Scripting Embodied Agents Behaviour with CML: Character Markup Language

Yasmine Arafa Abe Mamdani
IIS, Department of EEE, Imperial College London
London, SW7 2BT, UK +44 (0) 20 75946319
{y.arafa & e.mamdani}@ee.ic.ac.uk

ABSTRACT
Embodied agents present ongoing challenging agenda for research in multi-modal user interfaces and human-computer-interaction. Such agent metaphors will only be widely applicable to online applications when there is a standardised way to map underlying engines with the visual presentation of the agents. This paper delineates the functions and specifications of a mark-up language for scripting the animation of virtual characters. The language is called: Character Mark-up Language (CML) and is an XML-based character attribute definition and animation scripting language designed to aid in the rapid incorporation of lifelike characters/agents into online applications or virtual reality worlds. This multi-modal scripting language is designed to be easily understandable by human animators and easily generated by a software process such as software agents. CML is constructed based jointly on motion and multi-modal capabilities of virtual life-like figures. The paper further illustrates the constructs of the language and describes a real-time execution architecture that demonstrates the use of such a language as a 4G language to easily utilise and integrate MPEG-4 media objects in online interfaces and virtual environments.

Keywords
Embodied Agents, Interface Agents, Lifelike Characters, Mark-up Languages, Automated Animation Scripting, CML, Animated Expression.

INTRODUCTION

The merits of embodied agents for different applications are a matter of current debate. Commercial attempts so far have been disappointing. Microsoft, for example, has been eagerly involved in efforts to drive multimodal interfaces with embodied representations, by developing the Microsoft Active Agent tool [4]. Other commercial companies like Semantics have now caught on and are developing similar tools [6]. However, while these tools are a relatively easy way to add some simple agent behaviour to web sites and applications and have proven to be useful for rapid prototyping, they remain very limiting to developers who require adding believability to the agents being used.

Although, there are several commercial and proprietary agent animation tools available as well as, several emotion engines or mechanisms that are available to generate and govern believable behaviour of animated agents, there is no common mechanism or API to link the underlying engines with the animated representations. The Character Mark-up Language (CML) is developed to bridge the gap between the available underlying engines and agent animation tools.

To set the scene, the paper first sets forth the motivations for such mark-up languages developed at the Intelligent and Interactive Systems section, Imperial College London; describes the key features and capabilities it offers; and discusses some technical issues raised based on our design and development experience on the ESPRIT project EP28831 MAPPA[1], and IST project IST-1999-11683 SAFIRA[2]. The paper further sets forth the key functionality that such description and scripting languages will need to succeed in animated agent interaction applications. Finally, an implemented architecture is briefly described.

EMBODIED AGENTS MARK-UP LANGUAGES

As parallel developments several animation tools and many emotion and behaviour models and their respective implementations have been produced. The appearance of these has highlighted the need for powerful yet generic scripting languages to bridge the gap between behaviour generation and animation tools. Notably Virtual Human Mark-up Language (VHML) [VHML] and Human Mark-up Language (HumanML) [2] are examples of these.

LANGUAGE REQUIREMENTS

The design and formalisation process of any language needs to fulfil a set of salient consideration defined so as to meet the criteria for general use and implementation. In addition to the set the functional purpose of the language.

The following key design criteria are identified for the development of CML:

IUI'03, January 12–15, 2003, Miami, Florida, USA.
ACM 1-58113-586-6/03/0001.

[1] MAPPA: Multimedia Access through Persistent Personal Agents
[2] SAFIRA Supporting Affective Interactions in Real-time Applications

High-Level: Abstracted from the low-level technology elements yet retains tags for low-level elements to allow detailed, flexible control

Complete**:** That includes properties related to the functional purpose of the language (Character properties, Believability attributes, Animation script

Extensible: Provisions for user-defined tags, and Complex elements, Share and reuse

Parameterised Action Support: Provision for customised, and Dynamically generated scripts

Synchronised: modality control, Merged animation

DYNAMIC SCRIPTING OF EXPRESSIVE BEHAVIOUR

Character Model

The character visual model is based on the MPEG-4 standard and includes two sub models for Face and Body. MPEG-4 defines two sets of parameters to describe and animate the character model: 1- Facial Animation Parameter Set (FAPS) and Body Animation Parameter Set (BAPS); and 2 - Facial Definition Parameter (FDP) and Body Definition Parameter (BDP). Base animation objects are created providing a generic virtual face and/or body with neutral expression and a default body posture. The shape of the virtual character is defined using the FDPs and BDPs, then facial and body expression animation is defined by FAPS and BAPS. Real-time generation of expression and animated behaviour is achieved by specifying the appropriate attribute values of the FAPS and/or BAPS for each frame.

MPEG-4s model-based approach to the animation generation and transmission of facial/body information makes it suitable for real-time online applications and the implementation of multi-modal interactive character human-computer-interaction metaphors. With the small size of the expression and animation parameter set, compression ratios, that are one to two orders of magnitude better than using classical video compression, can be achieved without losing crucial content (e.g. the facial expressions and lip movements). Moreover, achieving high-level functionalities (like movement or scaling) is straightforward since the 3D models of the face and body are known. Stereoscopic visualization of a talking head or body model is also achievable without any bit-rate overhead. Body animation leverages all the same principles as facial animation for coding body models and their compressed parametric control streams during a session.

Character Mark-up Language

Animated lifelike character representations of agents will only be widely deployed when there are real-time mechanisms to contextually map character and affect models to effected animated personifications. To approach this the SAFIRA project utilises the design elements of an architecture for including personality and emotional responsiveness in an interface agent; semantic abstraction and annotation of the knowledge being manipulated; and mapping resulting semantic understanding onto appropriate behaviour; which is translated into context-sensitive character traits by varying the planned response using variable emotion indicators.

CML is developed with an aim to bridge the gap between the underlying Affect and process engines, and agent animation tools. CML provides a map between these tools using objects by automating the movement of information from XML Schema definitions into appropriate relational parameters required to generate the intended animated behaviour. Allowing developers to use CML as a glue-like mechanism that ties various visual and behaviour generation tools together seamlessly, regardless of the platform that they run on and the language they are developed with.

CML: Objectives

The key objectives of CML are as follows:

1. Define a framework to decouple animation tools and the underlying affect and planning engines
2. Establish a formal specification for unified / consistent interpretation
3. Provision for modular development
4. Create a mark-up language based on XML that allows users/agents to provide semantic and scripting annotations for handling and animating embodied agents

CML: Definition

The term Character is used to denote a language that encapsulates the attributes necessary for believable behaviour. The intention is to provision for characters that are lifelike but are not necessarily human-like. Currently the attributes specified are mainly concerned with visual expression, although there is a limited set of specification for speech. These attributes include specifications for animated face and body expression and behaviour, personality, role, emotion, and gestures.

CML is both a Representation and Scripting language Representing embodied agent attributes, and Scripting the animation of their behaviour. CML expresses ontologies for: Embodied agents, Animation, Emotions and Personality. Overall, CML is intended as a formal mechanism enabling software agents to dynamically create, identify, understand and share characteristic information entities.

Visual Behaviour Definition

Classification of behaviour is governed by the actions an agent needs to perform in a session to achieve given tasks, and is influenced by the agent's personality and current mental state. A third factor that governs character behaviour is the role the agent is given. A profile of both an agent's personality and its role are used to represent the ongoing influences to an agent's behaviour. These profiles are user-

specified and are defined using XML annotation. Behaviours are defined as tags, which essentially group and annotate sets of action points generally required by the intended behavioural action. The CML processor will interpret these high-level behaviour tags, map them to the appropriate action point parameters and generate an animation script.

The Character Mark-up Language defines the syntactic, semantic and pragmatic character presentation attributes using structured text. CML is based on the definition XML Schema structures. The character mark-up-based language extends the descriptions for facial expressions used in the FACS system. FACS (Facial Action Coding System) defines a set of all the facial movements performed by a human face [1]. Although FACS is not an SGML-based language in nature, we use their notion of Action Units to manipulate expressions. Character gesture attribute definitions are based on the research and observations by McNeill [3] on human gestures and what they reveal.

Affective expression is achieved by varying the extent and degree values of the low-level parameters to produce the required expression. The CML encoder will provide the high-level script to be used in order to specify the temporal variation of these facial expressions. This script will facilitate designing a variety of time-varying facial expressions using the basic expressions provided by the database.

```
<character>
   <name />
   <personality {trait set Extravert/
                    Introvert/..} />
   <role {Helper/Salesman/...} />
   <gender M/F />
   <charType Human/Non-Human/... />
   <animationTool MPEG-4/LA/MSAgent/... />
   <fapFiles {set of file names} />
   <bapFiles {set of file names} />
   <ttsEngine Festival/ViaVoice/... />
   <voice {voiceId or wav files} />
   ...
</character>
```

CML Character Definition

```
<emotion>
   <type {neutral/anger/disgust/fear
           /joy/sadness/surprise} />
   <intensity {low/intermediate/high} />
   <decay {0.n to n.n(unit)/default/slow/fast}
   <target {user/self/agent/object/...} />
   <priority 0 to n />
   <preEState emotion />
   <transEState emotion />
</emotion>
```

CML Emotion Definition

Classification of Motion

Action composition script is generated by a CML processor (delineated in Figure-1) which blends actions specified with an input emotion signal to select the appropriate gestures and achieve the expressive behaviour. CML also provisions for the generation of compound animation script by facilitating the definition and parameterisation of sequences of base movements.

The chosen base set of movements allows basic character control (movement and interactions) as well as assures the capability to perform unlimited character specific animations. The interactions can involve other characters and objects that must be referenced by a valid id within the Graphics Engine.

Base Motions : The initial set of the CML base motions are classified by the goal of the motion into: Movement, Pointing, Grasping, Gaze and Gesture as follows:

Movement defines motions that require the rotation or movement of a character from one position to another. Positions are defined by exact coordinates, an object position or a character position. Movement elements are either move-to or turn-to.

Pointing defines a pointing gesture towards a coordinate, object or character. Pointing elements are *point-to*.

Grasping defines motions that require the character to hold, throw or come in contact with an object or another character. Grasping elements are *grasp*, *throw* and *touch*.

Gaze defines the movements related to the head and eyes. Gaze elements are gaze, track, blink, look-to and look-at. The gaze and track elements requires that only the eyes be moved or track an object or character. *look-to* and *look-at* require the movement of both head and eyes.

Gesture includes motions that represent known gestures like hand movements to convey an acknowledgement, a wave, etc. Gesture elements are *gesture* and *gesture-at*

CML Specification

CML defines a script like that used for a play. It describes the actions and sequence of actions that will take place in a presentation system. The script is a collection of commands that tell the objects in the world what to do and how to perform actions. The language is used to create and manipulate objects that are held in memory and referenced by unique output-ontology objects.

Animated character behaviour is expressed through the interpretation of XML Schema structures. These structure definitions are stored in a Schema Document Type Definition (DTD) file using XSDL (XML Schema Definition Language). At run-time character behaviour is generated by specifying XML tag/text streams which are then interpreted by the rendering system based on the rules defined in the definition file. Its objective is to achieve a consistent convention for controlling character animation models using a standard scripting language that can be used in online applications.

Synchronisation between the audio and visual modalities is achieved through the use of SMIL-Synchronised Multimedia Integration Language specification [5].

CBML- Character Body Mark-up Language

Character body description is a set of low-level tags based on MPEG-4 BAPs, and a set of high-level tags representing body parts which are grouped from a set of respective low-

level tags. CBML base elements of **Movement** (Moving, Turning); **Gesture** (Iconic, Symbolic, Deictic, Pointing); **Grasping**; and **Posture (expression).**

A set of high-level tags representing general body gestures:

Natural -Defines the characters default or normal posture state based on a distinct personality; ***Relaxed*** - Defines a relaxed posture state of the character; ***Tensed -*** Defines a tensed posture state of the character; ***Inclined -*** Defines the orientation and degree of leans towards or against an object.

CFML- Character Face Mark-up Language

Character face description is a set of low-level tags based on MPEG-4 BAPs, and a set of high-level tags representing facial parts which are grouped from a set of respective low-level tags. CFML base elements of **Head Movement** (Tilt, Turning); **Head Gesture** (Iconic, Deictic, Pointing); **Brow Movement**; **Mouth**; **Gaze** and **Expression.**

GENERATING SCRIPT

Script generation through to the effected animation process components consist of a set of MPEG-4-compliant facial and body models; high level XML-based descriptions of com-pound facial and body features; XML-based descript-ions of user-specified personality models; behaviour definit-ions, a CML processor, and finally a CML decoder. The general function of this component is delineated in Figure 1.

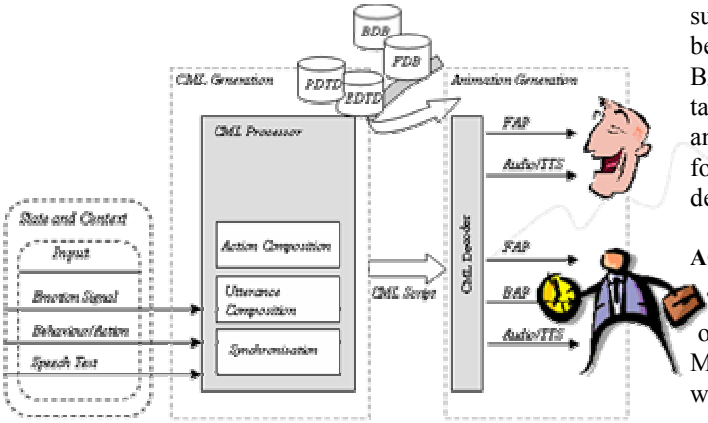

Figure-1 **Script Generation Function Abstract**

The architecture of an implementation generating and using CML is divided into three conceptual components of the supporting models and database for face and body animation, CML scripting and an animation rendering tool.

The script generation component assumes state and contextual input resulting for the underlying affective processing, planning, and domain knowledge-base engines. Based on these inputs and a defined character personality, the CML processor then generates the consequent synchronised behavioural action and utterance CML script. The script is then passed onto the CML decoder which parses the CML and maps its elements onto view-specific commands for final animation rendering.

SUMMARY AND CONCLUSIONS

The paper presents an approach to specifying scripting languages for character animation which is currently being developed and evaluated at Imperial College London. The CML took a top-down approach by defining high-level attributes for character personality, emotion and behaviour that are integrated to form the specification of synchronised animation script. New or unspecified behaviours are formed by blending together base elements and attributes thereby providing animators with the flexibility to generate animation script as required. CML provides a high-level generic mechanism for the selection and synchronised merging of animations while providing the flexibility for animators or script generation tools to define proprietary specifications based on the key elements provided plus any others that may be defined.

Forthcoming evaluation of each approach will further corroborate their merits. Our experience on the projects involved presented us with a possibility of a need to merge the specifications of both languages in a two-phase sequence for scripting character animation. Recent emergence of mark-up languages serving similar purposes have also exposed a profound need to agree on taxonomy for the affective and motion elements used, the granularity level of such definitions, what taxonomy for the mark-up tag definitions should be adopted, are current languages sufficient for the requirements of believable affective behaviour animation delivery?, whether Mpeg 4 FAPs and BAPs are sufficient or too granulated to provide the taxonomy required for real-time animation?, what affective and personality theories should be adopted to define the tags for affective expression?, what granularity of affective description is required?, as well as many more open issues.

ACKNOWLEDGMENTS

The requirement for and specification of CML was originally a contribution of the ESPRIT project EP28831 MAPPA. CML as described in this paper is developed within the European IST project IST-1999-11683 SAFIRA.

REFERENCES

[1] Ekman, P. and Rosenberg, E. L..What the Face Reveals: Basic &pplied Studies of Spontaneous Expression Using the Facial Action Coding System. Oxford Uni.Press, 1997.

[2] HumanML -http://www.humanmarkup.org/work/humanmlSchema.zip

[3] McNeill, D. Hands and Mind. The University of Chicago Press, 1992.

[4] Microsoft Corporation: Microsoft Agent Programming Interface Overview, 1998. http://www.microsoft.com

[5] SMIL - http://www.w3.org/AudioVideo/

[6] Semantics - http://www.semantics.com

-