A Conversational Avatar Framework for Digital Interactive StoryTelling

Ricardo Duarte, Efstathios Goudoulakis, Prof. Abdennour El Rhalibi, Prof. Madjid Merabti
R.L.Duarte@2010.ljmu.ac.uk, E.Goudoulakis@2008.ljmu.ac.uk, A.Elrlhalibi;M.Merabtij@ljmu.ac.uk
School of Computing and Mathematical Sciences
Liverpool John Moores University,
Byrom Street, L3 3AF, Liverpool, UK

Abstract– Digital Interactive Storytelling (DIS) is a relatively novel area of computer entertainment which aims at investigating interactive applications capable of generating consistent, emergent and rich stories. To provide new solutions for DIS, we have proposed a novel multi-agent DIS framework, DIEGESIS, which includes agents’ coordination and new planning and re-planning solutions for DIS. A natural development of DIS systems is to interface them with a 3D character animation framework, in which a 3D character would tell the story, or would be used for in-story character dialogues. In this paper, we propose to interface DIEGESIS with Charisma, our MPEG-4 based facial animation framework. We introduce DIEGESIS, explaining the planning algorithm for story generation, the story execution, and the agent coordination algorithms. We also introduce Charisma, a novel MPEG-4 facial animation framework for speech animation, which is also capable of recording and playback of MPEG-4 animation, alongside realistic dynamic lip sync animation based on Cohen-Massaro coarticulation model adapted to MPEG-4 facial animation (FA) specification. The preliminary experiments show that the coarticulation technique we have developed is suitable for storytelling and gives overall good and promising results when compared to related techniques.

Keywords-component; Interactive Storytelling, Planning Algorithms, Facial Animation, Speech Animation, MPEG-4, Coarticulation

I. INTRODUCTION

Planning systems have been used extensively in DIS for several years, however as we discussed in [1], there has been little research for planning algorithms from a DIS perspective.

The most common approach is to use a general Artificial Intelligence (AI) planning algorithm but, as we have advocated in [1], a new planning algorithm (combining features from existing algorithms with novel ideas) needs to be created specifically with DIS systems characteristics in mind.

To provide new solutions for DIS, we designed and are currently implementing and evaluating DIEGESIS, a novel multi-agent DIS framework, using planning and re-planning techniques. The use of constraints in the planning and re-planning solutions enables us to extend the expressiveness of the representation language we use, which is one of the most important suggestions for the new planning algorithm we proposed in [1].

Our proposal for re-planning aims to interleave plan generation and opportunistic plan execution, to re-plan when unexpected changes occur in dynamic environments whilst introducing minimal disruption to the original plan. In addition planning and re-planning for characters organised as a multi-agent system, requires managing and controlling the coordination of these characters actions.

In the facial animation research area, several developments during the past two decades have permitted the creation of synthetic-visual audio speech for virtual characters. While initial lip-sync studies attempted to concatenate phonetic segments, it was found that phonemes do not achieve their ideal target shape at all times, due to the influence of consecutive phonetic segments on each other, which results in phonetic overlap, and due changes that occur between segments, a phenomenon known as coarticulation.

In this paper we present the integration of our two frameworks, Charisma and DIEGESIS, to create a dynamic, efficient narration and storytelling system, which features can be summarised as follows:

- Interactive multi-agent framework capable to generate emergent narratives, featuring agents’ coordination techniques, a new planning, and a new re-planning algorithm.
- MPEG-4 facial animation specification compliant [2], capable of dynamic creation and playback of Facial Animation Parameter (FAP) stream animation.
- Provides an easy to use MPEG-4 FA editor and player in order to create, edit, test and fine-tune facial animation and speech synchronization resultant from the narrative in the DIS system.
- Supports the development of applications for a variety of platform, including web-based applications.

The remainder of the paper is organised as follows: in the next section, we briefly discuss relevant state-of-the-art in DIS systems, planning algorithms, dynamic coarticulation based systems, and existent systems that are capable of dynamic narration in DIS. In section three, we introduce DIEGESIS and Charisma, and the different components that compose these frameworks. In section four, we present the unification of DIEGESIS and Charisma into one framework, presenting the extra interfacing sub-components. In section five, we discuss the evaluation of the framework resultant from the unification of DIEGESIS and Charisma. Finally, in section 6 we present our conclusions and future work.

II. RELATED WORK

Recently there has been an increase in the development of high level behaviour planning for storytelling. Recent examples of DIS systems include Fabulator [4], Gadin [5], I-Storytelling [6], and Othello [7], each of them including its
own feature set. Reviews of related DIS systems and their features can be found in [1] and [8].

Planning algorithms are one of the most widely used techniques for DIS. A review of planning algorithms that are commonly used in DIS systems such as Graphplan [9], can be found in [1].

One of the differences between the above mentioned systems and DIEGESIS is that we focus on creating and implementing new planning and re-planning solutions specifically designed for DIS, where the existing systems use generic AI planning algorithms to achieve their goals.

Another difference is that in some of the systems [4, 6] there is a main character/agent and the whole story is generated around him/her. In contrast, our system is independent of a main agent, with all of the agents being able to contribute equally to the story based on the storyteller’s modelling and the user’s choices. However, an interesting additional feature would be to have a virtual representation of the storyteller to tell the story as it is generated dynamically. This can be achieved with a talking head applying the latest 3D facial animation, coarticulation and text-to-speech technologies.

A few systems have attempted to provide narrative qualities to a DIS system. Theune et al. [10] integrates speech capable of prosodic changes to create emotional speech in key parts of the story increasing its climax. Porteous et al. [11] utilise a virtual environment which is capable of describing the story from the character perspective. Cavazza et al. [12] proposes also a 3D virtual environment for dialogue narratives using speech. While each one of these systems enhances the narrative experience, they all lack in creating an immersive experience which provides realistic character animation. In [12] [11] dialogues between characters occurs but there is no lip animation between the characters. Theune et al. [10] does not use a virtual environment or virtual characters to narrate the story, it only uses synthetic generated speech.

During the past thirty years several methodologies have been created to synthesise facial animation, which can be divided in two main categories, physical muscle based and geometric approaches. Documentation regarding the processes and methods used to create each one of these approaches is well documented in [2, 13]. Within existing facial animation frameworks we can distinguish three; Greta [14], Xface [15] and Sanchez et al. [16].

Greta [14] proposes a muscular based approach for its facial model, composed by two layers to aid in the skin deformation, which uses Blinn’s Bump mapping approach to simulate wrinkles. Greta is divided in three main offline components, a core library that represents the animation framework, an editor and a player. Xface [15] uses a free-form approach to create skin animation using only one layer to animate the model, wrinkle simulation is done by using raised cosine function. It is composed by four main components, a core library with animation framework and core functions, the editor, a player and remote client applications. Sanchez et al. [16] integrates another geometrical approach which implements a modified version of free-form deformation, designated planar bones, this method also permits to model facial wrinkles with good performance but limited realism, when compared with Greta [14]. This animation method proved to be efficient and capable of animating in real-time using MOCAP captured movements with high number of polygons. The target of this framework is to capture and playback MPEG-4 animation, its framework is mainly divided in two layers, the core framework and UI composed by a player.

There are a vast number of methodologies [13], however only viseme driven approaches attempt to understand the different coarticulation effects, and the relationship between phonetics and visemes. Pelachaud [17] implements a look ahead model by assigning different ranks of deformability to each phoneme, applying to these forward and backward rules, so that a phoneme will take the lip shape of a less deformable phoneme occurring earlier or later. However this model assumes that only vowels affect speech segments models. Löfqvist [18] produced a coarticulation model that introduces the concept that speech segments have dominance over the articulators, which can either increase or decrease their target value (i.e. mouth shape) over time. The target value is modulated by a dominance function to model the implicit coarticulation. This approach has been further developed by Cohen and Massaro [3] who introduced several functions that allow to control several aspects of the dominance functions over the articulators, such as: the dominance that a segment has over others; time offset; and duration. The integration of these parameters affects the shape of the dominance functions and consequently their target value. Several extensions we created to solve issues existent in Cohen-Massaro approach, which attempts to restrict the overall shape for speech segments and introduce temporal resistance [3, 19, 20].

In this section we have reviewed relevant work in the DIS research area and work related with MPEG-4 FA frameworks and lip-sync animation. In the next section we will introduce succinctly both DIEGESIS and Charisma, giving relevant details for each component of the frameworks.
description of DIEGESIS components can be found in [21] and [8].

The WM is the component which coordinates the whole framework and is composed of several smaller components. It has direct access to the KB and is responsible of keeping track of and updating the current state of the world, the state of the resources, the environment that the agents are aware of, and to manage the time of the world.

The WM is also responsible for allocating resources to the planner (therefore to the agents) and retrieving each agent’s plans, solving any conflicts between the agents, so the story can be generated and progress.

Another responsibility of the WM is to send messages to the User Manager (UM) so they can be displayed to the user, as well as to process the user’s input received by the UM.

Each character in the story is represented by an agent and uses an instance of the planner to generate plans of actions and regenerate them, if needed, based on a set of goals which is initially specified by the storyteller, but can be modified by the framework during runtime.

The planner (figure 2) consists of a new planning and a new re-planning algorithm, able to generate plans of actions based on each agent’s state and context, considering both the current world state and the available resources. In the re-planning phase, the previously generated plan is kept and is used to identify the action(s) needed to be altered to generate a new plan with a minimal disruption to the original plan.

![Fig. 2 – Planner architecture](image)

The planning algorithm is based on Graphplan for solutions expansion, and backtracking heuristic search enriched with constraints satisfaction and dynamic opportunistic restart when required. By the term planner we refer to the component that deals both with the planning and the re-planning processes.

The planner requires a domain definition (provided by the WM), which lists the facts about the current state of the world, the applicable actions or operators, requirements, and constraints. Actions are usually consisting of three parts: parameters, preconditions and effects. Each action has also a specific duration and a salience value, which are specified by the storyteller.

Each agent has a list of goals and is aware of how much time it has available to complete them. Using the planner, an agent tries to find a valid plan which achieves the goals within the specified time constraints.

The first step of the planner is to generate the layers of the planning graph until all the goals are present in a layer, considering the constraints (mutexes) between them. Two actions in the same level are considered mutex when their preconditions and effects are inconsistent.

After the planning graph is created, a backward search is being performed starting from the last generated state (containing all the goals) until a valid plan is discovered. While searching for and extracting the plan, the total duration of the plan is calculated based on the duration value of each action. If at some point the plan’s duration exceeds the available time that an agent has, the search/extraction stops.

Finally, the planner chooses the least important goal (based on the salience value of the goals), removes it, and starts the search for a valid plan and its extraction all over. This happens until a valid plan within the time constraints is been discovered.

The story is executed in a turn-based way, based on the agents’ plans. For each action an agent tries to execute, the WM queries the Resource Pool and checks the current state of the world to determine if it’s applicable. If it is, then the Resource Pool and the current state of the world (and therefore the environment that the agents are aware of) are updated, to apply the changes made by the execution of the action. If it’s not, depending on the conditions, the agent is either instructed to wait for the next turn and try again, or to re-plan.

B. Charisma Framework

Charisma proposes an MPEG-4 FA compliant framework, built upon three main design choices, modularity, performance and realism, where we attempt to reduce the number of compromises as much as possible while permitting to expand the framework into numerous situations. As it is possible to observe in figure 3, charisma is composed by three main layers of offline components, at the bottom layer the core library contains all classes and interfaces that communicate with the rendering engine, and permit the playback and recording of MPEG-4 FAP animation.

![Fig. 3 – Charisma framework architecture layers](image)

The middle layer is composed by the embodied layer which permits the inclusion of embodied modules, such as coarticulation, expressions, moods, personality, etc. (only coarticulation is currently implemented), and permits the communication between modules, the core library, the scene manager, the assets and the user, to create autonomous dynamic behaviour. The third and final layer contains all the UI applications: the editor, which permits to playback and record FAP animations, and dynamic generated animation,
and the player which permits the playback of FAP animation and to generate dynamic speech animation.

1) *Animation Manager*

Animation in Charisma utilizes a layered approach to animate MPEG-4 FAP animation and embodied dynamic generated animation, composed by two main sets of layers $S_{FPg}$ and $S_{Mod}$: $S_{FPg}$ set is composed of five layers for each feature point group (eyes, mouth, nose, etc.) offering compliance with MPEG-4 animation to play FAP streams.

$S_{Mod}$ manages animation from embodied modules, such as coarticulation and other modules, which permits dynamic animation generation. Animation in layers belonging to $S_{Mod}$ uses dynamic generated key-frame approach and integrated to each layer.

$$S_{FPg} = \{FP_{g1}, FP_{g2}, …, FP_{gN}\} \quad (1)$$

$$S_{Mod} = \{l_{vis}, l_{exp}, l_{gaze}, …, l_{n}\} \quad (2)$$

$$L = \{S_{Mod} \cup S_{FPg}\} \quad (3)$$

$$B = \text{LERP}(L) \quad (4)$$

This layer system permits the creation of dynamic FAP stream animation and blends it with dynamically generated animation from $S_{Mod}$ layers, the blending between $S_{Mod}$ and $S_{FPg}$ is adjusted manually by the user/animator using LERP/SLERP functions (4). Blend weights have a default value of 0.5 but can be set by the animator for fine tuning. The final results can be recorded to FAP stream animation.

**IV. THE COMBINED FRAMEWORK**

DIEGESIS is capable of generating interactive, dynamic, emergent, and consistent narratives, however the language of the outcome story presented to the user is not rich and expressive. Therefore, it seems as a good fit to use an intermediate system that generates natural language, such as those introduced in [10, 22], and presents it to the user by text and speech, using our coarticulation system in Charisma, providing a more natural interface to the user.

Using the techniques discussed in section III (A), the proposed framework generates a narrative (figures 4 and 5), based on the initial story model provided by the storyteller, the interaction between the agents, and the choices of the user.

Once a part of the narrative is generated, it may be converted to natural language, and handed to the coarticulation embodied module. This model loads a viseme set at the application launch that matches the language used in the natural language generator that represent all possible phonemes that the Mary Text-To-Speech (TTS) engine is capable of generating.

The coarticulation module sends two requests to the text-to-speech generator, one for audio containing the text for the speech, and following one for acoustic parameters which will contain the duration, times and prosodic effects for each phoneme. The phonetic information is matched and converted to their matching visemes. Coarticulation is achieved by using a modified version of Cohen-Massaro [3] coarticulation model, which is composed by $T = (t, \alpha_{sp}, \theta, \tau, c, T_{sp})$.

These parameters are integrated within two functions $D_{sp}(t)$ (1) and a weighted average function $F_{sp}(t)$ (3), which gives the final lip opening value for the visemes in the resultant utterance, from the natural language generator module.

In (1), $\alpha_{sp}$ represents the peak magnitude of the segment pair during time $t$. $\theta$ represents the rate of magnitude up to the segment centre and the rate of falloff after the peak. The function $c$ is usually left with a constant value of 1.

In (2), function $\tau$ calculates the temporal distance between the segment centre and its $t_{sp}$ time offset, at time $t$.

The average weighted function $F_{sp}(t)$ (3) was introduced by Cohen et al. [3] which calculates the final target value by including all segments at time $t$.

$$D_{sp}(t) = \begin{cases} \alpha_{sp} e^{-\theta \cdot |\tau|}, & \text{if } \tau \geq 0 \\ \alpha_{sp} e^{-\theta \cdot |\tau|} e^{-c}, & \text{if } \tau < 0 \end{cases} \quad (1)$$

$$\tau = t_{start} + \frac{\text{duration}}{2} + t_{sp} - t \quad (2)$$

$$F_{sp}(t) = \frac{\sum_{i=1}^{N} (D_{sp}(t) \times T_{sp})}{\sum_{i=1}^{N} D_{sp}(t)} \quad (3)$$

These are further processed to transforms and synced with audio and text as can be seen in figure 5, and transferred to the 3D scene where the talking head will narrate the content from the script.

The current framework provides a richer user experience, by using a 3D narrator to tell the generated story, and by generating a richer and expressive language for the narrative, when compared with the previous outcome of the framework.

In the next chapter we will evaluate the resultant framework and the several components that compose it.
V. EVALUATION & DISCUSSION

As discussed in [8], the scenario we are using for our DIS evaluation, is based on Homer’s epic poem, Iliad [23]. Figure 6 illustrates an example of an agent’s combined plan, taken from a part of the scenario. The agent represents the character Achilles in the story.

Using a dialogue generation solution, we will generate an expressive narration based on the outcome which DIEGESIS is generating, so it can be imported to Charisma.

An example of the narrative that could be generated by the dialog generation solution based on the outcome illustrated in figure 6, is the following: “Achilles picked up his sword and shield, and wore his armour. Afterwards, he left his tent, heading for the battlefield. There, he attacked Hector and killed him. Achilles picked up Hector’s body, returned to the Greek camp and dropped Hector’s body. He picked up Patroclus’ body and walked to the beach. There, Achilles prepared for Patroclus’ funeral, and then performed it. He returned to his tent, where Priam was waiting and, after a discussion, Achilles allowed Priam to take Hector’s body.”

The stories generated by DIEGESIS are more complex than shown in this example, as it supports multi-agents planning, re-planning and action execution taking into account constraints and complex interaction between the agents and the user. A detailed discussion and scalability evaluation of DIEGESIS can be found in [21] [8].

Once the natural language based story is generated using the tools present in the DIEGESIS framework and the dialogue generation solution (fig. 7), it is then sent to the TTS engine (MaryTTS) and pre-processed to generate lip-sync animation using our extension to Cohen-Massaro [3] solution.

Table 1: Transforms number for output generated from DIEGESIS on different fps

<table>
<thead>
<tr>
<th>FPS</th>
<th>Number of Transforms</th>
<th>Lip-sync generation time (ms)</th>
<th>Coarticulation total time (ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>25</td>
<td>11664</td>
<td>1506</td>
<td>3888</td>
</tr>
<tr>
<td>60</td>
<td>29168</td>
<td>1527</td>
<td>3800</td>
</tr>
<tr>
<td>90</td>
<td>42416</td>
<td>1536</td>
<td>3912</td>
</tr>
</tbody>
</table>

The framework generates the lip opening values for our weighted average function \( F(t) \) (3) (as can be seen in figure 9) over the total time given by the TTS engine for the voice. The number of transforms generated by the framework is dependent on the animation frames per second (fps). The time taken to generate coarticulation for three different frame rates, 25fps, 60fps and 90fps, were generated with an i7-3820 with 16GB ram and a GTX560 GPU card. The results can be seen in table 1, where it is possible to observe that coarticulation generation time decreases slightly with the number of fps, but the total time taken to create lip movements and audio does not support this since our TTS engines took different times to generate speech. However, the number of transforms required to generate the lip-sync movements is reduced by about 73% between 25 and 90 fps which permits great efficiency without reducing the animation quality significantly.

In figure 8 we evaluated visually our framework against the resultant coarticulation from another accessible MPEG-4 FA framework Xface, for the excerpt of the output “Achilles picked up his”. While this type of evaluation is subjective we cannot deny the importance of the visual animation quality in the coarticulation output. During our analysis in the screenshots and the audio from Xface coarticulation system it was possible to observe that Xface produces lip movements for each viseme, which shows that it does not consider coarticulation effects, or dominance that certain visemes have over others. During our analysis it was also possible to observe that our solution is capable of good realistic speech animation by using a Cohen-Massaro coarticulation solution, our lip smoothing solution is capable of creating a smooth animation in most circumstances, however it two visemes \( v_0 \) and \( v_1 \) have very different shapes it was possible to see a slight jump during the transition time.
The evaluation of a system of this magnitude proves to be very complex, with no established methodology to create a complete objective evaluation to compare our system with other narration systems, such as [11, 12], which led us to evaluate each major component against available state-of-the-art solutions.

VI. CONCLUSION

In this paper, we presented a conversational avatar framework for DIS, which resulted from the combination of DIEGESIS, a storytelling framework, and Charisma, an MPEG-4 FA framework, with a dialogue generation solution as an intermediate. The framework is capable of generating dynamic stories and dictating those to the user, using a realistic lip-sync animation with a virtual 3D character. The current language generator module is only in its design stage and its output is limited to certain plans, and the animation from the coarticulation solution requires a better smoothing algorithm between visemes. However, it is still able to produce realistic results.

In the next stage of our work we will focus in expanding our combined framework focusing on the natural language generation module, which will enable us to generate dynamically different stories, by increasing its natural language knowledge base, and integrating better syntax and semantic support. The animation of the framework also requires a better smoothing algorithm and an evaluation tool that permits comparison against real speech to make an objective and more conclusive evaluation.

VII. REFERENCES


