Towards Accessible Authoring Tools for Interactive Storytelling

Ulrike Spierling[1], Sebastian A. Weiß[2], Wolfgang Müller[2]

[1] FH Erfurt, University of Applied Sciences, Erfurt, Germany
[2] PH Weingarten, University of Education, Weingarten, Germany

spierling@fh-erfurt.de, weiss@ph-weingarten.de, muellerw@ph-weingarten.de

Abstract. This contribution presents and discusses *Scenejo* as an experimental platform for Interactive Digital Storytelling, focusing on the authoring process as initial viewpoint for its development. Special emphasis is on the construction of conversational threads for virtual actors using pattern matching, employing transition graph representations as the main interface for authoring. In the conclusion, the opportunities and challenges of graph structures are discussed.

Keywords. Interactive digital storytelling, authoring, transition graph, conversational storytelling, visualization of dialogues

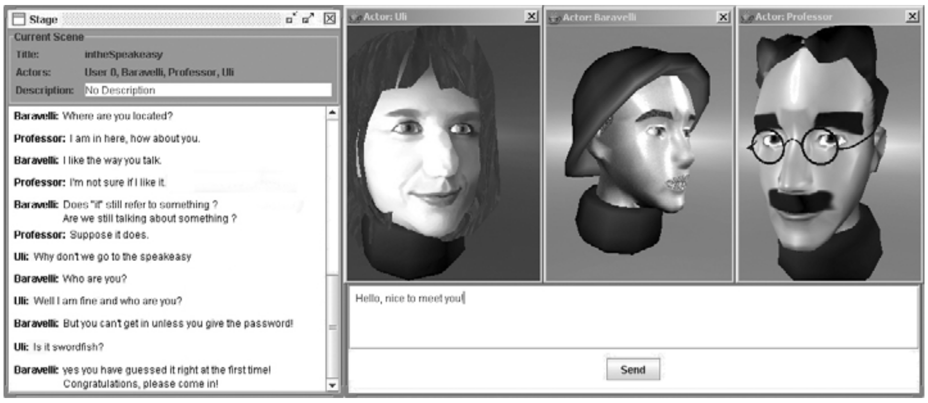
1 Introduction

Interactive Digital Storytelling (IDS) provides opportunities for the future creation of rich media applicable in entertainment and education, combining aspects of story, simulation and games. In current IDS research and applications, however, there are many open issues and unsolved problems. This is especially the case with applications that support conversational interaction styles with digital agents, providing text input style in natural language, instead of using the artificial game vocabulary most established in contemporary computer games. Examples of open research issues are (from a creator's point of view):

- **Authoring**: How do authors / writers access tools and methods for the creation of dialogues with digital agents, without programming?
- **Emergence**: How do authors get a grip on the course of an agent-based conversation, while allowing emergent behaviour resulting from an interaction style with few constraints?
- **Visualization**: How can variations of verbal conversations be represented visually for creators and planners of the dialogue?

In this contribution, we focus on these questions and present *Scenejo*, an experimental platform allowing practical experiences with agent conversations for non-programming authors. *Scenejo* uses the simplest of technologies for the control of the verbal dialogues, employing the public licence chatbot technology A.L.I.C.E. [1], which supports accessibility for authors who enter the field as novices in dialogue programming. *Scenejo* allows authors to either use plain AIML[1] to define a so-called knowledge base as written dialogue lines for each virtual actor, or to create more structured conversations using transition graphs in a graphical interface. The *Scenejo* platform is then capable of immediately playing the created dialogues between several agents, rendering them in real time as talking heads (compare Fig. 1) using TTS, and allowing users to participate in the conversation by typing text.[2]

In this contribution, we present the authoring tools conceived in *Scenejo*, including a discussion of the usefulness of specific visualizations for several steps of creation in an interactive storytelling project.



Title: intheSpeakeasy
Actors: User 0, Baravelli, Professor, Uli
Description: No Description

Baravelli: Where are you located?
Professor: I am in here, how about you.
Baravelli: I like the way you talk.
Professor: I'm not sure if I like it.
Baravelli: Does "it" still refer to something ?
Are we still talking about something ?
Professor: Suppose it does.
Uli: Why don't we go to the speakeasy
Baravelli: Who are you?
Uli: Well I am fine and who are you?
Baravelli: But you can't get in unless you give the password!
Uli: Is it swordfish?
Baravelli: yes you have guessed it right at the first time!
Congratulations, please come in!

Hello, nice to meet you!

Fig. 1. Conversation in *Scenejo*, mixing emergent small talk and directed scene transitions.

2 Related Work

With the concept of *Scenejo*, we follow an approach towards creation possibilities in a middle ground between predefined and emergent simulated dialogues, while thinking radically author-centred. This does not mean that we obtain a viewpoint in IDS that prefers author-driven plot to character-based story development. Rather, it indicates that authors should be able to define the way a conversation develops at any given time – including the occasional event of emergent conversation. Evidently, there are natural borderlines in complexity that are difficult to cross for any human author, which will be stressed again in the conclusion of this article. Having said that, we disagree with a generalized viewpoint that makes the "ability to program" a prerequisite for potential authors to enter the field of IDS, as supposed by Mateas and Stern in [10].

Examples of existing systems with a similar performance goal as *Scenejo* are Façade [11], art-E-fact [13] and CrossTalk [6].

---

[1] AIML: Artificial Intelligence Markup Language [1]

[2] The *Scenejo* use concept has been explained in [18], and the operation of the drama manager being responsible for the control of the interlocutors' turn taking is described in [12].

Façade is well known in the IDS community, being the first - and so far only - working system of its kind that enables natural language interaction, based on keyboard input, with virtual characters. It enables complex structures for virtual character behaviour, employing rules that reduce user utterances to context-based semantic units and for the selection of so-called "beats" as the next performed action. The downside of its complexity is the huge obstacle that an author would have to overcome should she wish to create a new and different story than the provided one with Façade's programming language ABL. There is no concept as of yet for an authoring tool, which would be an interesting next step according to the creators of Façade.

The project art-E-fact [7] was a foundation for developments aiming at a similar goal as Façade, but focusing on the authoring aspect from the start. With art-E-fact's authoring tool Cyranus [8], from the outset, a visualization of transition networks is offered. It consists of nodes containing dialogue moves of several characters, and edges defining the pre- and post-conditions for their execution. In the beginning, the problem with this approach was the likelihood that authors would come up with rather linear plots, supported by the affordance of the transition graph tool as the only means of creation. Cyranus is currently being enhanced by rule-based possibilities, which, in the first instance, demand that authors program these rules in Jess[3].

CrossTalk [6] is described as a dialogue system for animated presentation agents using plan-based dialogue generation, as well as a corpus of pre-scripted scenes. The generative part is based on automatic presentation agents [3]. For the development of pre-scripted scenes, the tool SceneMaker has been developed, working with cascaded finite state machines. A scene is any reusable module that can be referred to in a conversation, carrying meta-information to maintain context. A dialogue compiler transfers pre-scripted scenes into plans. While non-programmers are addressed by the system, currently, the tool is mainly a script language to be written with a common text editor, involving some programming expertise.

In a wider context, existing creation tools for building human-computer speech dialogue systems, using transition graphs [9], are also relevant, as well as editors for chatbot applications, supporting the definition of huge dialogue bases, e.g. for AIML [2]. These methods do not involve conversations between several digital agents, but can be adopted to solve subtasks within the *Scenejo* concept.

We conclude that while the beginnings of *Scenejo* do not yet achieve the complexity in calculated emotional depth that is accomplished in Façade, nor the contextual meta information that is followed in Crosstalk, continuing research is justified by its initial viewpoint and main focus on accessibility for writers and authors, similar to that found in art-E-fact. In contrast to art-E-fact, AIML authoring of a single chatbot's knowledge base is the first access to the creation of a conversational scene, providing at least chatbot emergence from the outset, instead of at first constraining the dialogue into the corset of a transition network. The latter, however, is introduced later as a visualization tool and to provide structure.

The next section explains the details of the concept, showing where overall plot management and character-based dialogue creation meet in the interface. A result of the work with the experimental platform, we discuss the opportunities and challenges of transition network visualizations for Interactive Digital Storytelling.

[3] Jess rule system: http://herzberg.ca.sandia.gov/jess

3 Transition Graphs and Pattern Matching for Creation and Representation

In this section, details of the conceived *Scenejo* authoring concept are explained. Special emphasis is placed on the use of transition graphs as a means for structuring content – on several levels of abstraction in interactive storytelling.

3.1 Story vs. Plot

Story

Structural principles of classic stories focus on interesting characters with traits and goals, in constellations with antagonistic forces. Basic abstract functions are sketched in Figure 2 (right) as an adaptation of Frank Daniels' character-driven drama model suggested by Struck [16].

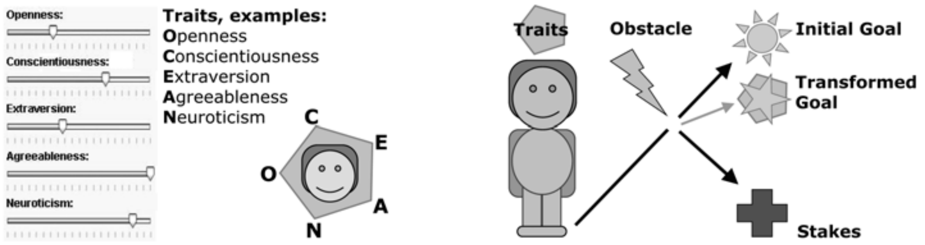


Fig. 2. Left: Trait tuning as currently arranged in the interface. Right: Abstract functions of character-based story, before plot generation.

In *Scenejo*, these structures are currently only implicitly supported. That is to say, if authors have an ambition to structure a story in a dramatic way, they define characters with trait parameters and emotional states. At the current stage of development, states can be set and evaluated as conditions within the dialogue, influencing the further development of a plot line. More sophisticated calculation models are planned for future work, as *Scenejo* currently mainly supports plot generation.

Plot

Character transformation forms the story, while telling over time forms the plot. The task of a storyteller is to restructure the story into a plot line, forming smaller elements, such as acts and scenes (compare Figure 3, Left). For interactive storytelling, there have been debates as to whether a predefined plot line is a desirable goal at all, since the interactive plot rather unfolds according to interactions of participants / players. In our project, we follow the hybrid concept suggested by Spierling [14], using several levels of semi-autonomy at which authors can define a balance between predefined and user-centric courses of action. Accordingly, in *Scenejo*, it's up to the author to either decide to arrange a prestructured plot line, or, to let conversations emerge based on previous utterances and state changes. Figure 3 (right) shows an example plot graph in *Scenejo* containing five scenes, while two of the scenes are apparently alternatives depending on user interaction. Within a scene, authors can define more detailed dialogue structures or just AIML dialogue bases for each actor (see section 3.3).

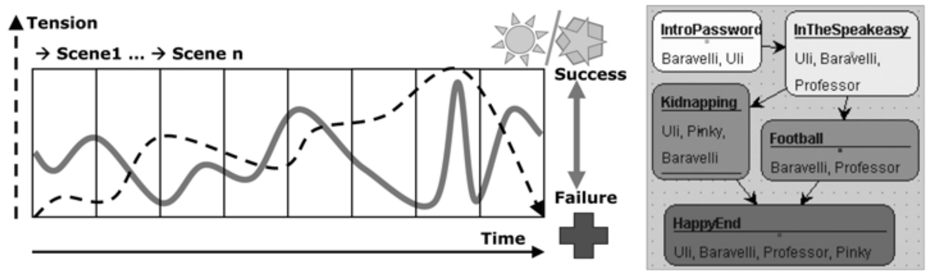


Fig. 3. Left: Telling over time, including scenes, separated by important plot points. Right: Plot graph in *Scenejo*, containing scene elements connected by scene transitions.

A “scene” element in *Scenejo* is a means to separate potential dialogues and different reactions to users according to a timed order. Put simply, it is a visualization of making potential utterances state-dependent. To achieve the same result technically, authors can also use only one scene and instead define conditions based on AIML predicates for structuring. This, however, increases conceptual thinking in terms of programming. The arrangement of several scenes, each working towards or against the character goals to provide tension, serves rather as a metaphor for cinematic story structure. Whenever more than one successive scene is defined, it is necessary to also define the transition conditions between them. Currently, these conditions can be the elapsed time, certain user utterances or state changes of a character predicate. Plot points, therefore, can be modelled as states to be reached through conversation. In the example in Figure 1, guessing the right password was the trigger to change the scene.

Technically, a scene is a sample of AIML knowledge bases available at a certain stage, building the search space for a pattern matching process. AIML is always attached to one particular actor represented by a specific chatbot. Therefore, the definition of a scene initially means to specify its cast and associated AIML. Categories of each actor’s knowledge base can be either permanent or scene-dependent, and therefore loaded or discharged during a transition between the scenes.

3.2 Chatbot Knowledge Base vs. Structured Dialogues

Stimulus-Response Principle

A chatbot with a knowledge base as mentioned above works through pattern matching following a simple “stimulus-response” principle. The simplest AIML element is a category, containing a pattern and a template. For each chatbot, a pattern defines a searchable word structure (receiving a matching “stimulus”), drawing a straight line to an attached answer template (the “response”). Further, more complex templates can be written that, for example, redirect stimuli to other patterns to allow synonym definitions, or that provide a random list of possible responses to support flexibility. Also, so-called “predicates” can be filled with values stemming from user utterances (or from internal state changes) to be used later during the conversation. Minimal dialogue context can be provided with AIML by defining the last utterance to get an answer for, or by setting a topic to mark a list of tagged categories with higher contextual priority. In summary, the language processing capabilities are far from being as rich as those found in NLP systems supporting grammar and semantic models.

One initial way to begin working with *Scenejo* as an author is to write AIML as a knowledge base for each actor using a text editor or available AIML creation tool. The following conceptual issues were discovered during the first authoring attempts and have been addressed in *Scenejo*:

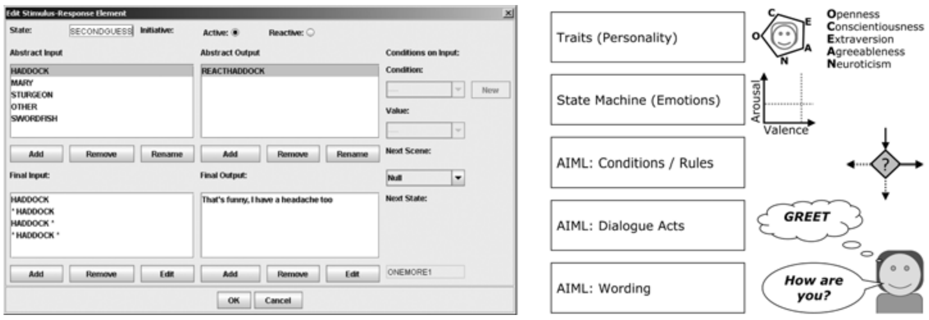
- Patterns always represent concrete wording, while the possibilities of utterances (stimuli) in natural language appear to be endless. Therefore, an intermediate structure is introduced, serving as a pseudo-semantic level, herein called “dialogue acts” or “abstract input/output”. This is summarized in a “stimulus-response” element in the authoring interface (compare Figure 4).
- Changing initiative between actors in a dialogue is difficult to handle, due to the stimulus-response nature of AIML, making every possible utterance an “answer” dependent on a “question”. Therefore, in addition to a “stimulus-response” element, an “initiative” element is introduced that is independent of a stimulus.
- Each written category represents a one-directional view of one chatbot towards any interlocutor, originally a human user. Modelling a dialogue between two actors / bots means also mirroring the opposite side, turning templates into patterns. A first interface has been conceived to support writing bot-bot dialogues, resulting in interlocked patterns and templates providing the right “hot words” for the other bot.

The Stimulus-Response Element (SRE) in the *Scenejo* Interface

In addition to writing AIML directly (which is always possible), the Stimulus-Response Element (SRE) supports the planning of dialogue lines by an intermediate abstract structure. Metaphorically, this is comparable to using reported speech in a story treatment and fixing concrete utterances later in the refinement stage of the script.

Figure 4 shows a graphical editor for the SRE, as well as a concept as to how this fits into a level structure for dialogue planning. Authors can conceive a conversational turn on the dialogue act level (as abstract input and output in the upper level of the editor) and provide concrete utterances in the lower level of the editor. Each abstract stimulus defined on the left has to be answered with at least one abstract response on the right. By using conditions for distinction, more than one alternative abstract response can be defined, which can eventually lead to different follow-up SREs in a dialogue graph, or to a scene change defined here by the author. Conditions may depend on property states from memory, emotional and other “predicate” states, as well as values from user utterances. In section 4, the mapping of this structure to AIML is explained technically (compare Figure 7).

The SRE is the start to structuring many SREs into a dialogue graph, while it actually defines the content of one node in the transition network, as well as some of the transition rules. This is explained in the next subsection.


Edit Stimulus-Response Element

State: SECONDGUESS Initiative: Active: [✓ ●] Reactive: ○

Abstract Input: HADDOCK, MARY, STURGEON, OTHER, SWORDFISH

Abstract Output: REACTHADDOCK

Conditions on Input: Condition: — Value: —

Next Scene: Null

Final Input: HADDOCK, * HADDOCK, HADDOCK *, * HADDOCK *

Final Output: That's funny, I have a headache too

Next State: ONEMORE1

Fig. 4. Left: SRE-Editor for a Stimulus-Response Element: Abstract utterances (upper levels of input / output) can be used to plan the dialogue. Right: Interdependences of dialogue acts with other levels of parameters in *Scenejo*. The three lowest levels are contained in the SRE.

3.3 Dialogue Graphs

The *Scenejo* dialogue graph editor can be used to "draw" conversational threads, simply by combining several SREs with arrows. Through connecting certain SREs to one graph, a situational context constriction is generated, resulting in situational preferences for verbal pattern matching of one actor. A scene can contain multiple conversational threads (each represented by a dialogue graph; compare Figure 5). In total, three kinds of matching AIML patterns can determine the next utterance of an actor: the patterns within the AIML base associated with the actor (e.g. for general small talk), the patterns associated only with the current scene (scene context knowledge), and the patterns within a dialogue graph (situational within a specific thread of conversation). The search priority, quite naturally, puts the most specific (here: one dialogue graph) first.

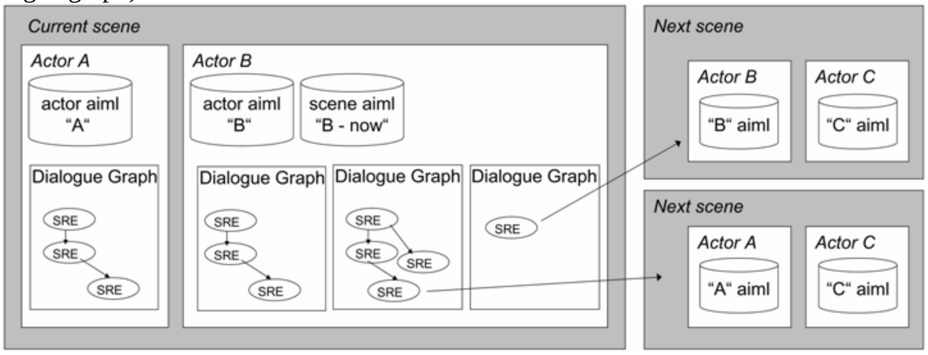


Fig. 5. The hierarchical concept: Dialogue graphs show one actor's potential conversational threads within a scene. Together with associated plain AIML, they form the knowledge base.

Figure 6 shows the graphical user interface. The dialogue graph also allows combinations of SREs that are not dependent on any matched pattern, by turning its mode to "initiative". This is an opportunity to give an actor the initiative in a dialogue, unlike original AIML concepts. Further, a dialogue graph represents one conversational thread from the viewpoint of one actor. In order to support the writing of a fixed dialogue between two actors, a dialogue graph of an interlocutor is loaded into the editor in reversed structure.

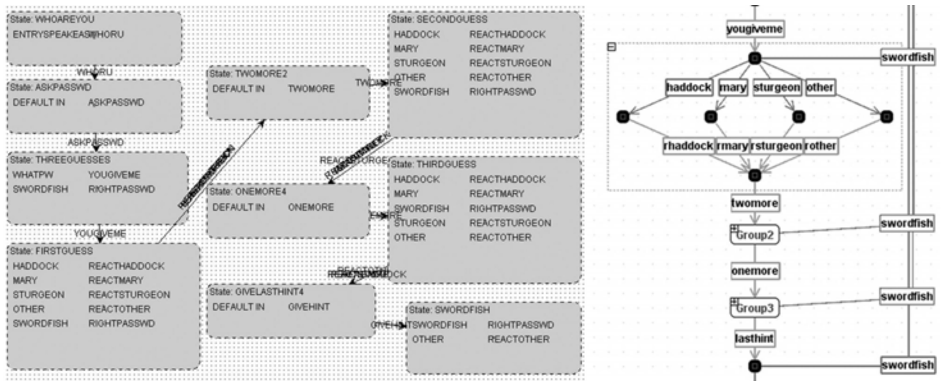


Fig. 6. Two versions of the dialogue graphs in *Scenejo*. The version on the right shows the option of visually collapsing nodes during editing for better clarity.

4 Implementation

There are two main tasks of implementing editable dialogue graphs. First, such graphs have to be represented by a data structure (in this case, primarily AIML), and second, a graphical tool for editing such structures has to be provided to authors as an alternative to simple text editing, letting them manage complexity more easily.

4.1 Mapping Dialogue Graphs to AIML

The implementation of the dialogue graph is actually made up of the implementation of its elements. A single SRE can be understood as one <topic> in AIML.

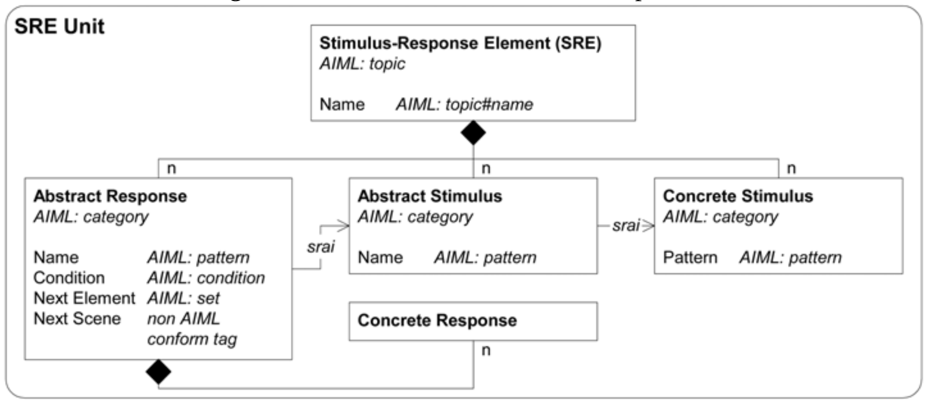


Fig. 7. Schematic view of a SRE and its AIML components.

As shown in Figure 7, abstract stimuli, concrete stimuli and abstract responses are implemented by the <category> tag. By combining all categories of one SRE into one unique topic name, these categories in the overall knowledge base of an actor are marked off from others imported from further plain AIML files.

4.2 Visual Authoring of Plot Lines and Dialogues

As explained in section 3.3, a conversational thread can be modelled visually in the form of a dialogue graph that is made up of several SREs. At this level of granularity, authors can also define possible state changes of an actor. In other words, one can define the influencing rules, based on set AIML predicates of an actor, that determine the selection of an explicit response. Further, for incremental changes of any memory and emotional states (AIML or other predicates) of an actor, additional non-AIML constructions have to be implemented that support the state machine. For example, this is currently the case with scene changes.

Figure 8 shows in more detail how these dependencies are implemented.

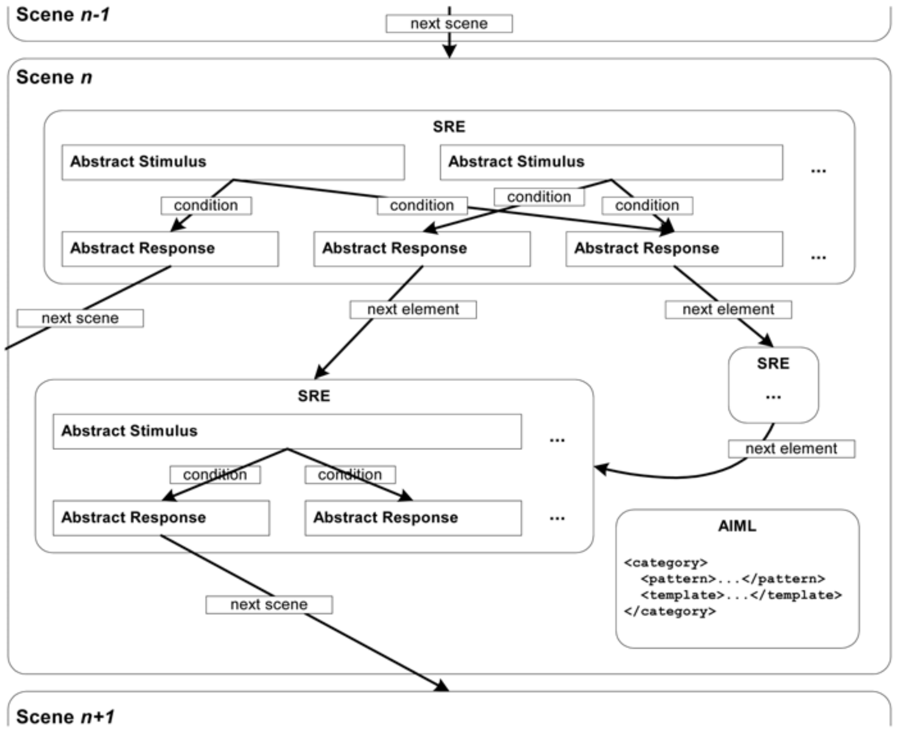


Fig. 8. Detailed view of scene transitions triggered from a SRE in a dialogue graph.

This concept represents the conversation as a collection of transition graphs of conversational threads. States are represented by nodes, transitions by edges. A transition within the dialogue graph connects one particular abstract response in a SRE with one successive single SRE.

5 Conclusion

5.1 Authoring and Emergence

Scenejo serves as an experimental platform to experience authoring processes. As such, during the creation of initial demonstrators and content with the prototype, more open issues for the development of authoring tools have been recognized.

First, the visual plot line of scenes appears to be a reasonable way to start structuring on the highest abstraction level. In general, the affordance of a directed graph as a tool very often results in linear structuring – which is not as much of a problem on the highest abstraction level. On the lower levels, such as those of the dialogue acts, however, the experience of emergence is much more important. For example, more emergence can be experienced when a huge pattern base outside of a few constrained dialogue graphs is provided. So far, the experiments have shown that typical chatbot principles of redirecting sentences back to the interlocutor result in entertaining small talk, which can enhance the impression of a dynamic dialogue construction. On the other hand, this is less goal-directed than when more structured threads are involved. Authors depend a lot on tuning and testing when emergence is introduced, so including immediate playback facility in the tool is essential.

For first time users, the graph tools provide accessibility. Frequent users, by contrast, may experience usability problems in terms of low efficiency. Especially the -authoring of a two-directed conversation requires an extension of the dialogue graph tool, resulting in additional views that need to be rather text-based for better facility of inspection. In the future, text interfaces shall be mixed with graphical interfaces for that reason.

Graph representations have the advantage of building recognizable visual patterns. Authors are able to build primitives of reusable patterns of conversational threads. Their visual structure allows for identifying them among others as a unique iconic form.

The SRE editor allows to summarize and structure possible utterances into more abstract meaning, and to plan a dialogue based on this abstract level. As a consequence, if an author would like to reduce all possible stimuli to only a limited set of "artificial language" verbs, this is possible with the system.

5.2 Future Work: Visualizing Large Transition Graphs

While the plot and dialogue graphs implemented so far are still of limited complexity, their size has already made a visualization of these graphs difficult. The limited display area hardly allows for a depiction of the complete graph on the screen. Also, the manual layout of larger story and dialogue graphs is cumbersome and time-consuming. A partial representation of story graphs, however, makes it difficult for a user to grasp the story line and to edit the story graph. Similar, an ineffective representation of a dialogue graph hampers the understanding of possible alternatives and developments in a dialogue of an actor. It is apparent that more complex story lines and dialogues will only add to this problem in the near future.

The method for modelling plot and dialogue in *Scenejo* can be understood as an approach of visual programming. Consequently, the visualization of transition graphs can be related to the problem of visualizing flow graphs in software visualization. In terms of modelling, the sequence of actions corresponding to the Unified Modeling Language, this corresponds to a representation of a UML state chart.

There have been several approaches to the visualization of transition graphs using automated layout algorithms. Algorithms can be distinguished into those considering aesthetic criteria only (e.g. edge crossings) and those allowing for additional constraints. Algorithms are typically either force-based or apply layered layout strategies algorithmically [17]. While most of these techniques are limited to represent directed acyclic graphs, they can be adapted to visualize cyclic structures by cutting cycles appropriately. Layered strategies are in general more appropriate to present directed graphs[4]. Follow-up experiments in *Scenejo* include the application of different layouters.

For the display of larger graphs on limited display areas, Focus & Context techniques, such as "Fisheye Views" [5], have been applied, which render those nodes and transitions that are currently in focus in more detail [15]. In general, an effective application of Focus & Context techniques demands a careful and in-depth analysis of the application context to exploit the additional space of detail information effectively. We are currently analyzing the user requirements on information detail during the authoring process based on *Scenejo* user surveys.

Further, the graph complexity can be reduced by mechanisms for graph abstraction. A pattern library of conversational threads can provide reusable collections of dialogue acts, either stored and reused by authors or provided as primitives. In the resulting transition graph, patterns can be represented by a single abstracted node element, thus simplifying the graph.

Acknowledgements

Thanks to Florian Steimle for many fruitful discussions on authoring concepts during his Diploma thesis, and for his contributions to the existing prototype of *Scenejo*.

References

1. ALICE. Homepage of the A.L.I.C.E. Artificial Intelligence Foundation. Online: http://www.alicebot.org (Last accessed: 07.31.2006)
2. ALICE. News Archive of the A.L.I.C.E. Artificial Intelligence Foundation. Online: http://www.alicebot.org/oldnews2006.html (Last accessed: 07.31.2006)
3. André, E., Rist, T.: Presenting through performing: On the use of multiple animated characters in knowledge-based presentation systems. In: Proceedings of IUI'00, ACM Press (2000) 1-8
4. Di Battista, G., Eades, P., Tamassia, R., Tollis, I.G.: Graph Drawing: Algorithms for the Visualization of Graphs. Pearsons Pub. (1998)

[4] For more details on graph drawing see [4].

5. Furnas, G.: Generalized Fisheye Views. In: Proc. CHI`86, (1986) 16-23
6. Gebhard, P., Klipp, M., Klesen, M., Rist, T.: Authoring scenes for adaptive, interactive performances. In: Proceedings of the Second International Joint Conference on Autonomous Agents and Multi-Agent Systems. Melbourne, Australia, ACM Press (2003)
7. Iurgel, I.: From another point of view: Art-E-fact. In: Proceedings of Technologies for Interactive Storytelling and Entertainment, Darmstadt (2004) 26-35
8. Iurgel, I.: Cyranus – An Authoring Tool for Interactive Edutainment Applications. In: Proceedings of Edutainment 2006, International Conference on E-learning and Games, Zhejiang, China (2006)
9. Klemmer, S.R. et. al.: SUEDE: A Wizard of Oz Prototyping Tool for Speech User Interfaces. In: Proceedings of UIST (2000)
10. Mateas, M., Stern, A.: Procedural Authorship: A Case-Study Of the Interactive Drama Façade. In: Proceedings of Digital Arts and Culture (DAC), Copenhagen (2005)
11. Mateas, M., Stern, A.: A Behavior Language: Joint Action and Behavioral Idioms. In: H. Prendinger and M. Ishizuka (eds.): Life-like Characters. Tools, Affective Functions and Applications, Springer (2004)
12. Müller, W., Spierling, U., Weiß, S.: Synchronizing Natural Language Conversation Between Users And Multiple Agents In Interactive Storytelling Applications. In: Proceedings TESI 2005, Maastricht, The Netherlands. (2005)
13. Spierling, U., Iurgel, I.: “Just Talking About Art” – Creating Virtual Storytelling Experiences in Mixed Reality. In: Virtual Storytelling, Proceedings ICVS 2003, Springer Berlin-Heidelberg (2003)
14. Spierling, U.: Interactive Digital Storytelling: Towards a Hybrid Conceptual Approach. In: Proceedings of Digital Games Research Association’s 2nd International Conference “Changing Views: Worlds in Play”, Vancouver (2005)
15. Storey, M.-A.D., Müller, H.A., Wong, K.: Manipulating and Documenting Software Structures using SHriMP Views. In: Proceedings of the 1995 International Conference on Software Maintenance ICSM'95. Opio (Nice), France, October 16-20 (1995)
16. Struck, H.G.: Telling Stories Knowing Nothing: Tackling the Lack of Common Sense Knowledge in Story Generation Systems. In: Virtual Storytelling, Using Virtual Reality Technologies for Storytelling, Third International Conference, Proceedings, Strasbourg, France (2005) 189-198
17. Sugiyama, K., Tagawa, S., Toda, M.: Methods for Visual Understanding of Hierarchical Systems. IEEE Trans. Systems, Man, and Cybernetics Vol. 21(2), (1981) 109-125
18. Weiß, S., Müller, W., Spierling, U., Steimle, F.: Scenejo – An Interactive Storytelling Platform. In: Virtual Storytelling, Using Virtual Reality Technologies for Storytelling, Third International Conference, Proceedings, Strasbourg, France (2005) 77-80