A Notation Based Approach To Film Pre-vis

**Ajay Chakravarthy*, Richard Beales*, Yvonne Jung†, Sebastian Wagner† , Christoph Jung† , Angelos Yannopoulousᵒ ,
Stefanos Koutsoutosᵒ , Rolf Schiffmann˘ , Rolf Hedtke˘ , Ignace Saenen˒**

*IT Innovation Centre, University of Southampton. {ajc,rmb}@it-innovation.soton.ac.uk.
†Fraunhofer IGD/ TU Darmstadt. {yvonne.jung,sebastian.wagner,christoph.jung}@igd.fraunhofer.de
ᵒNational Technical University of Athens, Greece. {ang, stefk}@mail.ntua.gr
˘Precision Mechatronics GmbH, Germany. {rolf.schiffmann@precision-mechatronics.com, rolf.hedtke@hs-rm.de}
˒ Larian Game Studios, Belgium. {ignace.saenen@larian.com}

Abstract— This paper describes ANSWER which is a new approach to the creative process of film production. It is based on a symbolic notation system called Director Notation (DN) through which film directors are free to express their creative intent much like a music composer uses musical notation. The notation populates a semantic model which is machine processable and this input is automatically converted into a lightweight 3D pre-vis. ANSWER is integrated with DFT BONES Dailies, so that in post-production editors can quickly synchronise and compare pre-vised and actual footage. In this paper we will describe the various software tools which have been developed as part of the ANSWER project to achieve this goal.

Keywords- Director Notation; Pre-Vis; Real-time Rendering; Animation; Ontology; Story Boarding; Film Production.

I. INTRODUCTION

The process of film production is lengthy, time consuming and usually expensive. Modern production techniques such as computer-generated visual effects, mixed reality production and stereoscopic 3D have further complicated an art that has always involved a great deal of planning, preparation and coordination [5]. Paper-based story boards and artist created pre-vis are currently used by directors to record their vision during film composition. ANSWER [7] introduces Director Notation [2, 3], a symbolic notation system using which film directors can express their creative intent and author film scenes. The symbol set used in DN is semantically mapped to a machine processable model or ontology, which is used to generate automatic lightweight 3D pre-vis animations. In this paper we will describe the ANSWER toolkit which is a set of software components developed during the course of the project to support the use of DN for film production.

Current tools often used for pre-vis of movies such as Maya require the scene author to work at a very low and technical level. For example, the latest version, Autodesk Maya 2011 [9], promises to accelerate pre-visualization and virtual movie-making production by enabling artists to layout and manage multiple camera shots in a single animation sequence. The timings of shots can be changed and alternate versions can be created and reviewed. But this requires having skilled modellers and animators at hand. There are also some commercial tools available for creating pre-vis films like FrameForge [10], which is a level-editor-like authoring tool that comes with a comprehensive set of assets and effects. But as it does not provide any high-level interface, it is more comparable to a typical animation engine.

The rest of the paper is organised as follows. Section 2 introduces Director Notation which is the language used by directors to express their creative intent. In Section 3 we discuss the architecture for the ANSWER platform and in the subsequent sections we take each modular component of the architecture and discuss its functionality. Section 4 discusses the Notation Editor graphical user interface, which will be the point of interaction for users with the ANSWER platform. In Sections 6, 7 and 8 we detail the various tools which enable the transformation of user input into automated pre-vis.

II. DIRECTOR NOTATION

Director Notation developed by ICCS, NTUA is a system that simultaneously allows an artistic and a technical formalisation of the domain of film directing. Its main purpose is to analyse and represent media content during its production phase, rather than capturing knowledge about already existing media [4]. The artistic notation is used by the film director to formally record his ideas regarding the creation of a film (as opposed to using natural language or some kind of technical formal language similar to an animation script) – but it is also fully computer-processable. This leads to various applications – such as automatic synthesis of animated storyboards (a focus of ANSWER) or the automatic cost estimation of a production from the earliest phases of preproduction (a topic of future research). All applications are based on the fact that Director Notation is transformed into metadata that not only accurately describes film content but is also maintained and enriched within the whole module pipeline up to post-processing: we retain a machine processable annotation of what the director was recording in order to produce the film. The description in notation is *formal* – the language's semantics are strictly defined – so the automatically derived technical annotation corresponds exactly to the user's input.
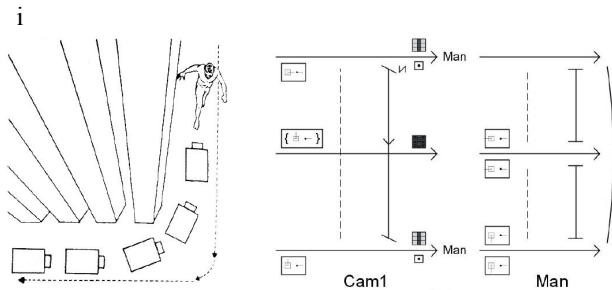
Figure 1: Left: A sketch of a short shot taken from a director's storyboard. Right: The same scene in the formal Director Notation with framing symbols [2].

DN currently consists of two parts: Acting and Camerawork Notation. The spatial layout for the Acting Notation follows the same basic principles, which govern the Camerawork Notation layout as was described in [2]. Comparable to Labanotation [15] both are organized in several parallel columns called "staff". The time factor is represented as a vertical timeline which starts at the bottom and expands upwards. All columns are correlated according to that timeline so that, if a horizontal line is drawn across the page, every symbol that the line passes through is in effect at that specific time instant. Each individual element is represented by a symbol that is temporally delimited by a pair of so-called relation symbols defining the time interval in which the element is in effect. Figure 1 shows a Camerawork Notation staff example compared to a storyboard sketch.

III. ANSWER ARCHITECTURE

The ANSWER architecture consists of a number of software modules (Figure 2). The Notation Editor is a GUI (Graphical User Interface) that enables the end-user to edit notation, producing Director Notation scores. It also displays the 3D animation that corresponds to the score, while DN scores are being edited. The Rule Engine is a web service provider that performs semantic analysis of DN and converts a DN score into a fully resolved form in order to provide input to other ANSWER software modules. The Instant Reality framework [8] transforms camera and actor instructions, as delivered by the Rule Engine, into a rendered 3D pre-vis animation. The major contribution here is in the framework developed by Fraunhofer IGD that simplifies the integration of virtual characters into an interactive real-time visualization system by introducing another level of abstraction (namely the interface and control language PML that takes concepts from multimodal markup languages used for simulating embodied conversational agents such as BML [14], though on a more generic level) on top of X3D [6].

Today's modelling and visualization systems are often highly specialized and rather sophisticated, but they utilize proprietary formats and methods that incompatible in their concepts of operation and in the supported data formats. On the one hand this prevents harmonization of 3D assets from different companies and thereby hinders distribution and utilization of 3D contents. On the other hand this leads to parallel developments of incompatible and isolated technologies. In order to alleviate this and to provide a basis for a sustainable solution, for visualization we therefore propose a framework that utilizes the scene-graph concept and builds upon the open ISO standard X3D. We are also investigating applications of ANSWER to the computer games industry; for this purpose the Dialogue Editor module transforms data provided by the Rule Engine so that they can be processed by a Game Engine.

Each time the notation author performs a change in the score, the Notation Editor makes a request for semantic analysis to the Rule Engine. Also through the Editor the notation author provides the Rule Engine with information regarding preview camera selection. Once the semantic analysis is completed, the Rule Engine issues instructions to Instant Reality that in turn generates a new animated visualization of the shot.
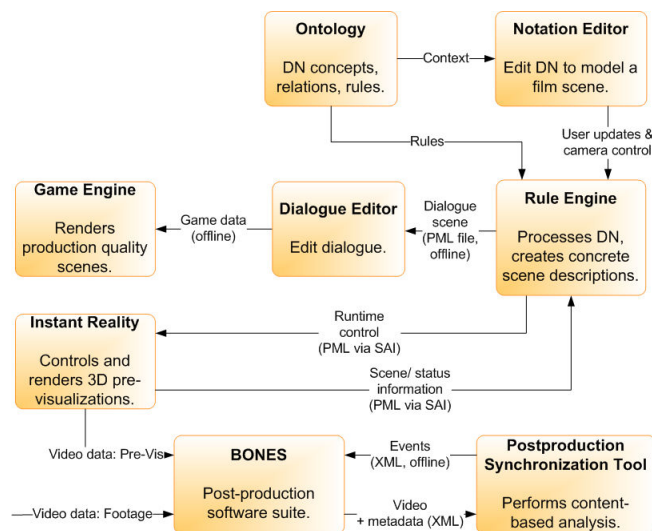

Figure 2: ANSWER Architecture

IV. THE NOTATION EDITOR

The Notation Editor developed by the University of Southampton, IT Innovation Centre provides the GUI (Figure 3) through which the director is able to notate the shots that he or she wishes to capture for a particular project. It forms the first point of interaction a director will have with the ANSWER technology and methodology. The director starts with a pre-authored film script annotated using MSML (Movie Script Mark-up Language) which has been developed as part of the ANSWER project [1]. MSML enables the tagging of script elements with the use of a higher level schema used to represent the standardized terms used for a film script in the film domain.

The director is able first to review his working script, and then outline the shots he wishes to capture. Each shot is described as a span of one or more dialogue or action elements in the script. The director then proceeds to describe in detail the camera and actor actions they want to take place in each of those shots using the inbuilt Director Notation authoring tools. Within the GUI the director is also able to

retrieve rendered visualisations illustrating each shot they have notated, allowing them to iteratively refine the notation to accurately reflect his artistic intent
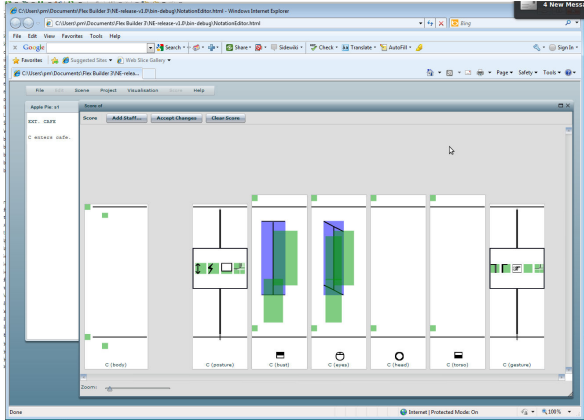

Figure 3: Notation Editor GUI

The Notation Editor GUI is configured dynamically based on the information encoded within the semantic layout model or ontology. This ontology acts as a configuration for determining the placement, size and functionality of the GUI components for the Notation Editor. This way users are able to customize the editor to suit their need by modifying the configuration in the ontology and do not have to deal with low level programming code. All interaction of the user with the Notation Editor is communicated via web services to the Rule Engine.

V. RULE ENGINE

The Rule Engine is developed by ICCS, NTUA and is responsible for semantic checking on the user input in order to validate the DN scores created in the Notation Editor GUI. Any valid input is transformed to an internal representation (numeric, as opposed to declarative), containing all information the user has notated with the symbols he or she inserted. This transformation happens regardless of whether the user has notated a full scene or not. The Rule Engine is integrated with the 3D engine (Instant Reality). It communicates with Instant Reality by means of the assets which are the 3D objects, which will be used during the rendering process. The 3D scene description generation is for a specific collection of those assets (the user chooses which ones). Here the – on linguistic grounds – elliptical user input, which uses relative timing and has behaviour classes like "head nod", is transformed into its fully resolved form. Therefore, the user notation input is enriched with scheduling information and translated into a more concrete representation by mapping the higher-level notation concepts to a less abstract description of scene and animation data (e.g. load and play animation file "waveHand.x3d").

This is achieved with the help of a so-called Gesticon [11]. Therefore, an XML-based dictionary containing all assets including their semantic meaning is generated in advance (i.e. it has to be declared, which file or 3D object contains what abstract type of object, gesture, etc.) to enable the concrete mapping to certain PML commands, as visualized in the upper right part of Figure 4. Since all higher level descriptions are given in XML they can easily be transformed and enriched to more concrete scene descriptions with standardized methods. In order to achieve smooth interoperability and correct animations, the asset sizes and miscellaneous properties of them are taken into account when transforming to an external format such as PML (Player Markup Language).[1] The Rule Engine software module generates messages during the distinct actions it takes to check, evaluate and transform users input. These messages are errors (syntax & semantics related) and warnings (semantics related).

The Rule Engine is internally split in two halves. The core system is a language agnostic system taking care of all checking, evaluation, transformations, communications, messaging and other minor things. The other part is language dependent (Director Notation dependent in this case). It consists of the configuration of the language itself (driving the checking), special code for the interpretation of specific parts of the language and finally a collection of rules which are applied to any symbol collection by the core system to achieve the evaluation (the result of which is the internal format). The output of the Rule Engine processes is a PML file which acts as input for the pre-vis generation engine.

VI. INSTANT REALITY – PRE-VIS

The Instant Reality framework [8] developed by Fraunhofer Institut für Graphische Datenverarbeitung (IGD[2]) is a visualization/ Mixed Reality system that within the course of this project is used to visualize the Director Notation content. The framework provides a comprehensive set of features to support classic Virtual Reality and advanced Augmented Reality equally well and combines various components to provide a single and consistent interface for application developers by utilizing X3D, which is easily distributable and sharable to others, as application description language. The visualization engine's responsibility is to process scene description and animation requests in PML format from the Rule Engine through a network interface, which in turn triggers an interactive visualization of the director's intent. Therefore, a flexible rendering and animation system is needed for fulfilling those generated requests.

---

[1] Link to the full PML documentation: http://www.answer-project.org/innovations/PMLSpecification.pdf
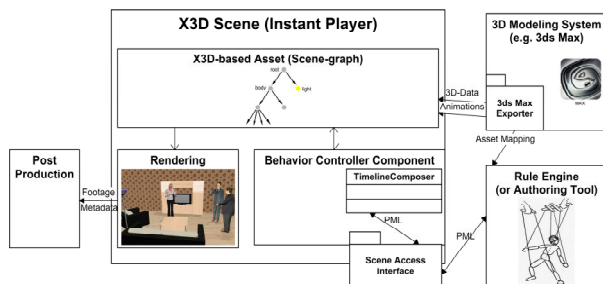[2] http://www.igd.fraunhofer.de/

Figure 4: Internal view of the visualization engine showing the interfaces and system architecture.

When all content is created and setup, the Rule Engine can send PML commands via the network layer/ SAI (Scene access interface) to the TimelineComposer scene-graph node (cp. lower part of Figure 4), which is responsible for all scheduling and also deals as the PML interface and processor. Starting and stopping of animations and other events is thereby accomplished by setting the node's 'command' field. This process is explained in more detail in [12, 13]. Finally, when the scene gets rendered, frames can be dumped for further evaluation and post-processing by the PST (see left part of Figure 6). The rendered pre-vis is converted by the Rule Engine into an FLV video and streamed back to the Notation Editor. This way the user is able to author a scene description using the Editor, invoke the Rule Engine and immediately view the automatically generated pre-vis from within the same working environment.

VII. POST PRODUCTION SYNCHRONIZATION AND BONES

The ANSWER Postproduction Synchronization Tool (PST) (Figure 5) developed by Fraunhofer Institut für Graphische Datenverarbeitung and Precision Mechatronics GmbH. supports the users during the Postproduction process in synchronizing footage and corresponding pre-vis. The ANSWER PST is designed as a self-contained stand-alone application, which is connected to the BONES Dailies [3] Postproduction System via a simple network interface. Data exchange is maintained by TCP/IP based commands as well as Job- and result descriptor files (XML).

Once a scene has been shot in several takes on set, it usually deviates from the intended DirectorNotation script. Unlike pre-vis, the footage scenes start with a film slate sequence. Further timing of events (like an actor entering the frame, camera action, etc.) may be different, or additional/missing events may occur. The PST has to compensate for these deviations, to allow the editor to synchronize a pre-vis (and correspondingly its DN representation) scene with a selected footage take. Differences between pre-vis and footage are evaluated with respect to timing of visual events in the video sequences. By extracting these events, correspondences can be found and

deviations of event pairs can be compensated for synchronization.

To implement the event extraction, the PST performs automatic content based analysis of real video footage as well as corresponding 3D-pre-vis clips, and returns extracted events such as appearances of human face and film slate, characteristic camera motion, video shot boundaries as well as generic image motion to BONES. An additional synchronization event is generated by the PST for both footage and pre-vis, which allows for simple synchronization of clips by single event reference. If required, for example the first appearance of a face in the video data can be used as event to synchronize to. Additional manual event editing in BONES allows the user to adjust the found events from the PST before using it in the further process steps.

After the events are extracted and transferred to BONES, matching can be performed. The PST tries to find pairs of corresponding events in pre-vis and footage. In this nearest-neighbour search it is assumed that footage is mostly coherent with DirectorNotation. For later synchronization at least one matched event per clip is necessary.
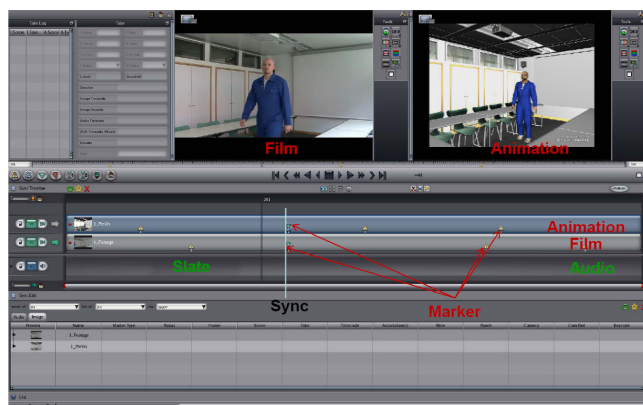

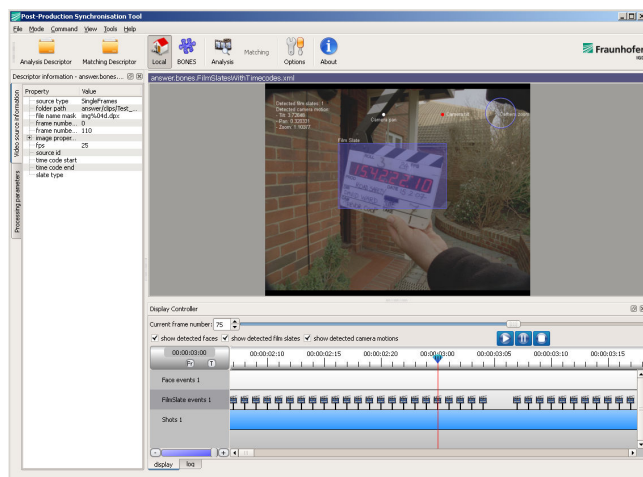Figure 5: Synchronization of Pre-vis and footage in BONES.

Figure 6: The ANSWER PST graphical user interface, showing automatically detected film slate and extracted time code digits.

---

[3] http://www.dft-film.com/software/bones_dailies.php

The matched events are returned to the NotationEditor (see Section 4), where the timing deviations can be compensated in DirectorNotation and a new synchronous version of pre-vis can be rendered.

After the previous steps have been performed for all selected footage takes of the project, the PST can generate a rough-cut of the footage scenes, which means cutting and arranging all footage subsequently, such that a continuous preview of the final clip/movie can be watched. It is now assumed that (1) all pre-vis clips are synchronous (with respect to recognized events) to corresponding footage and (2) that the length of pre-vis indicates the true length of footage. Then the relevant part of the footage take can be simply "punched out" using pre-vis start/end frame numbers. The final rough cut can then again be visualized in the BONES GUI.

VIII. DIALOGUE DESIGNER AND GAME ENGINE

The ANSWER Dialog Designer component developed at Larian Game Studios is an off-line conversion tool that allows the conversion of ANSWER generated cinematic content into a proprietary format suitable for visualization by a game engine. (see Figure 7).

The content creation tool features dozens of content generation workflows and use-cases that eventually lead up to the production of a fully localizable cinematic sequence that can be interactive or non-interactive; camera directed, or simply without camera; all within the concept of a dialog-oriented story script between multiple actors. The sequencer supports a real-time multiple camera system featuring a myriad of lens options that allow the artist to capture the action. It also allows the artist to re-iterate over said content and fine-tune details such as special effects, animation loops, audio and lip-synchronization.
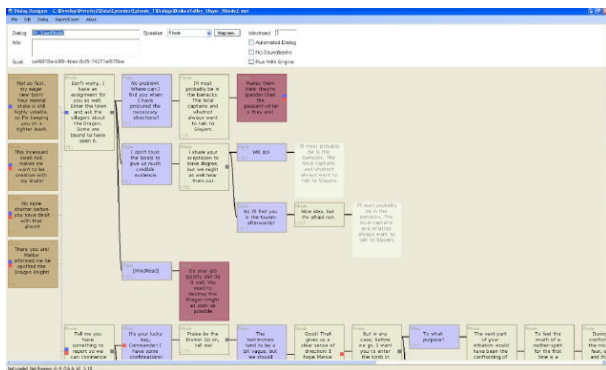

Figure 7: Dialogue Designer.

The ANSWER project uses an extension of this Dialog Designer tool to convert cinematic content, generated as PML data by the Rule Engine from Director Notation, into a game-engine-specific format. As a requirement, the converted data can subsequently be visualized by a game engine. In the scope of the ANSWER project, the game engine developed by Larian studios will serve as the perfect guinea pig, and effectively proves that ANSWER can bring

added value to similar interactive and non-interactive media projects as a content creation tool or for machinima films. For this purpose, Larian's Divinity 2 game engine has been intensively modified such that the camera system is able to support a growing subset of features from the PML specification. The actor system supports multiple animations and animation blending over time. The scheduler is able to work with multiple actors and actor scripts. At the same time, the dialog designer is used as a base-layer in the conversion tool, now capable of producing engine specific content. The conversion front-end features a full PML specification compliant XML-based parser, allowing the back-end converter to take full advantage of the PML data generated by the Rule Engine.

IX. CONCLUSIONS

We have introduced ANSWER, a new approach to film planning and pre-vis generation. A symbolic notation, Director Notation, has been developed that can be used to formally record the director's intent for the structure and content of individual scenes. Supporting the Notation, we have created a modular software platform that supports authoring of Director Notation scores through an interactive GUI and immediate review of the resulting pre-vis animation. For pre-vis generation we utilize X3D, because it is an open and declarative standard that allows defining scene description and runtime behaviour of 3D content descriptively. However, X3D currently does not provide mechanisms for behaviour control. Therefore, we developed several techniques for describing cinematic content including camera and character control. This is achieved by wrapping the functionality with the PML layer, another level of abstraction that builds on top of X3D and was first proposed in [12]. We expect solutions based on this approach to result in modular building blocks, which moreover can be re-used across a variety of application domains.

Metadata is captured implicitly via the Notation itself and explicitly through additional semantic annotations made in the GUI, which support the Notation. A post-production synchronisation tool has been developed that integrates with the BONES Dailies suite to exploit this metadata, allowing actual footage to be synchronised and compared with the original pre-vis. Finally, applications of ANSWER technology within the games sector are being explored using a Dialogue Designer tool integrated with Larian Studios[4] Divinity 2 game engine. The presented framework and techniques are thereby also useful for game design and creating machinima films etc.

---

[4] http://www.larian.com/

REFERENCES

[1] A.Chakravarthy, R.Beales, P.Walland, A. Yannopoulos, ANSWER: A Semantic Approach to Film Direction. *In: Internet and Web Applications and Services, International Conference on, pp. 645-648, 2009 Fourth International Conference on Internet and Web Applications and Services, 2009.*

[2] A.Yannopoulos, K.Savrami, T.Varvarigou. DirectorNotation as a Tool for AmI & Intelligent Content: an Introduction by Example. *Accepted for publication in the International Journal of Cognitive Informatics and Natural Intelligence (IJCiNi).*

[3] A.Yannopoulos, K.Savrami, T.Varvarigou. Artistic Notation Systems Integrated in Software Engineering of Knowledge Technologies: emphasising film, animation and games. *Accepted with revisions, ACM JOCCH (Journal of Computers and Cultural Heritage).*

[4] A.Yannopoulos, Y.Christodoulou, S.Koutsoutos, K. Savrami, T.Varvarigou, V.Alexandrou, ANSWER: Documentation, Formal Conceptualisation and Annotation of New Media. *In: Adjunct Proceedings of EuroITV2009, Leuven, Belgium, June 3-5, 2009.*

[5] R.Beales, A.Chakravarthy, R.Hedtke, J.Christoph, Y.Jung, S.Koutsoutos, and A.Yannopoulos. Automated 3D Pre-Vis for modern Production. *In: International Broadcasting Convention (IBC) 2009, 10-15 September, 2009, Amsterdam.*

[6] Web3D Consortium. Extensible 3D (X3D), 2008. *http http://www.web3d.org/x3d/specifications/ISO-IEC-19775-1.2-X3D-AbstractSpecification/index.html.*

[7] ANSWER, 2010. Artistic-Notation-based Software Engineering for Film, Animation and Computer Games. http://www.answer-project.org/.

[8] Fraunhofer IGD, 2010. Instant Reality. http://www.instantreality.org/.

[9] Autodesk, 2010. Autodesk maya 2011. http://area.autodesk.com/maya2011/features.

[10] Innoventive, 2010. Frameforge. http://www.frameforge3d.com/.

[11] B. Krenn., H.Pirker. 2004. Defining the Gesticon: Language and gesture coordination for interacting embodied agents. In Proc. AISB'04 Symp. on Language, Speech and Gesture for Expressive Characters, 107–115.

[12] Y.Jung., J.Behr. 2008. Extending H-Anim and X3D for advanced animation control. In Proceedings Web3D 2008, ACM, New York, USA, S. Spencer, Ed., 57–65.

[13] Y.Jung., S.Wagner., C.Jung, J.Behr., D.Fellner.: Storyboarding and Pre-Visualization with X3D. In Proceedings Web3D 2010, ACM, New York, USA, S. Spencer, Ed., 73–81.cole.

[14] H.Vilhjálmsson., N.Cantelmo., J.Cassell., N.Chafai., M.Kipp, S.Kopp., M.Mancini., S.Marsella., A.N.Marshall., C.Pelachaud., Z.Ruttkay., K.R.Thórisson., H.Welbergen., and R.J.Werf. The behavior markup language: Recent developments and challenges. In IVA '07: Proceedings of the 7th int. conference on Intelligent Virtual Agents (Heidelberg, 2007), Springer-Verlag, pp. 99–111.

[15] C.Griesbeck. (1996): Introduction to Labanotation. http://user.uni-frankfurt.de/~griesbec/LABANE.HTML.