ShapeShifting Screen Media: A Declarative Computational Model for Interactive Reconfigurable Moving Image Narratives

Marian F. Ursu[1], Jonathan J. Cook[1], Vilmos Zsombori[1], Ian Kegel[2], Doug Williams[2], Michael Hausenblas[3], Mika Lumi Tuomola[4]

[1] Narrative and Interactive Media Group, Department of Computing, Goldsmiths, University of London, SE14 6NW, UK
[2] Future Content Group, BT, Adastral Park, Ipswich, IP5 3RE, UK
[3] Institute of Information Systems, Joanneum Research, Graz, Austria
[4] Crucible Studio, Media Lab, University of Art and Design, Helsinki, Finland
m.ursu@gold.ac.uk

Abstract

This paper presents an approach and a computational model, implemented in a software system, for the creation and delivery of interactive moving image productions made with filmed material that level the aesthetic and editing quality of existing linear film and television programmes. They have been validated through a number of productions, notably through a dramatic production delivered on a national television channel in 12 episodes. The focus of the paper is on the declarative formal language, called *Narrative Structure Language* (NSL), which we developed for the representation of such narratives, and on an accompanying reasoning engine.

Introduction

Interactive moving image narratives have been approached from a variety of perspectives. In the commercial world, there are highly interactive games that are fairly simplistic in terms of the narratives they convey, and interactive television (iTV) in which engagers can choose between parallel or time-shifted linear programmes. Narrativity research into the language of the interactive screen media has produced theoretical accounts and prototypes, but they have not yet become commercial productions or established genres. Research driven by computational models for narrativity with multimedia objects has led to the development of various models and systems, depending on, for example, the purpose of the model, the computing paradigm used for modelling, the understanding of the term narrative, and the type of media objects employed in narrations. Reviews presented by Mateas and Sengers (1999), Cavazza, Charles and Mead (2002), Bulterman and Hardman (2005), and Riedl and Young (2006) define and discuss the most prevalent approaches. Despite a significant number of techniques and prototypes developed, this research has until now not been demonstrated in the realisation of a public or commercially validated interactive moving image narrative.

We have positioned our work in the context of this research in (Ursu et al. 2007a). Essentially, our approach is to empower the human-centred authoring of interactive narratives rather than attempting to build systems that themselves generate narratives. We have developed an approach to, a computational model, and an accompanying software system for the creation and delivery of interactive moving image productions, with a focus on filmed productions that aim for a level of aesthetic and editing quality comparable with existing linear film and television programmes. This represents the outcome of research collaboration between researchers in AI and digital media, creators of moving image productions and software developers.

The system we developed has been successfully used in the authoring of a number of interactive productions (including drama and documentary). Outstanding is the participatory black-comedy about love, 'Accidental Lovers' (Tuomola et al. 2006) which was fully authored and delivered with our system and in which both the plot and the discourse respond to the viewers' input provided in the form of SMS messages. Accidental Lovers was broadcast twelve times on Channel 1 of the Finish Broadcasting Company. The reported viewing figures were approximately the same as for other popular programmes broadcast in that time slot (10pm). They rose during the course of the twelve broadcasts.

This paper focuses on the declarative formal language we developed for the representation of such narratives, called the Narrative Structure Language (NSL) and, on two accompanying processes, namely reasoning and authoring. The reasoning procedures make use of AI techniques including logic programming, ontologies, symbolic representation, normative statements for rules, constraint satisfaction and heuristics for efficient computations. The paper gives an overall description rather than a detailed account of a specific technique.

ShapeShifting Screen Media

Scope. ShapeShifting is an approach applicable to any kind of digital media. However, in this paper it is regarded only from the point of view of screen media, such as movies and TV programmes. For brevity, in the remainder of

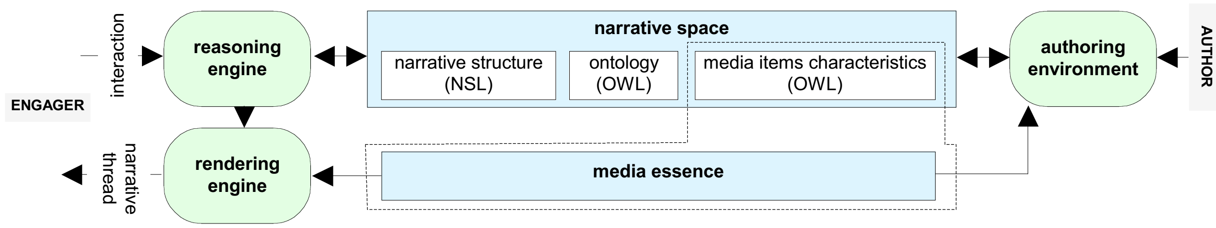

Figure 1. *The basic architecture of systems for (authoring and delivering) ShapeShifting programs*

the paper, we use the term *programme* to mean *moving image narrative*.

Definition. A *ShapeShifting programme* is an interactive programme that adapts itself – shifts its shape – to reflect the choices and preferences of the viewer/*engager*. ShapeShifting programs are ultimately composed of *media items*, such as video and audio clips, and graphics. Media items are the basic building blocks of ShapeShifting programmes. ShapeShifting programs are *authored manually* and *delivered automatically*: they are *automatically compiled 'on the fly' at viewing time*.

Approach. Our aim is to build a genre independent storytelling system that can deliver ShapeShifting programmes of quality comparable with that of traditional, linearly edited, movies or TV programmes.

We approach this task from a narrative- or story-centric, but not plot-centric perspective. Individual narrations are automatically compiled on the basis of an artefact that subsumes all the narrative structures required for the respective narration. We label such an artefact as a *narrative space* and individual narrations as *narrative threads*. The narrative space is authored. Narrative threads are computed from it automatically by the storytelling system on the basis of the engagers' interactions. Each narrative thread is a number of sequences of media items that are played in parallel.

We take a *declarative approach* to the modelling of the narrative space, by separating the symbolic representation from reasoning procedures. Each narrative space has its own symbolic representation. The reasoning procedures, collectively called the *reasoning engine*, are generic. The basic architecture of a system that supports ShapeShifting programs is depicted in Figure 1. We have implemented a fully working system in this architecture.

Knowledge Components. The narrative space has three main components: a domain ontology; a media item repository, which contains a description of the characteristics of each media item; and a representation of the narrative space structure.

The *narrative space structure* is the component that specifies how media items are to be compiled into individual narrative threads. When the distinction is not necessary, we use 'narrative space structure' and 'narrative structure' interchangeably. There could be many ways of representing narrative space structures, but we employ a logic-based language that we developed, called *Narrative Structure Language* (NSL). NSL captures elements of the grammars and rhetorical forms of the ShapeShifting screen media, whilst at the same time it contributes to their definition. The core of NSL is genre independent. NSL may be regarded as a combination of structure-, graph-, and script-based representation.

Media items are the basic ingredients of any narrative space structure. In NSL they can be specified both via direct references and indirect references in the form of expressions stating their required characteristics. The characteristics of each media item are expressed, as metadata, in a separate repository. Indirect referencing results in the decoupling of the narrative space structure from the media items. This allows a narrative space structure to be reused with different content. Narrative space structures could thus be regarded as *computational programme formats*.

For each program, the media item descriptions are made on the basis of an ontology; only attributes defined in the ontology can be used in the descriptions. Via definitions, the ontology can refine the terminology used in item descriptions, such that further characteristics can be inferred on the basis of those that are explicitly stated.

Obviously, the more comprehensive the ontology is, the easier it is to better specify the media items making up the narrative space structure. For example, a part of the domain ontology may define roles in the narrative on the basis of shooting style and subject content; for instance, a short closeup of the hero reflecting on his or her journey ahead may be used in exposition or resolution. Ontological definitions, however, require substantial authoring effort. Furthermore, reasoning with large ontologies may be too slow in the context of ShapeShifting programmes. There is a trade-off between the size of the ontology and how specific the references to media items in the narrative space structure are.

NSL. The Narrative Structure Language is the language in which ShapeShifting programmes are conceived and formally represented. Currently, NSL defines a *basic syntax* for interactive reconfigurable narratives, hence we refer to it as the *core NSL*. However, it is being continuously developed with new elements of screen language grammar and rhetoric. The core NSL is *generic* in that it is genre independent and can be used with any combination of types of media object. NSL is *declarative* with the exception of the support for context variables and dynamic content, as discussed below.

Any NSL compliant object is called a *narrative object*. Any narrative space structure is a narrative object. A narrative object is either atomic or structured. An atomic narrative object is a *media item*. In order to support the sketching and design of narratives in the absence of high quality material, NSL supports media items that do not have a reference to essence, called *placeholders*. A structured narrative object is a composition of other narrative objects. The current core NSL provides three primitive composition structures called *link*, *layer* and *selection group*. NSL imposes no restrictions with regards to the depth or order of the composition. The primitive structures are accompanied by sophisticated mechanisms for the specification of different composition features regarding the narrative objects to which they are being applied. Some of them are enumerated below:

- support for the specification of both *synchronous* and *asynchronous engager interaction*; these are modelled via an inbuilt structured annotation that can be associated with any narrative object;
- *implicit references* to narrative objects via expressions;
- totally *recursive primitive composition structures*;
- *disambiguation* and *default rules* (both user defined and predefined) for selection groups and for decision points in link structures;
- *alignment conditions* for narrative objects on different layers; essentially, these are constraints regarding timing of objects that are to be played concurrently;
- *synchronisation messages* between narrative objects within the same layer structure;
- given the recursive nature of the composition structures, alignment and synchronisation is practically possible between any narrative objects of the narrative space;
- *collision rules* that disambiguate contradictory alignment conditions messages;
- specification of *ordered sets of constraints* regarding the *selection* and *sequencing* of narrative objects within selection groups; the ordering represents strength of preference;
- selection groups with *dynamic content*, that is content provided at viewing time;
- *action points*, that is annotations of points in the narrative space, possibly within media items, with predefined actions; an action is interpreted when the respective point is reached in the interpretation of the narrative space; an action may be a segment of code, in which case the code is executed; this is called *code point*;
- *procedural hooks*; they are like action annotations, but with the effect of triggering an external procedure;
- *context variables* which can be used in conjunction with code point annotations;
- *know-how rules*; they are normative statements that encapsulate some expertise regarding the compilation and editing of narrative threads; they may make reference to any aspect of the narrative as long as that aspect is captured in the associated ontology or is directly represented in NSL; if a rule is expressed within a narrative object, then it applies to any sequence of media items computed on the object's basis to the effect of possibly amending the initially computed sequence.

Some of these mechanisms have been described in more detail in (Ursu et al. 2007a); a complete specification of the core NSL is given in (Ursu and Cook 2006). NSL has a readable syntax but is not intended to be used directly by media producers; we developed an Authoring Environment for this, which is presented in (Ursu et al. 2007b).

Reasoning. Time is not explicitly represented in NSL. However, media items have durations (fixed, variable/parameterised or infinite) and narrative objects can have time points, within themselves, with associated actions. These and the ordering structures of NSL are sufficient for the compilation of narrative thread playlists.

A narrative thread playlist is not computed in one go. The reasoning engine computes playlist *segments* iteratively, such as, for example, between consecutive interactions. A narrative thread playlist emerges via the appending of consecutive segments. A segment is a set of sequences of media items of determined length, all the sequences having to be rendered and played in parallel.

The reasoning engine has two components: the *inference engine* and the *optimiser*. The inference engine is simply an interpreter for NSL. The optimiser is a repository of heuristics that ensure the continuous delivery of each narrative thread.

The interpretation of a narrative space structure must always begin from the start object and continue according to the order expressed by the structures of the narrative space. The traversed media objects are added to the playlist. The position reached at a certain point in time in the narrative space together with the playlist generated up to that point and all the received interaction represent the *narrative state* at that time for that specific narrative thread. The inference engine, essentially, is a function that, given a narrative state at time t and a set of engager interaction, computes a playlist segment corresponding to the interval (t, t+Δt), and the new state at t+Δt. The interpretation could go as far as to the first encountered expression that cannot be evaluated due to the absence of engager input, therefore Δt could be computed. An alternative way of invoking the inference engine is by specifying Δt.

Essentially, narrative threads reach the engager by means of the following parallel but interdependent automatic processes: *reasoning* – compilation of playlist segments; *delivery* – supply of essence, if it is not at the engager's end, rendering and playout; *interaction capture* – getting the interaction from the engager to the reasoning engine. Each process is directly reflected in a system component in our software (refer to Figure 1).

The interface between the reasoning and delivery processes is a *queue* of playlist segments. The reasoning process pushes into the queue and the delivery pulls. A delay normally occurs between pulling a segment and playing it, for example due to rendering, and it is amplified if the essence has to be supplied over broadband to the engager's delivery platform. Such delays must be accommodated by *reasoning ahead* of the real delivery time in the computation of playlist segments. However, reasoning ahead is in turn constrained by delivery, as engager interaction is cued by the actual play-out. Further, the reasoning process itself may take considerable time in the computation of certain playlist segments, not to mention the delays caused by the back channel.

The optimiser encapsulates the intelligence required to ensure that playlist segments are available when needed. For example, resolving the timing issues mentioned above is the responsibility of the optimiser. It decides when to invoke the interpreter and with which time span Δt. The optimiser may even modify a narrative segment, thus overriding the author, in order to ensure the continuous delivery of the narrative thread. For example, the optimiser may infer that an engager input could not arrive in time for a set of constraints expressed on a narrative object to be solved. In such a case, it may decide to use the default value of the interaction, calculate the solution and remove the cue and space for interaction from the playlist. Some of these problems, though, may be identified at authoring time via semantic tests and signalled to the author.

Conclusions

We have developed means for *thinking about*, *discussing*, *authoring*, *testing* and *delivering* ShapeShifting screen media. Our approach together with the accompanying system have been validated via a number of professional productions, notably through the dramatic production Accidental Lovers (Tuomola et al. 2006). To our knowledge, we are the first to have fully realised a truly interactive dramatic production delivered on a national television channel, authored and delivered with a generic software system.

This paper focussed on the computational model we developed for ShapeShifting screen media. At this end, there are a number of directions we are considering for further developments. Some are already under development:

- refine higher level structures for NSL, possibly genre specific, that are closer to the author's thinking than to the mechanics of the composition;
- devise a more comprehensive compilation of semantic tests;
- develop a more powerful optimizer; this is strongly linked to the previous aim.

Some are only at the planning stage. They are mainly related to the capturing and incorporation of authoring expertise in the different components of the model/system. They include:

- devising a declarative language for the expression of heuristics in the optimiser;
- refining an authoring knowledge base and mechanisms to apply such knowledge during authoring, to constrain, advise or even carry out more routine tasks of authoring, and during the compilation of narrative threads.

Incorporating authoring expertise was an initial goal when we started this research work, but we needed the basic model refined and the basic system implemented, before we could start to work on this issue. We believe that the foundations have been successfully laid.

Acknowledgements

This work was carried out within the European Union's 6^{th} Framework Programme Integrated Project "NM2: New Millennium, New Media" (IST-2003-004124).

References

Bulterman, D.C.A.; and Hardman, L. 2005. Structured Multimedia Authoring. *ACM Transactions on Multimedia Computing, Communications and Applications* 1(1): 89-109.

Cavazza, M.; Charles, F.; and Mead, S.J. 2002. Character-Based Interactive Storytelling. *IEEE Intelligent Systems*, Special Issue on AI in Interactive Entertainment: 17-24.

Mateas, M.; and Sengers, P. 1999. Narrative Intelligence: An Introduction to the NI Symposium. *In the Working notes of the Narrative Intelligence Symposium*, *AAAI Fall Symposium Series*. Menlo Park: Calif.: AAAI Press.

Riedl, M.; and Young, M. R. 2006. From Linear Story Generation to Branching Story Graphs. *IEEE Journal of Computer Graphics and Applications*, 26(3): 23- 31.

Tuomola, M.L. (director); Saarinen, L.E (writer); and Nurminen, M.J. 2006/2007: Accidental Lovers ("Sydän kierroksella"), Crucible Studio, Helsinki University of Art and Design Finland. Broadcast on Channel TV1 by YLE, The Finnish Broadcasting Company.

Ursu, M.F; Cook, J. 2006. D5.9: Languages for the Representation of Visual Narratives. *Restricted project deliverable in the EU FP6 IST-004124 IP project "NM2: New Media for a New Millennium"* (permission required).

Ursu, M.F.; Cook, J.J.; Zsombori, V., Zimmer, R.; Kegel, I.; Williams, D.; Thomas, M.; Wyver, J.; and Mayer, H. 2007a. Conceiving ShapeShifting TV: A Computational Language for Truly-Interactive TV. To be published in the Proceedings of the *EuroITV 2007 Conference*, Amsterdam.

Ursu, M.F.; Cook, J.J.; Zsombori, V.; Kegel, I.; Stevens, T.; Mayer, H.; and Hausenblas, M. 2007b. Authoring Environment for ShapeShifting Screen Media Productions. Published in this Proceedings (AIIDE 2007).