Adding Aspects of “Implicit Creation” to the Authoring Process in Interactive Storytelling

Ulrike Spierling

FH Erfurt, University of Applied Sciences,
Altonaerstr. 25, 99085 Erfurt, Germany
spierling@fh-erfurt.de

Abstract. In Interactive Digital Storytelling (IDS), new design processes are needed for authors to fully embrace the emergent properties of this dynamic novel medium. As a counterpart to traditional explicit authoring in storytelling, “implicit creation” is introduced as a conceptual framework. By describing the development steps of a specific IDS application – the Killer Phrase game – and starting out with explicit methods, the conditions for adding implicit creation are discussed, which lead to suggestions for future research.

Keywords. interactive digital storytelling, implicit creation, authoring, emergence, emergent narrative, storyworld.

1 Introduction

Interactive Digital Storytelling (IDS) provides opportunities and challenges for the future creation of entertainment and education applications by combining aspects of story, simulation and games [18]. A difficulty frequently emphasized in IDS is that of maintaining coherence in a story with an interesting plot while also allowing the audience to interact with it. This challenges traditional ways of conceiving stories, because either the participant’s interaction can destroy a well-crafted plot line, or the author’s plot may be so tightly woven that interaction is reduced to only frustratingly meaningless choices.

This challenge is currently addressed by the research field of Interactive Digital Storytelling in the development of sophisticated story engines, which ensure the coherence of a story on behalf on an author at interactive runtime sessions. The task is particularly ambitious in realtime multi-agent environments, in which users can interact with a storyworld with only few restrictions. Prior to this user interaction, the authors of this experience have the important task of configuring the behaviour of individual agents and a story engine, a major component of the overall goal of content creation. This is the reason for the current interdisciplinary divide between engineers and people who have the ideas for content: designers, authors and writers. The divide is caused by difficulties with the process of creation. Aspects that have been recently discussed are:

- **Programming:** There has been much discussion about the requirement of the author's ability to program in order to achieve creative goals in IDS [11].
- **Anticipation of Emergence:** There has been less discussion on the general difficulty of anticipating the emergent course of actions generated automatically or semi-automatically by the interaction of agents (software agents and users). When the constraints for interaction are loosened, the "story" can take unexpected directions. The flow of the resulting "emergent narrative" can better be explained by the metaphor of role playing than by storytelling [10].

This contribution discusses both aspects, but with an emphasis on trying to get to the bottom of the aspect "emergence", and what it means for the creative process. "Emergent narrative" is a recently discussed concept, often identified as a paradox. The merit of employing a potential emergent system for IDS is here seen not as a goal in itself, but as a possible way to avoid the clash between pre-scripted narrative and the freedom offered and afforded by a virtual environment. An emergent property is one that exhibits perpetual novelty [7] each time the system runs, under the influence of a sequence of input combinations. Hence, it supports user agency by generating a variety of constellations at runtime, which are, on the other hand, hard to foresee, and can only be addressed "implicitly" by a designer.

After presenting related work, "implicit creation" is introduced as a concept opposite to "explicit" authoring. Issues of IDS creation are discussed with the help of an example application: the "Killer Phrase" game designed with the platform *Scenejo*. Based on conclusions drawn from this process, the next steps towards a conceptual model of implicit creation are extrapolated. These go beyond the vision that the main benefit of developing authoring tools is that they can prevent writers from programming by providing a GUI for the task of behaviour description. The concluding insight is that the coding itself is not the main problem, but that a whole new design process of Interactive Storytelling needs to be embraced by new authors, letting them define and configure the parameters of the story that "implicitly" lead to an interactive narrative experience.

2 Related Work

Current IDS implementations can have multiple forms, in which varying kinds of roles for participants, (virtual) actors and authors occur. The work presented in this paper focuses on a form that lets users and agents interact in real time with frequent turn-taking, leading to an emerging course of actions instead of following a pre-authored path. The term "emergent narrative" was introduced by R. Aylett [2]. The concept has been implemented in the demonstrator "FearNot" of the VICTEC project [3], which lets the user participate in a role play with virtual characters that are only defined by parameters. Each agent chooses actions based on a complex model of emotions and cognition. Currently, there is no authoring tool for the system; the virtual actors are directly programmed in XML-like structures. The architecture doesn't differentiate between a storyworld and models of the agents. A similar case has story engines based on HTN planning methods [4]. While authors can define a storyworld as a hierarchical ontological structure of goals and sub-goals for agents, it is hard to anticipate concrete actions that result from it. The project “Façade” [11] solves the anticipation problem by letting a drama manager take partial control over the flow of actions, while dialogue pieces are pre-authored within so-called “beats”. Content creators had to program the drama manager. The underlying creative concept has been called “Expressive AI” [12]. All aforementioned approaches make partial use of automatically generated dialogue by using dialogue templates with partially scripted components. Storytron is another story engine project, which has been under development for several years [5] and which differentiates between engine and storyworld. It contains the authoring tool “SWAT” based on “verbs” directly defining possible actions, events and states in a storyworld.

This paper is based on previous work with the conversational storytelling platform “Scenejo” [21], using the simplest agents based on AIML chatbots and uncomplicated dialogue management with finite state machines. In contrast to the examples cited above, Scenejo still leaves much of the explicit decision making to authors, who are supported by a graphical authoring tool. Only a few IDS systems exist with first authoring tools, some of which were presented in a TIDSE 2006 workshop [20]: Scenejo [21], Cyranus [9], Scribe [13], U-Create [16]. Other storytelling tools built in EU-funded research [8] [14] don’t address real-time multi-agent systems while they deal with cross-media and the constrained choice options in the TV-sector. With all these tools, the flow of the narrative plot can be directly defined by visual graph structures with explicit transitions between actions.

Some issues of creating IDS, such as artistic models and workflow, “Expressive AI” and storyworld creation problems, are debated in online forums[1]. In section 5, storyworld creation will be reconsidered. It can be concluded that there are story engines enabling forms of “emergent narrative” without accessible means of authoring and there are systems with authoring tools that allow for explicit definition of actions, following traditional metaphors of controlled content creation. As of yet, there is no real synthesis of the two. This is not only perceivable in the lack of respective tools, but, moreover, in the lack of relevant concepts that, in the following paragraphs, will be called “implicit creation”. Systems supporting “explicit” authoring are in fact meaningful for the development of the field, because they allow non-programming creators to access IDS and create useful examples for further research. Hence, they are a valid starting point to be further developed towards “implicit” creation.

3 Explicit Authoring vs. Implicit Creation

First of all, it is argued that the way story “writers” craft their work changes significantly with the introduction of interaction. This is especially true in the creation of a piece to be run on a multi-agent platform allowing “emergent narrative”, where it is impossible to define every little detail explicitly, in advance of the user interaction. The traditional method of creation, shaping every detail of the plot, is referred to herein as “explicit authoring”. In contrast, “implicit creation” becomes necessary, where configurations of states “imply” certain behaviours of agents. As for a concep-

[1] e.g., http://grandtextauto.gatech.edu/, http://storytron.com/smf/, http://tale-of-tales.com/

tual model of creation, new metaphors are needed to explain the main difference constituted by this indirect form of design.

Figure 1 illustrates this difference by using the metaphor of gardening (this metaphor was used previously by W. Wright for simulation games, such as SimCity [15]). The left side of the illustration shows that with explicit creation, a creator generates content that serves as the predefined template for the runtime narrative to follow the given form, such as is the case in branching interactive storytelling. In this metaphor, it is a paper flower, crafted explicitly by the author in full detail and beauty. By contrast, the right side – with implicit creation – shows that a runtime narrative actually has to be "planted" beforehand by a creator. All details emerge while the plant is growing, and untypical variations can also occur in the runtime narrative. The difficulty of crafting finally lies in the design of the "seeds", independent of any ability to code. As such, this sketched creative process is the same as found in the design of simulation games. Moreover, the vision to "get a grip" on the emergent process is likely to shape up as an illusion, as the definition of emergence implies. As J. Holland [7] says: *"Much comes from little"* – meaning that emergence is defined by its unpredictable attributes, letting few rules give rise to extraordinarily complex situations.

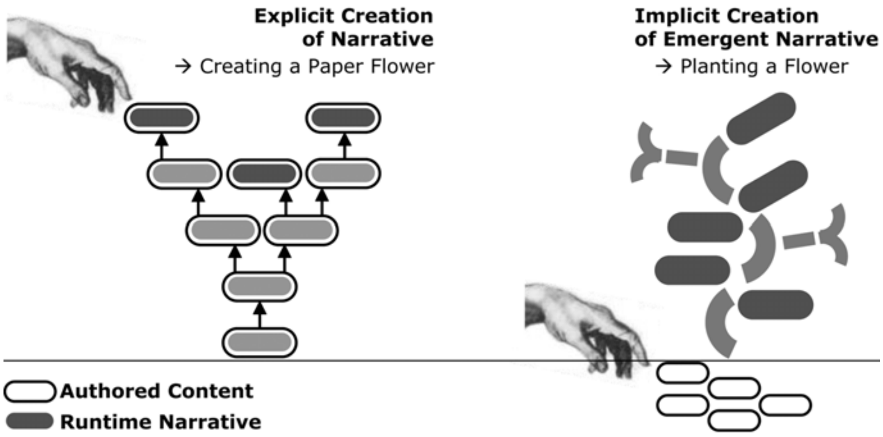


Fig. 1. The gardening metaphor explaining the difference between explicit and implicit creation for generating a narrative structure during runtime.

As mentioned before, IDS is considered to be a combination of story, simulation and games – therefore, the art form requires more than traditional story creation as a telling of events. For a simulation game, a critical design step is the design of a dynamic model. Figure 2 shows the general concept of transition functions building the core of a dynamic model [7]. According to Holland, a "perfect model" would result in a complete detailed mapping of real world configurations to model states and of laws of change in the real world to transition functions in the model. However, as he points out, the art of model building lies in selecting the "right level of detail" that is useful for the purpose, distinguishing salient features from the non-essential and capturing the laws of change at the chosen level. It is very unlikely that we manage to observe the world successfully in every detail, and doing so would result in a model that is way too complex.

In the following sections, it will be argued that, for IDS, it is this “right level” of detail that has to be identified in order to apply emergent features. In other words, the claim is made that the design of a model as an abstraction, leading to emergent behaviour of a “desired” sort, must be a part of the creational process and, as such, part of the content. The modelled behaviour is not (only) the responsibility of the underlying runtime engine, which more or less provides underlying environmental conditions (for example, psychological and physical models resembling “reality”). It will further be assumed that for each “interactive story” (aka “storyworld”) created, there is a need to identify the usefulness of “emergence” for its narrative elements and make this a part of the design workflow – considering that narrative properties exist that don’t necessarily have to show “perpetual novelty” (see section 1), and therefore don’t need to be modelled as emergent features, and can be scripted in a more traditional manner.

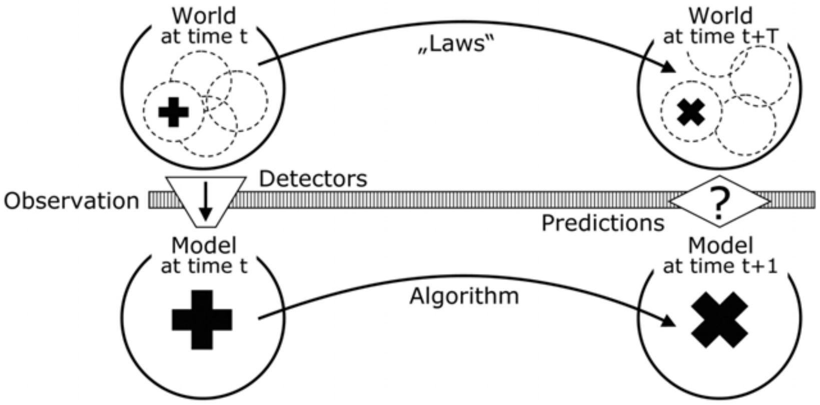


Fig. 2. The observed world configurations and their identified “laws of change” have to be transferred to a dynamic model of states and calculable transition functions (“algorithm”). (Illustration adapted from [7]).

The next section describes a case study, illustrating the experiences made with the model building and authoring process. Starting from an “explicit” view on authoring, first steps towards implicit creation are made. Finally, further needs are outlined with their consequences for implicit creation.

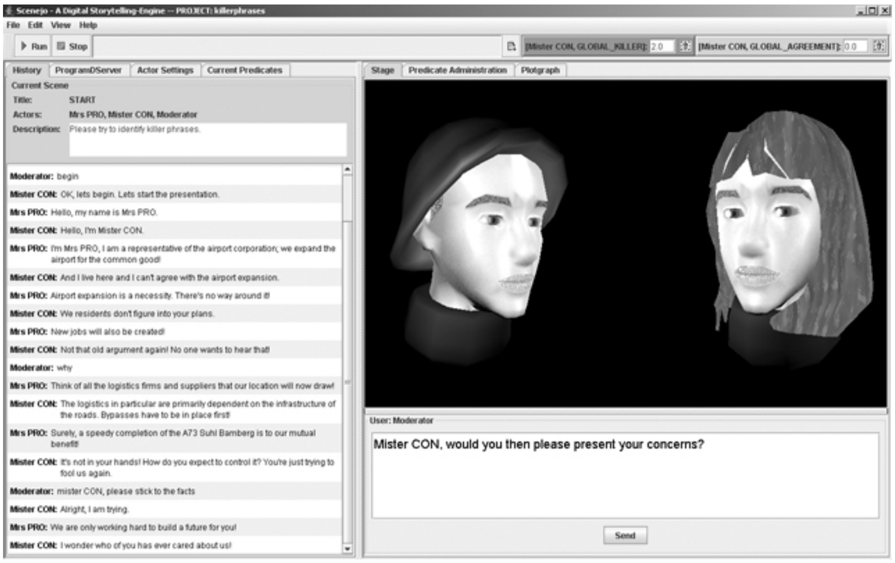
4 The Creation Process of the Killer Phrase Game

The “Killer Phrase Game” is a digital conversational game where the player is the moderator of a discussion between two debating characters. The application is running with the platform *Scenejo* [21]. *Scenejo* connects several A.L.I.C.E. chatbots [1] in a conversational loop of turn-taking, which is controlled by a drama manager component. On top of the creation possibilities of AIML[2], the mark-up language for

[2] AIML: Artificial Intelligence Markup Language. The knowledge base for the chatbot A.L.I.C.E. [1] is implemented in AIML.

“explicitly” defining dialogue, the authoring tool provides condition definitions for utterances, processing and affecting each bot’s inner states separately. The virtual characters attained in this manner are animated heads talking with TTS, while their textual conversation is also displayed on screen. The user is able to interrupt the conversation by typing text. As such, the interaction concept of *Scenejo* strongly resembles that of Façade [11]. In fact, any new content created would result in a similar interaction style – a user takes turns in a conversation with several digital characters.

The killer phrase game has been designed as an experiment for a University seminar on “moderation and mediation”.[3] The game tackles the topic of how to identify and react to so-called “killer phrases” within a discussion. Killer phrases are “creativity killers”, often used in a knee-jerk manner, that destroy new ideas before they are discussed with solid arguments. The game assumes a scenario with two parties, the planners and the residents, arguing about novel plans for an airport expansion. The partly predefined conversation between the two parties, carried out across the table, contains killer phrases. The learner plays the role of the moderator and has to manage the meeting (see game screen in Figure 3).



[Mister CON, GLOBAL_KILLER]: 2.0 [Mister CON, GLOBAL_AGREEMENT]: 0.0

Current Scene
Title: START
Actors: Mrs PRO, Mister CON, Moderator
Description: Please try to identify killer phrases.

Moderator: begin
Mister CON: OK, lets begin. Lets start the presentation.
Mrs PRO: Hello, my name is Mrs PRO.
Mister CON: Hello, I'm Mister CON.
Mrs PRO: I'm Mrs PRO, I am a representative of the airport corporation; we expand the airport for the common good!
Mister CON: And I live here and I can't agree with the airport expansion.
Mrs PRO: Airport expansion is a necessity. There's no way around it!
Mister CON: We residents don't figure into your plans.
Mrs PRO: New jobs will also be created!
Mister CON: Not that old argument again! No one wants to hear that!
Moderator: why
Mrs PRO: Think of all the logistics firms and suppliers that our location will now draw!
Mister CON: The logistics in particular are primarily dependent on the infrastructure of the roads. Bypasses have to be in place first!
Mrs PRO: Surely, a speedy completion of the A73 Suhl Bamberg is to our mutual benefit!
Mister CON: It's not in your hands! How do you expect to control it? You're just trying to fool us again.
Moderator: mister CON, please stick to the facts
Mister CON: Alright, I am trying.
Mrs PRO: We are only working hard to build a future for you!
Mister CON: I wonder who of you has ever cared about us!

User: Moderator
Mister CON, would you then please present your concerns?

Fig. 3. The Killer Phrase Game prototype running in the *Scenejo* Platform.

Although the application is called a “game”, there are also elements of narrative expression and of simulation. The design tasks included: 1.) generation of narratives containing characters, actions (performed as utterances), changing states and turning points for possible endings, 2.) a simulation model made up of states and transitions, and finally 3.) the game design defining local objectives for the player, and tuning / testing the gameplay.

[3] More detailed insights on the educational achievements of the concept have been published in [18] and [19].

4.1 Building a Story and a Model: Experiences

The design conditions for the game were affected by its educational purpose. Rather than by artistic motivation, the work was done in iterative brainstorming sessions by an interdisciplinary team, consisting of the instructors of the course “moderation and mediation”, and of the designers and programmers of the platform *Scenejo*. While the *Scenejo* designers provided the logical frame for a computable conversation, which was a concept of actions (utterances), states and rules, the domain experts were supposed to provide its content – decisions on concrete conversational states and defined rules calculating the story’s transitions and turning points. At the beginning of this process, it turned out that it was much easier for the domain experts to imagine concrete conversations than to explain them as an abstract model. So, the easier start was to provide concrete cases of potential conversations in the form of a linear script of explicit dialogue lines – particularly at first, stories with a “best case” and a “worst case” progression and ending.

In order to achieve interactivity in the sense of a gaming simulation, the necessary next step was the modelling of dependencies between utterances, according to the principle sketched in Figure 2 (states and transitions defining a “behaviour” of the world). This transfer is not a straightforward task. The easier part was to analyse the scripts and categorise the single utterances, for example, into killer phrases and groups of arguments. Building a simulation model, however, required the connection of these groups of utterances to world states that can change over time. Identifying world states at “the right level of detail” required the knowledge of the domain experts, given that the model was supposed to be simple for the first implementation.

The first achieved model (see Figure 4) made use of the character-centered approach of *Scenejo*. Each virtual actor is modelled with its/his own mind states. During play, these individual states will be affected by events (for example, by actions of users or other virtual actors). For example, offending one character would count down the value of his cooperativeness. The moderator may have to act to raise this value again, since only with a high level of cooperativeness, will this character finally agree on a compromise.

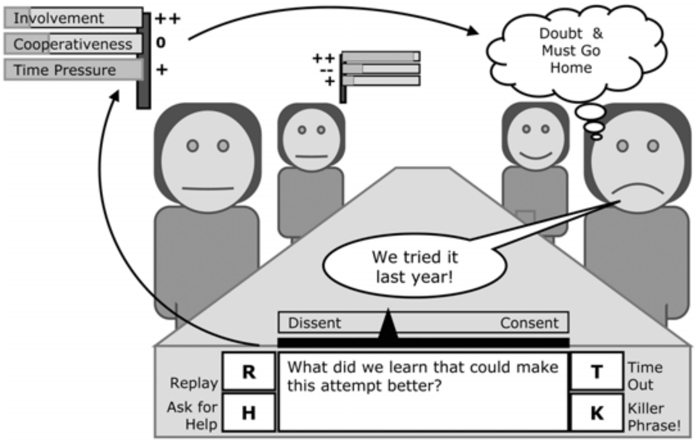


Fig. 4. Design sketch of the conceived structural elements of the “killer phrase” game, such as the parameter states of each virtual actor

This is an abstract model, tailored to a certain game goal, yet without personality traits for each actor. Modelled traits would affect the transition rules in a way that lets different characters react differently to the same actions and events. This would be useful to enable apparent novelty and idiosyncrasies in the conversational turns each time the game would be played, and will be considered as future work. As a drawback in terms of the educational purpose, we expected that – next to an immense increase in complexity – the correlations that can be made between actions and outcome would be disguised. Instead, for the first prototype, we were looking for reduced intricacy, also in terms of getting a grip on the authoring process, which was conceived at the same time. Finally, the next hitch was the proper assignment of the model's states with concrete bot utterances and potential user utterances.

As a result, the initial model had been reduced again to an even more abstract and simple game-like model. Figure 5 shows the simplified model, which was then implemented. Instead of modelling each character's mind, a generalized view is taken by only modeling overall levels of stress – here: the "Killer Phrase Level". This model places a scale on the overall "mood", with the effect that valid arguments of each party can only be played out by the system if the killer phrase level is low.

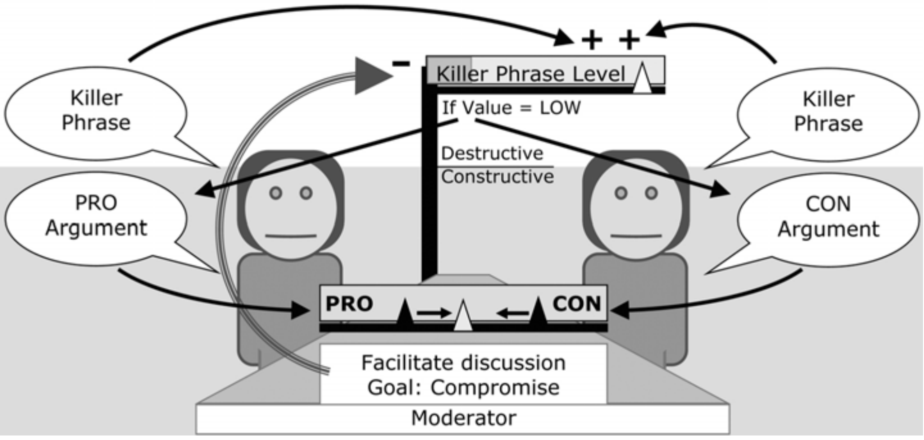


Fig. 5. Simplified model of states and transitions.

Thus, it can be stated that the "chosen level of detail" (in Holland's terms, compare section 3) has led to a result that makes a point as a simple gaming simulation and as an interactive short story, at least creating awareness of the phenomenon of a "killer phrase". Compared to "reality", the implemented model appears as an oversimplification, far from Holland's "perfect model". Phrasing it in narrative terms, a model of a simple "storyworld" has been created by people who wanted to make a point. This model is part of the content and, at first, independent of potentially underlying engine models resembling physical realities (such as psychological models).

These preparations, outside of the authoring tool, showed the indirect work of turning an explicitly scripted dialogue into a model that "implies" how the conversation can go if not scripted in detail. This major step has to be done by story artists or by simulation designers, who are completely involved in the content matter, and not by the engineers of the runtime platform. For example, as a common practice in modelling gaming simulations for organisational change, the “Critical Incident” technique is used to turn oral stories into a model, by segregating essential dependencies.

4.2 Affordances for the Authoring of Actions, Events and States

Transforming the conceived model into a runtime application requires methods to define procedures generating actions and events to influence states. The way this can be accomplished depends on the platform available. In the case of the Killer Phrase game, the *Scenejo* authoring tool provided an interface for non-programmers. It works in accordance with its unique platform peculiarities, which construe actions as utterances of individual characters (bots), finally to be coded in AIML.[4]

The actions (utterances) of each bot have to be constructed with pre-conditions and post-conditions. A pre-condition consists of 1.) a text pattern of another agent’s utterance (bot or user) and optional 2.) state conditions required for the action to be triggered. A post-condition is 1.) an optional state change, and 2.) the spoken utterance. A so-called “Stimulus-Response” element manages this procedure in *Scenejo*. The killer phrase game contains threads of dialogue between two bots, which can be interrupted by the user. To code these dialogues in the above way, it is necessary to match the post-conditions of one bot with the pre-conditions of the other, and vice versa. The “Stimulus-Response” interface also allows the author to define utterances not only in direct speech, but also in an abstract form, to denote a dialogue act. For example, a dialogue act can be named “Reproach Ignorance Killerphrase”, connecting to a selection of possible concrete utterances, such as “We residents don't figure into your plans” or “We're only asked our opinion after everything is already set in stone”.

On a higher level of abstraction, dialogue graphs can be structured into scenes. Scenes are always connected by an explicit connection; at this level, no emergence is at work, but only branching paths as the result of certain choice events.

4.3 Conclusions for Explicit Authoring and Implicit Creation

The creation process of the Killer Phrase game on the *Scenejo* platform showed limitations and possibilities of explicit writing methods, as well as of implicit techniques.

In terms of accessibility for newcomers to the modelling process, it was easiest to start with explicitly phrased text in a linear order on several task levels. For turning the dialogue script into something more procedural – letting a similar dialogue “emerge” from interaction, each time with variations – it was necessary to define rule-based models. Such “implicit creation” has been achieved for the following aspects:

- The game state varies based on user interaction and on some randomisation in the bot’s action selection, depending mostly on the development dynamics of the “Killer Phrase” state and “Agreement” level state in the designed model.
- Interruptions within the dialogue threads are possible at any given moment, since the utterances are modelled as conditional actions.
- Abstract dialogue acts can result in variations of concrete utterance templates.

[4] The main *Scenejo* authoring tool components have been described in [12].

The following aspects have still been authored explicitly:

- The utterances are hand-crafted. Therefore, it can't be excluded that exact repetitions in the spoken text occur. This can be diminished by creating a huge body of variations explicitly. Turning to complete "implicit creation", i.e., generating the language, requires modelling outreaching current possibilities – ontologies of the theme (airport expansion), of discussion styles, and more.
- The order of actions within a dialogue thread is predefined, unless a user interrupts. However, the order of all available dialogue threads varies according to the game state model. "Implicit creation" would have to result in a conversational model of the argument.
- The game contains a small higher-level plot structure, providing a predefined branching point to win or lose the game (compromise or escalation). More sophisticated models would contain planning structures and goals for the actors to be tuned on a higher level.

Although only few aspects of this application were model-based instead of explicitly scripted, it showed emergent performance, in the sense that it turned out to be difficult to anticipate the runtime behaviour completely during the authoring phase. Possible utterances were explicitly formed, but not their order, and only some of their connections. From a perceptual point of view, there might be no difference between the appearance of a successfully emergent aspect and a failure in authoring. The consequences are that the phase of tuning and testing gets intricate, and that "debugging" tools for dynamic content are needed, in the sense of inversely tracking down responsible rules for an occurring effect, in order to fine-tune them. Thus, the process of creation becomes similar to programming complex software, even if no programming language, but rather visual editors are used. The *Scenejo* tools have proven to be accessible and effective. However, the authoring process on the action level, including the definitions of pre- and post-conditions, is nonetheless a tedious task close to programming.

5 Steps towards Implicit Creation for Authors

In this section, the concept of emergence is contrasted with the notions of storytelling and creation. The phenomenon of patterns created by an ongoing emergent process is usually seen as the opposite of anything based on intentions, such as creation. However, there is a creative, inductive process of finding rules that attempt to model patterns of interest – a selection. J. Holland, 1998: *"Emergence must somehow be bound up in the selection of the rules (mechanics) that specify the model, be it game or physical science. […] Knowing what details to ignore is not a matter of derivation or deduction; it is a matter of experience and discipline, as in any artistic or creative endeavor."* [7, p. 113]

There is a correlation between the above concept and writing a novel or other forms of "traditional" storytelling. D. Herman, who provided an integrating view on narrative theories including other disciplines [6], defined the notion of a "storyworld" as an "ecology of narrative interpretation", from the perspective of recipients. Actions, events and states are parts of the inventory of "local principles of storyworld design” [6, p. 27]. Presented actions in a story are mostly incomplete and “underspecifying” the storyworld – much of it is only implied, without being explicitly articulated during narration. For example, one pithy dialogue line in a film can have three functions [17]: 1.) provide information, 2.) display emotions, and 3.) reveal traits by its diction. The art of storytelling and the art of model building both rely on omitting details. Nevertheless, they are not the same and have to be distinguished. Figure 6 illustrates the dual meaning of a storyworld.

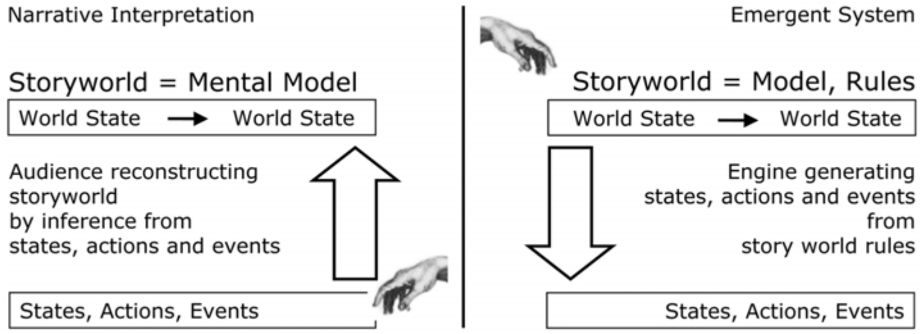

Narrative Interpretation
Emergent System
Storyworld = Mental Model
World State
World State
Audience reconstructing storyworld by inference from states, actions and events
States, Actions, Events
Storyworld = Model, Rules
World State
World State
Engine generating states, actions and events from story world rules
States, Actions, Events

Fig. 6. Two meanings of “storyworld”: Left, the recipient’s mental model, built from interpreting created states, actions and events (according to [6]). Right, the designer’s created dynamic model, leading to generated states, actions and events (after [7]).

“Implicit creation” in IDS is the content creator’s task of letting actions, events and states result from automatic generation during narration, by specifying a dynamic model (a storyworld). Various engines with underlying dynamic models for the generation of perpetual variations of actions already exist: scientific models from physics and psychology, such as for gravity, vision, kinematics, emotions, cognition and linguistics. However, actions based on models of “reality” alone do not tell a story or provide a storyworld. The concern with developing unique storyworlds as a basis for coherent actions is part of the content creation, not of engineering. Faced with the complexity of emergent systems, content creators need to approach implicit creation in steps, starting with explicit creation methods for its greater accessibility. There is a need for future research in identifying appropriate steps and developing supporting tools.

6 Conclusion

In this contribution, the notion of “implicit creation” has been introduced. It paraphrases concepts and metaphors (such as the gardening metaphor) for authors who attend to the building of emergent narrative, even when it is not necessary for all possible narrative structural aspects to be rendered as emergent properties. The difference with explicit creation is that the created content does not fully describe the resulting actions, states and events in every detail, but it “implies” them. The current discussion on authoring systems mainly focuses on GUI tools that free people from programming, in the sense of replacing the code generation form of typing by clicking. While this is commendable, it is not the only problem for authors. Instead, conceptual models for implicit creation still have to be created and communicated.

References

1. ALICE. Homepage of the A.L.I.C.E. Artificial Intelligence Foundation. Online: http://www.alicebot.org (Last accessed: 08.31.2007)
2. Aylett, R.: Narrative in Virtual Environments - Towards Emergent Narrative. Proceedings, AAAI Fall Symposium on Narrative Intelligence, TR FS-99-01, AAAI Press (1999)
3. Aylett, R., Louchart, S., Dias, J., Paiva, A. and Vala, M.: Fearnot! - an experiment in emergent narrative. Proceedings IVA 2005, Springer LNAI 3661 (2005) 305-316
4. Charles, F., Lozano, M., Mead, S.J., Bisquerra, A.F. and Cavazza, M.: Planning Formalisms and Authoring in Interactive Storytelling. In: Proceedings of TIDSE 2003, Darmstadt (2003)
5. Crawford, C.: Storytron Interactive Storytelling. Project Website: http://www.storytron.com/ (Last accessed: 08.31.2007)
6. Herman, D.: Story Logic: Problems and Possibilities of Narrative. University of Nebraska Press, Lincoln (2002)
7. Holland, J.H.: Emergence: From Chaos to Order. Oxford University Press, Oxford (1998)
8. Inscape Consortium: Inscape Storytelling EU Project Website, Research Papers, http://www.inscapers.com/downloads_research_papers.html (Last accessed: 08.31.2007)
9. Iurgel, I.: Cyranus – An Authoring Tool for Interactive Edutainment Applications. In: Proceedings of Edutainment 2006, Int.Conf. on E-learning and Games, Zhejiang (2006)
10. Louchart, S., Aylett, R.: Solving the narrative paradox in VEs - lessons from RPGs In: Proceedings of Intelligent Virtual Agents 2003, Springer LNAI (2003) 244-248
11. Mateas, M., Stern, A.: Procedural Authorship: A Case-Study Of the Interactive Drama Façade. In: Proceedings of Digital Arts and Culture (DAC), Copenhagen (2005)
12. Mateas, M.: Expressive AI – A hybrid art and science practice. Leonardo: Journal of the International Society for Arts, Sciences, and Technology 34 (2), (2001) 147-153
13. Medler, B., Magerko, B.: Scribe: A Tool for Authoring Event Driven Interactive Drama. In: Proceedings of TIDSE 2006, Darmstadt, Springer LNCS (2006)
14. nm2 Consortium: nm2 – New Millenium, New Media. EU Project Website. http://www.ist-nm2.org/publications/deliverables/deliverables.html (Last accessed: 08.31.2007).
15. Pearce, C.: Sims, BattleBots, Cellular Automata God and Go. A Conversation with Will Wright. Game Studies Journal Vol. 2 (1), http://www.gamestudies.org/0102/ (2002)
16. Sauer, S., Osswald, K., Wielemans, X., Stifter, M.: U-Create: Creative Authoring Tools for Edutainment Applications. In: Proceedings of TIDSE 2006, Springer LNCS (2006)
17. Schütte, O.: 'Schau mir in die Augen Kleines'. Die Kunst, gute Dialoge zu schreiben. (in German.) Verlag Lübbe (2002)
18. Spierling, U.: Learning with Digital Agents – Integration of Simulations, Games, and Storytelling. In: Burmester, Gerhard, Thissen (eds.): Proceedings of Digital Game Based Learning – Symposium Information Design 2005. Universitätsverlag Karlsruhe (2006) 115-147
19. Spierling, U.: “Killer Phrases”: Design steps for a game with digital role playing agents. In: Proceedings of ISAGA 2007, 38th ISAGA Conference, Nijmegen, The Netherlands (2007)
20. Spierling, U., Iurgel, I.: Pre-Conference Demo Workshop “Little Red Cap”: The Authoring Process in Interactive Storytelling. In: Proceedings of TIDSE 2006, Springer LNCS (2006)
21. Spierling, U., Weiß, S. and Müller, W.: Towards Accessible Authoring Tools for Interactive Storytelling. In: Goebel, Malkewitz, Iurgel (eds.): Technologies for Interactive Digital Storytelling and Entertainment 2006, Conference Proceedings, Springer LNCS (2006)