sign animation, gesture, body motion, body posture, etc.) that are to be realized by the available *communication channels* (e.g., voice, face, eyes, mouth, body, arm,...). To some extent, this is similar to what happens with standard multimedia (and multimodal) presentations, where different modalities, like audio, images or text are performed independently on different devices (or channels) and properly synchronized to obtain the desired communicative effect (Figure 1).

| Multimodal interface |                | Synthetic character   |          |
|----------------------|----------------|-----------------------|----------|
| Output modalities    | Devices        | Performance abilities | Channels |
| -Prerecorded audio   | -Headphones    | -Speech               | -Voice   |
| -Synthetic speech    | -Loudspeakers  | -Speech animation     | -Face    |
| -Text                | -Screen        | -Sign animation       | -Eyes    |
| -Images              | -Haptic device | -Gesture              | -Mouth   |
| -Movies              | -...           | -Motion               | -Body    |
| -Animations          |                | -Posture              | -Head    |
| -touch               |                | -...                  | -Arm     |
|                      |                |                       | -Hand    |
|                      |                |                       | -...     |

Figure 1. Parallel view of multimodal interfaces and synthetic characters.

This view of synthetic characters as the integration of different communication abilities completely abstracts away from the specific communicative context in which the character is integrated (be it a human-character dialogue with emphatic involvement or an impersonal information presentation linked to an html page). Equal importance is assigned to the various communication modalities and channels available to the character so that efforts can be focused on the representation of their parallel or sequential interaction during the presentation. (In particular, we do not always assume the presence of a sentence to be uttered and with respect to which the overall animation is to be synchronized.) This approach helps maintain an application/task

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ICMI'05, October 4-6, 2005, Trento, Italy.

Copyright 2005 ACM 1-59593-028-0/05/0010... \$5.00.

independent view while identifying the requirements for a scripting language for synthetic characters that aims at standardization and high reusability.

In designing SMIL-AGENT, a representation and scripting language for life-like characters, we started right from this perspective of generality. Joining the valuable discussion emerged in the research field of Embodied Character Agents about the standardization of scripting languages (see for example [16], [12]), we decided to focus our efforts on the modeling of multichannel virtual characters, inspiring our approach to the generality and simplicity guidelines that support the standard language for multimodal presentations, SMIL<sup>1</sup>.

In the next section of this paper we justify the main motivations of our multichannel approach, whereas section 3 details more specifically the design solutions adopted in SMIL-AGENT, together with implementation issues related to the realization of SMIL-AGENT players.

## 2. CHARACTERS AS MULTICHANNEL INTERFACES

The idea of explicitly modeling synthetic characters as the synergic contribution of separate communicative channels acts as the basis for many agent scripting languages described in the literature (e.g., CML, AML [1], MPML[14]). In fact, apart from languages that aim at a fine-grain modeling of discourse facets, with script centered around the verbal message (e.g., APML [5]), most existing languages do consider at least a distinction between the audio and visual modality, or a tripartite distinction between voice, face and body channels, with final synchronization of the separate performance contributions via <seq> or <par> statements that define the sequential or parallel integration.

In this paper, we argue for an approach that generalizes over this channel separation and back synchronization, without imposing any constraint on the set of available communicative channels and associated performance abilities.

### 2.1 Independence from communication context

Synthetic agents may be employed either in very complex communication scenarios (e.g., in human-computer dialogues aimed at problem solving, tutoring, advising), or in adaptive multimodal presentations, and also in simple, canned, information presentations linked to html pages, possibly manually written by a human author. It is therefore important for the scripting language to be sufficiently high-level and independent from specific communication contexts. In some existing languages, instead, the scripts may also contain information about the state of the external dialogue (MPML [13], VHML [8]), or information about what caused an emotion to be displayed by the agent (CML [1]), or the alternation of performance actions by different agents (MPML [13]). However, this information requires that the script interpreter and the character(s) used for the presentation specifically support the processing of this “external” data.

More portability and standardization can be obtained by allowing the scripting language to represent only the details of the character

behaviour: all data related to the dialogue management, or the integration of the agent within a larger multimodal presentation should be orchestrated by other external modules. This is exactly what happens with multimodal presentations scripted in the SMIL language: self-contained scripts, which do not contain application- or task-dependent information, may be either written manually by a human author or be computed by an external multimodal presentation/dialogue system.

### 2.2 Independence from specific character

Some scripting languages take for granted that the presentation will be played by a specific class of characters (e.g., MPML makes reference to the Microsoft Agent package) and rely on their performance abilities. Some of them also allow for low level directives with the advantage of sophisticated control over FAPs, BAPs<sup>2</sup> or keyframes. However, this dependence to the underlying implementation approach of the character significantly reduces portability.

The most promising solution to this problem seems the introduction of an explicit declaration of the agent configuration referred to in a certain script. In particular, the language should allow to:

- describe the communicative channels to be used in the presentation script;
- describe the performance abilities supported by the channels.

Figure 2 shows two sample synthetic faces (developed with the Xface support tools [2]) with their communication abilities.

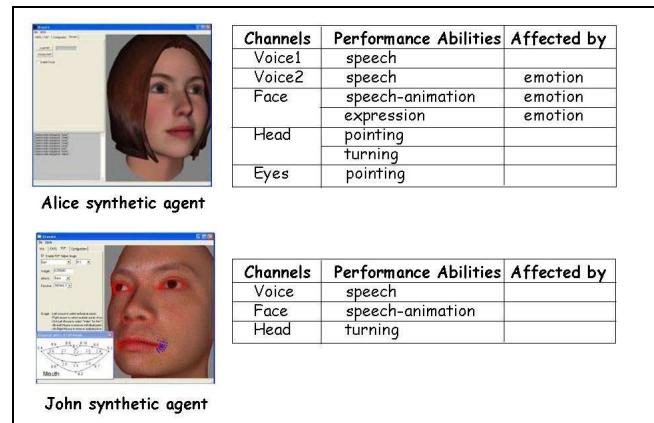


Figure 2. Sample agent communication abilities

Synthetic characters that match the agent configuration specified in the head of the script will be good candidates for realizing the scripted performance. In this way, scripts are not written with just one specific character implementation in mind, but are written for characters that support certain communication abilities. For the sake of language extendibility, the list of possible types of communicative channels (e.g., voice, face, eyes, mouth, body, arm,...) and of performance abilities (e.g., speech, speech

<sup>2</sup> Facial Animation Parameters and Body Animation Parameters used in MPEG-4 for parametrizing deformations on a face or a body with lesser parameters.

<sup>1</sup> <http://www.w3.org/AudioVideo/>

animation, gesture, body motion, message signing, ...) should be left open, so that new values can be added to write scripts for characters that support enhanced communication abilities<sup>3</sup>. For the same reason, flexibility should be allowed in the listing of features that may potentially influence the behaviour of the various communication channels: emotion, personality, culture, role of the character, etc...

### 2.3 Channel independence

In scripting languages oriented to the sophisticated control of dialogue and discourse facets (like for example APML [5][9]), the strict interrelation between performance abilities like speech and speech animation is assumed, with the evident benefit of facilitating the marking of discourse portions that need to be emphasized both in the audio and visual modality<sup>4</sup>. See for example the following APML sample script taken from [11].

```
<APML><turn-allocation type="take">
<performative type="inform" certainty="certain">
<belief-relation type="gen-spec">This is </belief-relation>
a spasm of <deictic obj="chest"> chest</deictic></performative>
<belief-relation type="cause-effect"> resulting from
<performative type="inform" certainty="certain">
overexertion when heart is diseased.</performative></belief-relation>
</turn-allocation></APML>
```

Figure 3. Sample APML script

This strict coupling of speech and speech animation, however, imposes specific computational requirements to the synthetic character implementation: the effective animation of the face requires knowledge about the discourse structure and the actual visual rendering of the rhetorical relations defined in the Rhetorical Structure Theory ([6]).

For the sake of generality (to avoid strict dependence of the animation software from the specific discourse theory), it might be desirable to move some of the rendering decisions for animation up in the discourse module that automatically generates the script. In this way, a synthetic body would be simply instructed to display a certain expression/posture at a certain time of the performance, without imposing to the character the knowledge, for example, of the cause-effect rhetorical relation that justifies it.

More in general, other types of channel interdependencies are allowed in existing scripting languages. See for example the interlaced use of voice, face, and eyes channels in the VHML script taken from [7] (Figure 4).

A great improvement in terms of portability may be attained by distinguishing in the script the description of each single channel from the specifications of the channels' synchronization [10]. (A practical example is discussed in following section 3.2.)

<sup>3</sup> One character might allow the fine grain control of body parts to realize sophisticated types of gesturing decided by the script author. Other characters might just allow generic directives to the overall body.

<sup>4</sup> For example, a "large" adjective appearing in the message to be animated might need to be stressed by the speech synthesizer as well as emphasized by raising eyebrows and opening eyes.

```
<?xml version="1.0"?>
<!DOCTYPE vhtml SYSTEM `./vhtml.dtd`>
<vhtml>
<p>
<surprised wait="1s" intensity="80"> Wow! <break/>
I didnt expect that to happen <eye-blink wait="300ms"/>
<eye-blink wait="300ms"/><break size="large"/>
it took me by surprise</surprised>
</p>
<p target="1">
<embed type="html" src="example_surprised.html"/>
</p>
</vhtml>
```

Figure 4. Sample VHML script

Furthermore, a finer-grain control of emotions can be gained over the alternative communicative modalities. E.g., (i) it is possible to express different intensities of emotion to be rendered by the voice and the facial expression; (ii) it possible to express opposite emotions with different channels to get ironic or comic effects.

### 2.4 Adaptivity

Many scripting languages described in the literature come together with an automatic system for dialogue/discourse management that is able to generate scripts for synthetic character performances to be included in multimodal interactions. Scripts can therefore be automatically tailored according to the current dialogue context and discourse history, as well as to the features contained in the user model. However, character presentations find application also in less dynamic scenarios, e.g. in educational hypermedia, where scripts might be written in advance by domain experts. The recent advances of the Adaptive Hypertext and Hypermedia research field [3] demonstrate, in fact, the need of effective languages and tools for the encoding and processing of pre-prepared material to be flexibly recomposed. For example, AHA! [4] provides a rich language for annotating optionality in HTML and SMIL presentations and XASCRIP [17] provides a language to describe conditionally simple multimedia presentations<sup>5</sup>.

In a similar way, a scripting language for synthetic characters of large use should allow the definition of optional alternative parts of the presentation to be selected according to user features or interaction context.

### 2.5 Standardization and extendibility

Standardization is essential for a wide spread of a representation and scripting language. Many of the initiatives described in the literature to define effective scripting languages for synthetic agents take SMIL as a sort of point of reference, especially to define the modality synchronization model. Indeed, SMIL seems a good baseline because it has been explicitly designed for "integrating a set of independent multimedia objects into a synchronized multimedia presentation" by specifying the temporal behaviour of a multimedia presentation and the layout of the presentation on a screen [18]. As it will be described in following section 3, we propose to strictly adhere to the (widely agreed upon) SMIL representation and scripting approach, to sprout a

<sup>5</sup> Although both these languages are processed in the backend, there is no reason why in principle they could not be processed directly by the client.

new “dialect” for integrating a set of communication channels into a synchronized character presentation.

As explained in section 2.1 above, however important standardization is, an effective scripting language should also be modular to allow easy introduction of new communication modalities, as well as new possible values for the supported emotions, personalities, moods, cultures, user profiles,... This requirement is extremely important if the language is to be reused with characters with different features and abilities. We can envisage four profiles of people who might be involved with the definition, extension and use of an open scripting language:

- Language developer: Understands the inner rationale of the language and defines its syntax and semantics.
- Player implementer: Understands the inner rationale of the language and implements one or more channel players according to the language specification. For those features of the language that are underspecified, the implementer defines platform/character dependent behaviour (e.g., how the player should behave in case the script contains multiple directives for the same channel referred to the same time interval) as well as the communication protocol for interoperability with other applications (e.g., how the player for the synthetic agent can be integrated in a wider dialogue module or can be plugged in a web browser).
- Expert author: Identifies the set of characters and channel players to be used in a certain organization and describes the available resources in formal channel description scripts. The expert author might also extend the list of attribute values for emotions, speech-acts, actions, languages,... supported by the available channels.
- Author: Given a set of available channels identified and described by the expert author, writes scripts for animated agent presentations.

All the four types of users should be supported with proper methodologies and tools.

### 3. SMIL-AGENT

In this section, we present SMIL-AGENT, a representation and scripting language that is intended to play as a sort of SMIL dialect for the specification of information presentations by a synthetic agent (as an additional type of media object). SMIL-AGENT allows integrating into a synchronized presentation a set of independent performance directives to be realized by different communicative channels of a synthetic agent.

In designing SMIL-AGENT, our initial intent was the specification of a scripting language that would enhance the portability and reuse of Xface, a set of open tools for the creation of MPEG-4 based 3D Talking Heads [2]. The sophisticated deformation and animation of the surface of the face supported by Xface allows the output of believable emotions and expressions, coupled with the use of state-of-the-art expressive speech synthesizers.

However, we soon realized that, although Xface assures high integration in diverse communication scenarios, an essential requirement for our language was the independence from the

actual character and communication context in which the scripting language is used, as discussed in sections 2.1 and 2.2 above. We thus decided to strictly mirror SMIL 1.0 specification [18] and include modifications/extensions to the baseline syntax and semantics of the language just when required by the specifics of driving the behaviour of a synthetic character. The SMIL concept of *media objects* (i.e., audio, img, video, animation, text, textstream, ref) has been replaced by the notion of *agent performance elements* (i.e., speech, speech-animation, sign-animation, expression, action, song,...). In a similar way, instead of specifying in which region of the screen a certain media object should be displayed and when, SMIL-AGENT allows to define which communication channel should be used to realize a certain agent performance element and when performance should start.

### 3.1 Extendibility in SMIL-AGENT

To comply with the extendibility requirement argued in section 2.5, SMIL-AGENT formal syntax specification defines a separate language partition of attribute values that can be extended by expert authors according to the actual communicative channels and performance abilities supported by a certain synthetic agent. In practice, this is realized by a separate dtd file collecting the list of possible values for: (i) available types of communicative channels; (ii) supported emotions, speech acts, actions, languages; (iii) features that can be tested for adaptivity. See for example Figure 5.

```
<!-- specification of possible communicative channels -->
<ENTITY % channel
"      (anthropomorphic-agent | cartoon-agent | head | voice |
       face | eyes | mouth | torso | arms | chest | hands) ">

<!-- specification of possible emotions -->
<ENTITY % emotion
"      (fear | anger | sadness | happiness | disgust |
       surprise | sorry-for) ">

<!-- specification of possible user preferences to be tested in
switch statements -->

<ENTITY % channel-supported-test
"      (user-profile | user-expertise | personality |
       culture) ">

<ENTITY % user-profile
"      (child | adult) ">

<ENTITY % user-expertise
"      (naive | average | expert) ">

<ENTITY % user-test-attribute "
       user-profile      %user-profile;      #IMPLIED
       user-expertise    %user-expertise;    #IMPLIED
">
```

Figure 5. Sample values that can be edited by expert authors.

Expert authors can edit this “open” part of the language specification. However, how the various attributes and their values are actually used in scripts is strictly defined by the language syntactic and semantic specification, therefore guaranteeing the consistency of scripts and their correct parsing and interpretation.

### 3.2 Channel independence

As discussed in section 2.3, a scripting language that aims at standardization and reusability with different synthetic agents should not assume strict channel interdependences, given that their treatment is highly dependent on character implementation. Still, however, a declaration of available channels and respective

communication abilities to be used in a certain script is required for consistent parsing and interpretation. SMIL, to which SMIL-AGENT is inspired, provides for a declaration in the head of the script of the layout of the presentation, e.g. the structure of the regions where the media objects (image, video, animation,...) will be allocated. Similarly, SMIL-AGENT provides for an agent-layout declaration defining which communication channels will be used to allocate the performance directives of the character presentation. See Figure 6 for a sample agent configuration.

```
<smil-agent>
  <head>
    <agent-layout>
      <visual-layout background-color="maroon" width="250"
        height="230"/>
      <channel id="agent-body" title="John's complete body"
        type="anthropomorphic-body"
        src="http://i3p.itc.it/channels/john-body.xml"/>
      <channel id="agent-voice" title="Masculine Italian voice"
        type="voice"
        src="http://i3p.itc.it/channels/masculine-italian-voice.xml"/>
      <channel .../>
    </agent-layout>
  </head>
  <body>
    ...
  </body>
</smil-agent>
```

**Figure 6. Sample agent configuration described in the head of a SMIL-AGENT script**

The supported performance abilities of each single channel can be saved in a separate file (which is referred in the channel element with a `src` attribute, as in Figure 6) or can be directly included in the head of the script (as in Figure 7).

```
<smil-agent>
  <head>
    <agent-layout>
      <visual-layout background-color="maroon" width="250"
        height="230"/>
      <channel id="agent-body" title="Alice's complete body"
        type="anthropomorphic-body">
        <supported-modality type="speech-animation"/>
        <supported-modality type="action"/>
        <supported-modality type="expression"/>
        <supported-modality type="song"/>
        <supported-affect type="happiness"/>
        <supported-affect type="sorry-for"/>
        <supported-affect type="anger"/>
        <supported-affect type="disgust"/>
        <supported-affect type="sadness"/>
        <supported-action type="pointing"/>
        <supported-action type="greeting"/>
        <supported-action type="turn-giving"/>
        <supported-action type="turn-taking"/>
        <supported-action type="nodding"/>
        <supported-test type="user-profile"/>
      </channel>
      ...
    </head>
    ...
  </smil-agent>
```

**Figure 7. Sample description of channel performance abilities**

Similarly, overall agent layout configurations can be saved in separate files to be referred to in the head of several scripts (Figure 8).

The actual channel independence is realized in the body of the scripts by keeping performance directives separate and integrating them back in parallel or in sequence, possibly with extra information about time synchronization.

```
<smil-agent>
  <head>
    <agent-layout
      title="Alice, English voice,
        body with signing capabilities"
      src="http://i3p.itc.it/agents/agent-config14.xml"/>
    </head>
    <body>
      ...
    </body>
</smil-agent>
```

**Figure 8. Reference to external agent configuration**

Figure 9 shows a fragment of SMIL-AGENT script for a dialogue turn taken from [11], and representing the animation of the sentence: "I'm sorry to tell you that you have been diagnosed as suffering from what we call angina pectoris, which appears to be mild."

```
<body>
  <par system-language="english">
    <speech channel="alice-voice" affect="sorry-for"
      type="inform" id="say-suffering-angina">
      <mark id="*1*">I'm sorry to tell you that you have been
        diagnosed as suffering from <mark id="*2*">what we call
        angina pectoris, <mark id="*3*">which appears to be mild.
    </speech>
    <seq channel="alice-face" >
      <speech-animation affect="sorry-for"
        content="say-suffering-angina"
        end="*3*" intensity="1.5"/>
      <speech-animation affect="positive"
        content="say-suffering-angina"
        fill="freeze"/>
    </seq>
    <action channel="alice-right-hand" action-type="pointing"
      content="say-suffering-angina" begin="*2" end="*3">
      <param>bust</param>
    </action>
  </par>
  ...
</body>
```

**Figure 9. Sample portion of SMIL-AGENT script body**

The SMIL-AGENT formalism is certainly less compact and less discourse-oriented than, for example APLM. However, it allows plenty of flexibility in expressing which channel should realize a certain performance directive. For example, given a synthetic agent with sophisticated control of body motion and various channels corresponding to different body parts, the pointing in the direction of the character's bust could be realized by a hand, a finger, a hand plus the head in synchronization, etc..., according to which channel is specified in the `<action>` element of the script (in the example in Figure 9, for example, the agent's right hand is used). Furthermore, alternative voices could be easily selected at different stages of the presentation, or the face could support a wider set of emotions than voice<sup>6</sup>.

<sup>6</sup> It is worth pointing out here that SMIL-AGENT is a language that is intended as a high-level scripting language, which relieves the script author from low level details about how the performance abilities of the various channels are actually implemented. This means, for example, that when the author chooses a female synthetic face with large lips from the repository of available communicative channels, he does not need to specify (and therefore to know) that the articulation of labiodentals for speech animation is different from that of women with small lips. The author will just specify `speech-animation` among the desired performance abilities (as in

At present, the formal SMIL-AGENT specification does not impose specific constraints about additional annotations of the text to be synthesized and included within `<speech>..</speech>` tags. A wise solution is to make reference to already existing standards for speech mark-up languages (e.g. SSML [22]).

Channel independence further allows for presentations without any spoken message: scripts can be written for dialogue turns simply containing facial expressions (possibly changing over time) and/or body movements. SMIL-AGENT can also be used to script presentations in sign language for deaf users. Figure 10 shows how portions of SMIL-AGENT scripts might look in this case. In this example, we are assuming that the synthetic agent is able to accept as input a message written with the SignWriting formalism transcribing ASL (American Sign Language)[20]. The sentence represented here is “Cinderella lives with her stepmother and two stepsisters.” and has been taken from: <http://signwriting.org/library/children/cinderella/cind01.html>. Other transcription formalisms might be acceptable as well ([21]).

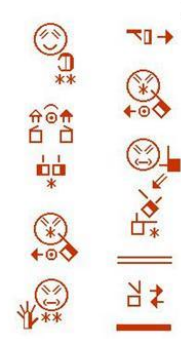
```
<sign-animation channel="alice-agent" affect="sadness"
  system-language="ASL" type="inform"
  id="cinderella-story">

</sign-animation>
```

Figure 10. How to instruct a character to sign a message.

Similarly to SMIL, SMIL-AGENT comes with the specification of a time model that details how the temporal synchronization of elements included in `<seq>` and `<par>` constructs should be implemented, providing a formal description of how the implicit, desired and effective begin/end of each element should be computed (see [8] for more details). Note, however, that even if the language provides means to control the timing of gestures or expressions, nothing is said in SMIL-AGENT on how the inter-articulatory phasing between gestures or expressions should be, given this is highly dependent on the actual implementation of the single characters.

### 3.3 Adaptivity in SMIL-AGENT

Even though the automatic generation of scripts by dialogue management modules or multimodal generation components is certainly the optimal solution, as discussed above there are many application scenarios where the manual authoring of scripts is acceptable. SMIL syntax already includes a `switch` statement that allows the definition of optional parts of a script to be

Figure 7), leaving to the animation engine the burden of correctly dealing with the speaker-specific features.

selected according to the run-time value of system parameters (e.g. output language, platform, screen size,...). In SMIL-AGENT we extended the powerfulness of `switch` to improve the adaptivity potential of the language. The existing syntax and semantics of the `switch` construct is preserved. However, expert authors are granted the possibility of defining new test parameters<sup>7</sup>. For example, given a synthetic character whose behaviour can vary according to the viewer profile, the expert author may add “user-profile” to the list of attributes that can be tested in `switch` statements, also specifying the values that the attribute can take<sup>8</sup>. Figure 11 shows how optional contents, to be synthesized with alternative voices, can be defined according to system- and user-defined parameters (language and user profile, in this case).

```
<body>
  <switch>
    <speech channel="alice-italian-voice" affect="happiness"
      id="say-mummy-cake" system-language="italian"
      type="inform" repeat="2">
      La mamma ha fatto una grande torta.
    </speech>
    <speech channel="alice-childish-voice" affect="happiness"
      id="say-mummy-cake" system-language="english"
      type="inform" user-profile="child" repeat="2">
      Mummy has made a big cake.
    </speech>
    <speech channel="alice-voice" affect="sadness"
      id="say-mummy-cake" system-language="english"
      type="inform" user-profile="adult" >
      Mummy has made the applecake again.
    </speech>
  </switch>
  ...
</body>
```

Figure 11. Sample optional choices in the script body.

The SMIL-AGENT specification does not impose any restrictions on how parameter values are communicated from the user to the SMIL-AGENT player. In the simplest scenario, the playback engine may use an explicit dialogue box to query the user before the presentation starts. But in more sophisticated communication scenarios input parameters may be imported from an external user model maintained by some other application.

Consistently with what happens in SMIL, in SMIL-AGENT scripts the `switch` statement can also be inserted in the head to define alternative `agent-layout` configurations to realize the presentation described in the body. This may be useful to adapt the expressive capabilities of the character to the user abilities or preferences: e.g., at presentation time an English speaking character may be selected according to the user mother tongue as in Figure 12, or a character with the ability to communicate with the sign language may be selected for a deaf user.

<sup>7</sup> Other approaches to the extension of SMIL for adaptivity feature, instead, the introduction of special tags that need to be properly processed, as happens for example in the AHA! system where tags like `<ref src=".." type="aha/text"/>` need a special treatment to determine the actual material to be included in the presentation [4].

<sup>8</sup> Practically, this means extending the list of pairs `<attribute-name, possible-values>` for the entity `user-test-attribute` shown in the dtd portion in Figure 5.

```

<head>
<switch>
  <agent-layout system-language="english">
    <visual-layout ... />
    <channel id="agent-body" title="Alice's complete body"
      type="anthropomorphic-body"
      src="http://i3p.itc.it/channels/alice-body.xml"/>
    <channel id="agent-voice" title="Female English voice"
      type="voice"
      src="http://i3p.itc.it/channels/fem-en-voice.xml"/>
    <channel ... />
  </agent-layout>
  <agent-layout system-language="italian">
    <visual-layout ... />
    <channel id="agent-body" title="John's complete body"
      type="anthropomorphic-body"
      src="http://i3p.itc.it/channels/john-body.xml"/>
    <channel id="agent-voice" title="Masculine Italian voice"
      type="voice"
      src="http://i3p.itc.it/channels/mas-it-voice.xml"/>
    <channel ... />
  </agent-layout>
</switch>
</head>

```

Figure 12. Sample optional choices in the script head.

### 3.4 Implementing players for SMIL-AGENT

The main leitmotiv that guided the syntactic and semantic specification of SMIL-AGENT was *generality*, a fundamental prerequisite for standardization: The language had to be general enough to allow for the implementation of players for various synthetic characters and to be integrated in various types of applications. However, implementing players for SMIL-AGENT is obviously more challenging than implementing a player for SMIL<sup>9</sup>, given that more knowledge and skill is available in the community on how to manipulate well consolidated media object formats (like for example, wav, au, mp3, jpg, gif, mpeg,...) than on how to morph over keyframes or FAPs, how to synchronize phonemes and visemes, or how to blend emotions and visemes.

Obviously, given that the set of possible channels and performance abilities is left open in the language to accommodate to future developments in the field of synthetic characters, it is not possible to implement a unique SMIL-AGENT player universally working with every agent platform. Rather, a series of players should be implemented to support classes of characters based on similar technology. For example, at our institute there is an ongoing effort for the implementation of a player for MPEG-4 based synthetic faces which support: speech, speech-animation, affective facial expressions, gestures, head movements. As shown in Figure 13, for the sake of modularity, the player will include a core SMIL-AGENT2FAP processing submodule to which different synthesizers (to get different languages or voice quality) and facial animation players can be plugged in (in the figure, the XfacePlayer and LUCIA<sup>10</sup> players are taken as examples).

Visual speech, emotions and expressions are treated as separate channels where the timing is driven by the visual speech to be synchronized with the audio. For each channel, a sequence of FAPs (or morph targets) are created and then blended. At the moment, for blending these channels, our player implementation uses a methodology derived from [14].

<sup>9</sup> Many players and authoring tools for SMIL scripts are commercially available. See <http://www.w3.org/AudioVideo/> for an updated list.

<sup>10</sup> <http://www.pd.istc.cnr.it/LUCIA/home/default.htm>

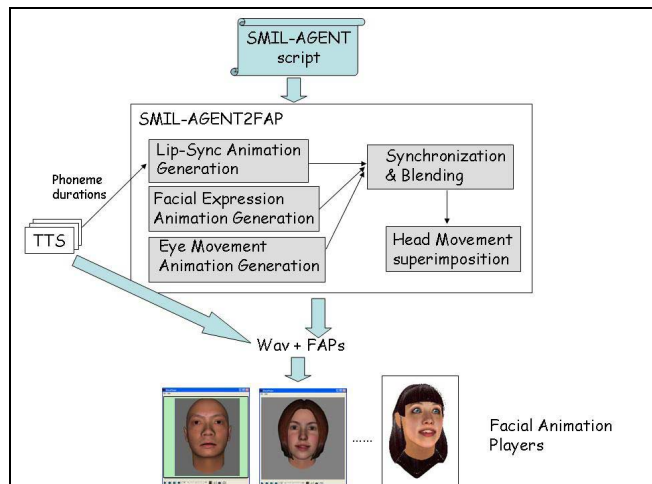


Figure 13. Sample processing of SMIL-AGENT scripts for MPEG-4 based synthetic faces.

## 4. CONCLUSION

The field of animated characters is rapidly maturing, with an emerging need of standardized languages and tools to help the effective integration of characters into the most diverse applications. This paper proposes SMIL-AGENT as a representation and scripting language for synthetic characters, which abstracts away from the specific implementation and context of use of the character. SMIL-AGENT has been defined starting from SMIL 0.1 standard specification and aims at providing a high-level standardized language for presentations by different synthetic agents within different communication and application contexts.

With respect to other existing scripting languages, SMIL-AGENT generalizes the concept of *separate representation* of the various communication modalities of a synthetic character (e.g., voice, speech animation, sign animation, facial expressions, gestures,...) and their explicit interleaving in the presentation performance. This helps script authors to write performance descriptions which are not bound to a specific character implementation. Furthermore, SMIL-AGENT explicitly *abstracts away* from all data related to the dialogue management and the integration of the agent within larger multimodal presentations, thus assuring the portability of the language (and of the synthetic characters supporting it) to different task and application contexts.

Admittedly, the success of SMIL-AGENT greatly depends on the implementation of effective players for the sophisticated synthetic characters being developed in the ECAs field. At ITC-irst we are currently working on a player for MPEG-4 based synthetic faces. SMIL-AGENT should be appealing to other research groups as well, given its propensity to standardization and its extendibility, which allows adding to the language new performance abilities or control features as requested by increased levels of sophistication in synthetic characters animation.

## 5. ACKNOWLEDGMENTS

SMIL-AGENT has been developed within the Intelligent and Interactive Information Presentation (i3p) group of the Cognitive and Communication Technologies Division of ITC-irst. The

working group for SMIL-AGENT includes: Koray Balci, Marco Guerini, Nadia Mana, Elena Not, Fabio Pianesi, and Massimo Zancanaro. We gratefully acknowledge everybody's contribution in the specification of the language. Special thanks to Dan Cristea from the University of Iasi for his valuable comments and suggestions during the first stages of SMIL-AGENT design. Finally our gratitude to the anonymous reviewers for the many fruitful and thorough suggestions.

## 6. REFERENCES

- [1] Arafa, Y., Kamyab, K. and Mamdani, E. Toward a Unified Scripting Language: Lessons Learned from Developing CML and AML. In H. Prendinger and M. Ishizuka (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004, 39-63.
- [2] Balci, K. Xface: MPEG-4 based open source toolkit for 3d facial animation. In *Proceedings of AVIO4, Working Conference on Advanced Visual Interfaces* (Gallipoli, Italy, 25-28 May, 2004).
- [3] Brusilovsky, P. (2001) Adaptive Hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110.
- [4] De Bra, P. and Stash, N. Multimedia Adaptation Using AHA! In *Proceedings of the ED-MEDIA 2004 Conference* (Lugano, Switzerland, June, 2004), 563-570.
- [5] De Carolis, B., Carofiglio, V., Bilvi, M. and Pelachaud, C. APMML, a Markup Language for Believable Behavior Generation. In *Proceedings of the Workshop on "Embodied conversational agents – let's specify and evaluate them!"* (held in conjunction with AAMAS02, Bologna, Italy, 2002).
- [6] Mann, W. and Thompson, S. *Rhetorical Structure Theory: A Theory of Text Organization*. Technical Report, USC/Information Sciences Institute, Marina del Rey, CA, ISI/RS-87-190, June, 1987.
- [7] Marriott, A. and Beard, S., gUI: Specifying Complete User Interaction. In H. Prendinger and M. Ishizuka (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004, 111-134.
- [8] Marriott, A. and Stallo, J. VHML- Uncertainties and Problems... A discussion. In *Proceedings of the Workshop on "Embodied conversational agents – let's specify and evaluate them!"* (held in conjunction with AAMAS02, Bologna, Italy, 2002).
- [9] Pelachaud, C., Carofiglio, V., De Carolis, B., de Rosis, F. and Poggi I. Embodied Contextual Agent in Information Delivering Application. In *Proceedings of First International Joint Conference on Autonomous Agents & Multiagent Systems, AAMAS02* (Bologna, Italy, 15-19 July, 2002).
- [10] Pianesi, F. and Zancanaro, M. La specificazione del comportamento di agenti conversazionali – problemi per un modello a delega ed una proposta alternativa. In *Proceedings of the Italy's Association of Cognitive Sciences (AISC) Second National Conference* (Ivrea, Italy, 19-2- March, 2004).
- [11] Poggi, I., Pelachaud, C., de Rosis, F., Carofiglio, V. and De Carolis, B. Greta. A Believable Embodied Conversational Agent. In O. Stock and M. Zancanaro (eds.) *Multimodal Intelligent Information Presentation*, Springer, 2005, 3-25.
- [12] Prendinger, H. and Ishizuka, M. (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004.
- [13] Prendinger, H., Descamps, S. and Mitsuru, I. MPML: a markup language for controlling the behavior of life-like characters. In *Journal of Visual Languages and Computing*, 15, 2004, 183-203.
- [14] Pyun, H., Chae, W., Kim, Y., Kang, H. and Shin, S. Y.. An example-based approach to text-driven speech animation with emotional expressions. Technical Report 200, KAIST, July 2004.
- [15] Saeyor, S. and Ishizuka, M. MPML and SCREAM: Scripting the Bodies and Minds of Life-Like Characters. In H. Prendinger and M. Ishizuka (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004, 213-242.
- [16] Rist, T. Some Issues in the Design of Character Scripting and Specification Languages – a Personal View. In H. Prendinger and M. Ishizuka (eds.) *Life-Like Characters. Tools, Affective Functions, and Applications*. Springer-Verlag, 2004, 463-468.
- [17] Rocchi C. and Zancanaro M. Template-based Adaptive Video Documentaries. In *Proceedings of AIMS2004 Workshop*, held in conjunction with UbiComp 2004. Nottingham, September, 2004.
- [18] *Synchronized Multichannel Integration Language for Synthetic Agents (SMIL-AGENT) 0.1 Specification*, ITC-irst - Technical report, T05-05-07, May 2005.
- [19] *Synchronized Multimedia Integration Language (SMIL) 1.0 Specification*. <http://www.w3.org/TR/REC-smil/> (accessed on 21<sup>st</sup> of April 2005).
- [20] Sutton, V., SignWriting Site. [www.signwriting.org](http://www.signwriting.org) (accessed on 29<sup>th</sup> of April 2005).
- [21] Streiter, O. and Vettori, C. (eds) *Proceedings of the Workshop on the Representation and Processing of Sign Languages*. Held in conjunction with LREC 2004, (Lisbon, Portugal, 30 May, 2004).
- [22] Speech Synthesis Markup Language (SSML) Version 1.0. <http://www.w3.org/TR/2004/REC-speech-synthesis-20040907/> (accessed on 27<sup>th</sup> of April 2005).