On the Fly Collaborative Story-Telling: Revising Contributions to Match a Shared Partial Story Line

Pablo Gervás
Universidad Complutense de Madrid
28040 Madrid, Spain
pgervas@sip.ucm.es

Rafael Pérez y Pérez **Ricardo Sosa** **Christian Lemaitre**
Universidad Autónoma Metropolitana, (Cuajimalpa)
México D.F., México
rperez@correo.cua.uam.mx
rsosa@correo.cua.uam.mx
clemaitre@correo.cua.uam.mx

Abstract

Computational improvisation is a challenging topic. It involves collaborative creativity, the modelling of interesting cognitive process like those needed to keep the coherence and interestingness of an emergent story, the ability to foresee possible interesting directions that the improvisation might take, etc. In this paper we present an architecture for story-telling improvisation. It is based on the engagement-reflection computer model for plot generation. It involves the interplay of two agents in order to generate a novel, coherent and interesting story. Our purpose is to provide an analysis of the key requirements to develop a computational improviser and the solutions we envisage to achieve this goal.

Keywords: Colaborative storytelling, improvisation, engagement-reflection.

1 Introduction

Improvisation can be defined as the act of creation of a work or its performance in real time, individually or by a sequence of contributions by a number of interacting agents. Improvisation is known to be a type of collaborative creative activity that takes place extemporaneously with continuously updating preparation but without prior planning. In improvisation, a combination of planned and unplanned actions takes place. The contributions of each improvising actor often follow three basic restrictions: they must be consistent with the contributions of other players, they are expected to result in an interesting plot that emerges from their interaction with the rest of the players, and they must be produced by avoiding noticeable gaps in the run of the scene.

From the viewpoint of computational creativity, this set up presents a number of interesting questions. On one hand, the restriction on the quality of the emergent material suggests that some kind of shared intentionality may be required to drive the production of each actor's contribution. Actors may be searching for particular effects when they produce certain contributions. To a certain extent this may be modeled as some kind of preparation[1] activity, during which the actor contemplates possible effects of his immediate actions and produces his contribution based on that preparation.

On the other hand, the restriction on overall consistency of the set of contributions implies that actors must continuously consider the contributions of other players. A contribution prepared by one player may need to be revised, altered or even scratched altogether if another player generates conflicting material before that contribution is actually executed. The fact that interaction takes place in sequences without gaps complicates matters further, since it precludes the elementary solution of waiting until everybody else's contribution has finished before starting to prepare one's own.

This problem is worth studying both from the point of view of understanding how humans address it and from the point of view of devising computational methods for emulating this behaviour in particular situations. However, theatrical improvisation involves too many complex levels of interaction to be modelled successfully in computational terms: a text must emerge from the interaction, but other ingredients such as diction, gesture, body language... play too crucial role to be dispensed with without compromising the validity of the analysis.

A possible solution is to try to find a simpler problem that retains the fundamental issues concerning preparation, revision, emergent quality and real-time interaction, but has a lower complexity of the material to be considered. In this paper we put forward a model for on-the-fly collaborative storytelling that may satisfy these criteria. Two story tellers take turns in advancing a shared story line. While one contributes, the other one listens. Because he will be expected to take over as soon as the speaker stops, he cannot postpone the task of preparation until the speaker has finished. So he prepares ahead a ten-

[1]Preparation defined as "to be prepared: to be in a state of readiness, ready; to be mentally ready, inclined, disposed; to be in a condition or position to do something" (Oxford English Dictionary, www.oed.com) is used in a more flexible way than the more definite nature of planning used in the traditional AI literature.

tative sketch of his own contribution. Any conflicts arising with the other story teller's contribution as it emerges will force him to revise his prepared contribution.

We believe this model to contain all the elementary details that puzzle us in theatrical improvisation, while being restricted to the somewhat simpler task of story generation, for which a number of computational solutions already exist. In this paper we propose an architecture for engaging two such existing computational programs for story telling - in truth, two copies of the same solution, possibly running under different configurations - in an exercise of on-the-fly collaborative storytelling. We discuss the restrictions that the specification imposes on the story tellers, and we consider how their collaborative creative tasks can be modelled.

Improvisation has been part of theatre performance for centuries; it is well documented that performers of the *Commedia dell'arte* were excellent improvisers[2]. During the first half of the twentieth century in Chicago, Viola Spolin introduced the theatre games based on improvisation; since then, the number of improvisation practices has increased. We are interested in furthering our understanding of the relation between improvisation and creativity within computational settings. Some improvisation issues that we identify as key elements for this research include:

- Criteria must be established to evaluate the quality of improvisations and the means to validate this measure.

- Coherence must be maintained during an improvisation. This is an important challenge since actors cannot modify what has already been told.

- Actors have different knowledge and experiences and therefore different ways of interpreting the world.

2 Previous Work

The work described in this paper involves story telling, improvisation, and interaction between programs that can roughly be classed as agents in the sense that they communicate and exchange data in a colaborative effort to produce a common result. For each of these elements, a brief summary of the relevant background is provided in this section.

2.1 Automatic Story Telling

Of the various approaches to automated story telling described by Bailey (1999), those based on modelling the processes that a human author follows in generating a story are most interesting from the point of view of modeling creative endeavour from a computational point of view. The work of Turner (1994) on the MINSTREL system was pioneering in the sense that earlier attempts at automated storytelling - such as the work of Meehan (1977) or Rumelhart (1975) - focused more on modeling the world about which stories are told or the actual form of the story as a linguistic artefact, respectively. Although several research efforts have addressed storytelling in different ways since then, it is not until the work of Pérez y Pérez and Sharples (2001) on the MEXICA system that modelling the actual processes of creative story composition has been addressed especifically.

The work presented in this paper presents an extension of a similar analysis to the modifications to the creative process induced by a colaborative improvisational setting.

2.2 Improvisation as Collaborative Story-Telling

The on-line Webster dictionary defines improvisation as "an unplanned expedient" or "a performance given without planning or preparation". It is clear that improvisation is something different from planning. However, Moraes and da Rocha Costa (2002) claim that planning can be understood as improvisation under external constraints. In their model, there is a "director" who is responsible for providing the actors with a full script of the story to represent; so, the actors' job consists in finding ways of reaching the goals imposed by the script. This has been, in fact, a research line of hierarchical planning where planning and acting can be interleaved. By contrast, we consider improvisation a collective activity where the plot (or script) emerges as result of the interaction between agents.

Philip Agre - in Agre and Chapman (1987); Agre (1997) - has characterized computational improvisation as the continual dependence of an agent's action upon its circumstances. This proposal interprets improvisation as a running argument that an agent continually updates among various alternatives. These options together form a dynamic argument structure which undergoes constant change as a result of agent activity and its impact on the world, including other agents. Under this approach, the emerging behavior is an considered as epiphenomenon of the interactions between agents and their world. There are a few basic constraints mentioned in the literature of theatre performance that a good improvisation must fulfill. Following Trastoy (2005), some of these constraints are listed below.

Improvisation is a sort of story-telling and the whole performance must have some basic structure such as introduction to the problem, development and resolution.

Nothing should be agreed in advance within the group of actors. The public might suggest a topic to be developed during improvisation and actors can take some seconds to define basic issues about roles and other matters, but nothing about the plan of the story.

The dynamic of the story is driven by conflicts. The story is in fact a collective search for solutions to those conflicts. During the unravelling of the plot new conflicts might arise. The challenge for actors is to keep the coherence and interestingness of a dynamic and unpredictable story, finishing with a good synthesis of the different problems.

The role of the director of an improvisation troupe is quite different from the role of the director in traditional theatre. The former is responsible for the general setting of the performance while the latter controls every aspect of the play.

Thus, an improvisation performance is a sort of collec-

[2]This article was originally published in Bellinger (1927)

tive story-telling game where the golden rule is that no actor must block the story initiated by its predecessor. That is, he can never say something like "No, what he said is not true, the truth is..." Besides that rule, actors are free to generate the next segment of the story as they want.

Our approach to computer improvisation as story-telling incorporates some of the main characteristics of the improvisation troupes. We envision a set of agents who may play specific roles in the story. The role of the director is played by the programmer who defines some basic features of the story: the length of the story, the maximum length of each "intervention" of the agents, the number of characters, which character is assigned to which agent, the agent who will start the performance, etc.

2.3 Agent Architectures

The Open Agent Architecture (OAA) Cheyer and Martin (2001) is a framework for developing multi-agent systems intended to enable more flexible interactions among a dynamic community of heterogeneous software agents. The operation of the architecture is based on the idea of delegation: agents do not hard-code their interactions (method calls or messages) in a way that fixes how and whom they will interact with, instead the interactions between OAA agents are expressed in terms of needs delegated to a Facilitator agent. This Facilitator agent coordinates the agent community so that it can achieve its task. It does this by providing services such as parallelism, failure handling, and conflict detection, which relieves each client agent from having to worry about these issues itself. OAA's Distributed Agents are simply programs - or wrappers around programs - that share several common functionalities, and which are possibly distributed across various machines.

3 The MEXICA Story Telling System

MEXICA is a computer model of creativity in writing that develops frameworks for short stories. It is inspired by the engagement-reflection account of writing given in Sharples (1999). In MEXICA a story is defined as a sequence of actions. Each action has associated a set of preconditions and post conditions, defined by the user of the system, which are comprised by emotional links and tensions between characters. Emotional links are represented as a continuum between hate and love with discrete values ranging from -3 to +3. In this way, the precondition of the action *Hunter killed Jaguar Knight* might be that the hunter hates the knight (an emotional link of intensity -3; see first line of Table 1); the post condition of the action *Princess decorated Eagle Knight* might be that the knight is very grateful towards the princess (an emotional link of intensity +2; see second line of Table 1). In MEXICA, the tension in the story increases when a character is murdered, when the life of a character is at risk, when the health of a character is at risk (i.e. when a character is hurt or ill), or when a character is made a prisoner. Like emotional links, tensions can be employed as preconditions or post conditions. Actions also might include post conditions that deactivate tensions. In this way, the action *Princess healed Jaguar Knight* has as a precondition the fact that the knight must be injured or ill (a tension due to health at risk) and as a post condition the fact that the knight has been cured (the tension is deactivated) and that the knight is very grateful towards the princess (an emotional link of intensity +2) (see third line of Table 1). Finally, MEXICA includes inferred tensions, i.e. tensions that are activated when the system detects that: 1) two different characters are in love with a third one (tension due to love competition); 2) when a character has two opposite emotions towards other one (tension due to clashing emotions); 3) and when a character hates other character and both are located in the same position (tension due to potential danger). If the conditions that activate an inferred tension disappear, the tension is deactivated. Each active tension has associated a value that the system records each time an action is performed. In this way, the system represents as a graph the value of the tension in the tale over story-time. A story is considered interesting when it includes increments and decrements of the story-tension, e.g. if a princess is kidnapped (an increment in the tension) and then rescued (a decrement of the tension). All actions' post conditions are recorded in a structure known as the story-context. So, the context represents the state of affairs in the story in progress. MEXICA has two core processes: the creation of knowledge structures in memory and the plot generation.

3.1 Construction of Knowledge Structures

MEXICA builds its knowledge structures from a set of narratives known as *previous stories*. Previous stories are provided by the user of the system and they are composed of sequences of actions. So, previous story 1 is formed by action 1, action 2, action 3, and so on. For the sake of a clearer explanation, we first describe how story-contexts are updated when MEXICA processes the previous stories and then we elaborate the explanation to clarify how knowledge structures are created. The process of updating story-contexts work as follows: 1) MEXICA takes the first action in the first previous story, triggers its post conditions and updates the story-context; 2) MEXICA takes the second action in the first previous story, triggers its post conditions and updates the story-context; and so on. In this way, each time an action is performed the story-context is updated. So, we can refer to the story-context after action 1 is performed as context 1, to the story-context after action 2 is performed as context 2, to the story-context after action 3 is performed as context 3, and so on. Or we can say that action 1 generates context 1, action 2 generates context 2, action 3 generates context 3, and so on. Notice that context 2 is not necessarily made up by the addition of the post conditions of actions 1 and 2. As mentioned earlier, some action's post conditions might deactivate tensions between characters, and inferred post conditions might become active or inactive at any moment. Thus, the story-context is a very dynamic structure that progresses over story time.

Thus, the process to build knowledge structures works as follows:

1. MEXICA takes the first action in the first previous story, triggers its post conditions and updates the

Precondition	*Action*	*Postcondition*
The hunter hates the knight (an emotional link of intensity -3)	**Hunter killed Jaguar Knight**	
	Princess decorated Eagle Knight	The knight is very grateful towards the Princess (an emotional link of intensity +2)
The knight must be injured or ill (a tension due to health at risk)	**Princess healed Jaguar Knight**	The knight has been cured (and therefore the tension has been deactivated) The knight is very grateful towards the Princess (an emotional link of intensity +2)

Table 1: Three actions with their pre and post conditions (defined by the user of the system).

story context creating context 1. Then, it copies context 1 into a new structure created in memory known as atom 1. Next, it copies the following action in the previous story - in this case action 2 - into atom 1. In this way, atom 1 is linked to action 2.

2. MEXICA takes the second action in the first previous story, triggers its post conditions and generates context 2. Then, it copies context 2 into a new memory structure known as atom 2. Next, it copies action 3 into atom 2. So, atom 2 is linked to action 3.

3. If atoms 1 and 2 are alike, the system copies the action linked to atom 2 into atom 1 and destroys atom 2. So, atom 1 is linked to action 2 and action 3.

The following lines exemplifies this process. Imagine that the first previous story includes the following sequence: *Farmer wounded Jaguar Knight*; *Princess cured Jaguar Knight*; *Jaguar Knight murdered Farmer*; *The End* (see Figure 1). The first action, where the knight is wounded, generates context 1 which is comprised by the tension *Jaguar knight's life is at risk* and the emotional link *Jaguar Knight hates Farmer*. MEXICA copies context 1 into memory, creates atom 1 and links the following action in the sequence (in this case *Princess cured Jaguar Knight*) to atom 1 (see case *a* in Figure 1). Notice that, within atoms, characters are substituted by variables. In this way, atom 1 represents the knowledge that when the life of a character X is at risk (where character X is any character) and character X hates character Y (where character Y is any character but X) a logical way to continue a story is that a third character Z heals character X. This information will be essential during plot generation. Next, the system triggers the post conditions of the second action in the story, i.e. when the knight is healed. So, context 2 is created; it is comprised by the emotional link *Jaguar knight is very grateful towards the Princess* and a second emotional link *Jaguar Knight hates enemy*. Notice that the tension Jaguar knight's life is at risk is deactivated as a result of the princess curing the knight. So, it disappears from the context. Context 2 is copied into memory to create atom 2 and action 3 is linked to such an atom (see case *b* in Figure 1). MEXICA takes action 3 and triggers it post conditions; however, because this is the last action in the story the process stops. The same process is repeated for each previous story. At the end, if the system is provided with enough stories, each atom in memory might have several linked actions. Each atom in memory represents a possible state of affairs in the story world in terms of emotional links and tensions between characters. Linked actions provide different routes that a narrative can follow during story generation given a specific story-context.

3.2 Plot Generation

There are two core processes that interact during plot generation: engagement and reflection (see Figure 2). During engagement the system produces sequences of actions as follows: an initial action is selected; MEXICA triggers all its post conditions updating the story-context; the system employs the story-context as cue to probe memory and tries to match an atom that is equal or similar to it; the system retrieves all the actions linked to the matched atom and selects one at random as the next action in the story; the system updates the story-context and the engagement cycle starts again. If the system cannot match any atom in memory an impasse is declared. By default the cycle repeats until three actions are generated or an impasse is declared. Then, the system switches to reflection.

During reflection the system:

1. Verifies that the preconditions of all actions generated during engagement are satisfied (notice that preconditions are ignored during engagement). If necessary, the system inserts actions in the story produced so far to satisfy preconditions.

2. Evaluates the interestingness and novelty of the story in progress. A story is interested when it includes increments and decrements of tension (e.g. if the princess is kidnapped and then rescued); a story is novel when it is not similar to any of the previous stories (MEXICA compares sequences of actions between the story in progress and all the previous stories).

3. Breaks impasses.

Then, the system switches back to engagement and the cycle continues. The interaction between engagement and reflection generates MEXICA's output. Atoms are knowledge structures comprised by emotional links and tensions. They are general enough to enclose different alternatives to progress a story, but at the same time they are specific enough to drive in a coherent way the development of a tale. So, a narrative can be expressed in terms of clusters of emotional links and tensions between characters that progress over story time. We exploit this characteristic to propose an architecture for improvisation.

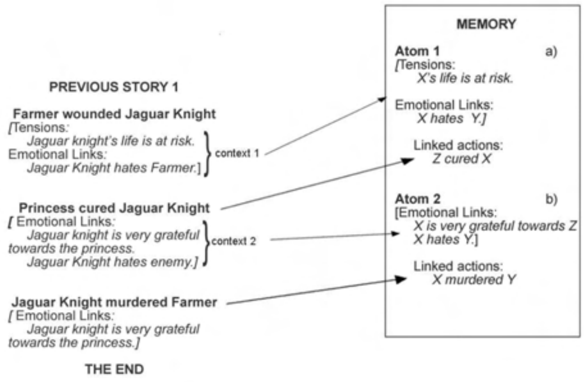

Figure 1: How atoms are created in memory: a) illustrates atom 1 comprised by one emotional link and one tension and linked to the action Z cured X; b) shows atom 2 comprised by two emotional links and linked to the action X murdered Y. Z, X and Y represent variables. Atom 1 is built from context 1 and atom 2 from context 2.

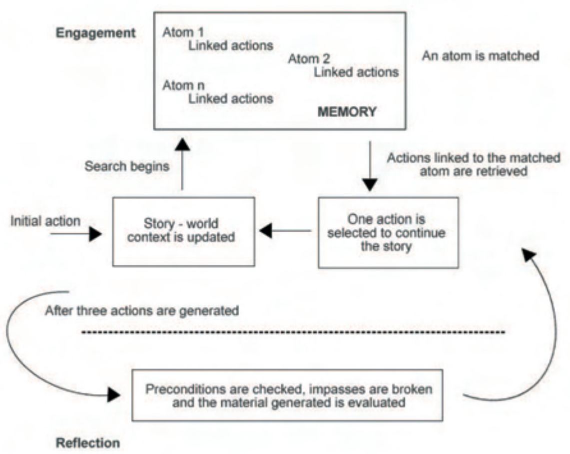

Figure 2: The engagement-reflection cycle

4 An Architecture for On the Fly Collaborative Storytelling

Two aspects determine how colaboration takes place between story telling programs: how each program addresses the task of creating stories in this way, and how the colaboration between the story tellers is orchestrated. We are assuming that from the point of view of creativity, the first aspect is fundamental, whereas the second aspect concerns a tecnical issue of interconnecting two systems. An ideal solution to this second problem should be independent of the actual storytelling processes employed by each participant.

4.1 The Participating Storytellers

We employ two agents: MEXICA 1 (M1) and MEXICA 2 (M2). The basic process of our system will work as follows:

- the user provides an initial action (action 1).
- M1 and M2 create their own story-context (so, we have context 1 of M1 and context 1 of M2).
- M1 generates one action to continue the story (action 2), updates its story-context (creating context 2 of M1) and communicates to M2 the action 2.
- M2 receives action 2, updates its own story-context (creating context 2 of M2) and generates a new action (action 3) to continue the story.
- M2 updates its own story-context (creating context 3 of M2) and communicates to M1 the new action (action 3) in the improvisation.
- M1 updates its story-context (creating context 3 of M1), generates a new action (action 4), and so on.

In such a setting each one agent would only start preparing its actions once the other one had finished his contribution. The improvisation problem would become equivalent to the two story-tellers taking turns in extending the story. Several modifications to the basic process can be applied to enrich the simulation. The agents involved in the basic model basically act in the role of authors, because there is no difference between the way they respond to a contribution of the other agent and the way they respond to a contribution of their own. This can be changed by extending the number of actions that each agent can contribute in his turn. If an agent can produce more than one action without passing the turn to the other agent, this forces the passive agent to model a new attitude: that of a listener of the story. At any point during an improvisation, one agent would be operating in speaker mode, and the other one in listener mode. While in speaker mode, an agent generates actions and communicates them to other participants in real-time. While in listener mode, an agent may generate actions, but it does not communicate them. Instead, it puts them in a store of tentative contributions to wait until its turn comes to operate in speaker mode. This store of contributions must be revised whenever a new contribution is received from a speaker.

An agent acting in *speaker mode* would operate as follows:

- At the start of its turn, it communicates to other participants all those of its stored tentative contributions that were not in conflict with those communicated during previous turns, and which have not already been contributed by other agents.
- Then it carries on generating new actions. At each stage of the turn, it generates one action to continue the story, updates its story-context and communicates to the other participants the new action.
- It carries on in this way until his turn finishes.
- When it does, the agent switches to listener mode and some other agent switches to speaker mode.

An agent acting in listener mode must carry on two parallel process that it must combine to result in a single context. On one hand it must silently generate tentative contributions to the ongoing storyline. On the other hand, it must keep its version of the story line updated with whatever contributions are provided by other agents acting in speaker mode. To achieve this, a listening agent must keep a store of his tentative contributions, which are possible continuations of the story but which have no fixed place in the story line until he actually communicates them to other participants. To achieve this, it must maintain two different versions of the context: the *current context* corresponding to contributions actually communicated by speaker agents, and a *tentative context* resulting from applying to the current context the list of his tentative contributions.

The combination of the two tasks of an agent in *listening mode* would operate as follows:

- While it receives no contributions from outside speakers, it generates possible actions to include in its contribution when its turn comes (but it does not communicate them to other participants!), and it updates its tentative context with them.
- When an agent in listener mode receives an action from another agent acting in speaker mode, it updates its own current story-context by adding that action, and it revises its tentative context in the following way: if the new action was already contemplated in its tentative context, it is retracted from it (and all tentative contributions beyond it are retracted with it); and if the new action is in conflict with some action in the tentative context, the conflicting action (and all contributions beyond it) are retracted.

A tentative action stored by one agent A is said to be *in conflict* with the set of actions communicated to it by another agent B if that action is incompatible with those described for the same character in the set of actions already communicated by B, or if it gives rise to tensions or emotional links incompatible with those arising from the same set of actions already communicated by B.

In this model, both agents prepare material in parallel, all the time monitoring the speaker's contribution, and accepting the need to revise their prepared material in the face of conflicts. This constitues a richer model of the process of improvisation than the original basic process.

A further option for enriching the model might be to ensure that each agent operate on different resources or with a different configuration of the engagement-reflection cycle.

Each agent might have different content in their knowledge structures, i.e. the previous stories for each agent can be (either slightly or radically) different. In the same way, the pre and post conditions of story actions may have some differences between agents. This will produce unique contexts, i.e. contexts that do not exist in the agents' knowledge-base and that might lead to interesting plots.

In normal conditions MEXICA evaluates the material generated during engagement each time it switches to reflection. Although MEXICA generates plots through engagement-reflection cycles, the systems is also capable of producing material employing only the engagement routines or only the reflection routines. Thus, each agent can perform in different configurations of the cycle. For example, one agent can perform under engagement only and the other under reflection only, or one under engagement only and the other under engagement & reflection, etc. That is, agents might evaluate coherence, interestingness and novelty at different times during the whole improvisation process. By default MEXICA generates three actions during engagement and then switches to reflection to evaluate the material generated. The system evaluates coherence by checking that all actions' preconditions are fulfilled. If necessary, the system inserts actions to satisfy preconditions. If the system is not able to produce an action during engagement it switches to reflection and inserts one action to continue the story. In this way, one engagement-reflection cycle is completed. The outcome of these processes might range from one to several actions (depending on how many actions are inserted during reflection). Thus, during improvisation both agents run in parallel one engagement-reflection cycle. In this way, one agent is contributing with the next action in the story and the other one is trying to prepare material in advance, i.e. to anticipate possible directions that the improvisation might take in order to avoid lags. This process changes slightly if one of the agents is running only during the engagement mode or only during the reflection mode.

In this way, we represent the fact that real actors have different knowledge, experiences, perceptions of the world, theatrical resources, etc., and nevertheless they are able to produce an improvisation.

4.2 Interconnecting the Storytellers

In order for our simulation to constitute a plausible model of colaborative storytelling as carried out by humans, it is important that the information shared between the story tellers be restricted to communication acts equivalent to saying out aloud a sentence - or a group of sentences - intended as a contribution to the story so far. For the sake of simplicity of the model, these communication acts may take the form of valid utterances in some formal or semiformal language rather than natural language sentences. This avoids to a certain degree the need to address the problem of natural language understanding, which is known to be complex. We operate under the assumption that the only requisite for our model to be plausible is that whatever form is being used to communicate be easily convertible into the internal representation that the storytellers are using. The OAA's Interagent Communication Language (ICL) constitutes a good vehicle for the type of semiformal communication that is envisaged.

With respect to implementation details, this particular task of the model has not been addressed yet, since most work has focused so far on getting the colaborative storytellers operative. It is our intention to model the actual process of conversation by launching each storyteller as an individual agent within an Open Agent Architecture setting. In this way, the OAA Facilitator would act as mediator between the agents, and the communication protocols provided would guarantee the required level of abstraction from the low level detail of communication between each storytelling process. As an additional advantage, we contemplate the possibility of taking advantage of OAA's functionality to run each story teller on a different machine, and connect them into a distributed network of storytellers.

5 Discusion

An important issue to consider is whether the proposed solution differs in any significant way from the simpler turn-taking version where each participant only starts preparing his contribution once other players have finished theirs.

This needs to be addressed at three different levels. On one hand, it is expected that the proposed solution would show an improvement in efficiency, reducing possible delays between the contributions of different speakers. However, the response times of the story generators for preparing tasks involving the type of short contributions envisaged here may be too short for any significant difference to become apparent.

On the other hand, for similar response times the quality of the contributions must be considered. If a player has been lucky and no conflicts have arisen, he can start straight away presenting a more complex contribution than he might have put together if he started preparing from scratch. If severe conflicts have appeared, severe enough to force complete rejection of everything prepared so far, the speaker would not be worse off than if he had not prepared at all. There is an intermediate possibility, where only part of the plan needs to be rejected. This situation still leaves the speaker slightly ahead of the game, since he already has some material to kick-start his contribution. Additionally, for automatic storytellers this option has the advantage of introducing a factor of variation: the contributions that may result from building upon part of a previous plan that has had to be pruned may be different from what would have been planned from scratch.

Regarding the improvisation issues that we identified earlier as key elements for this research, we would like to mention a few relevant questions. An improvisation might be considered "good" when: it is interesting, coherent and novel; and it is the result of the interaction of at least two independent agents with different content in their knowledge-bases and/or different operation modes. MEXICA provides the mechanisms to satisfy this requirement. On one hand, we are employing MEXICA's methods to evaluate the interestingness, coherence, and novelty of a story. In MEXICA a story is interesting when it includes increments and decrements of the tension; a story is considered as novel when it is not similar to any of the previous stories in its knowledge-base; and it is considered coherent when all actions' preconditions are fulfilled within the story. On the other hand, each MEXICA agent can have different knowledge since their knowledge structures are created from the files of previous stories provided by the user. So, if they are different, the atoms for each agent are different. MEXICA can work in four different operations modes; furthermore, the system includes more than 20 parameters that control different functions within system. Thus, the behavior of each MEXICA agent can be controlled by the user. If we build agents with different knowledge and behavior we avoid developing a simple turn-taking computer program. Coherence in improvisation is a very complex problem. The current version of MEXICA handles the coherence issue by modifying the material previously generated, which is not an option for improvisation. So, we require to develop new routines that help us to deal with this situation. But at least MEXICA is able to point out problems of coherence. Regarding the representation of different knowledge and experiences, in MEXICA each character has its own representation of the story-world context. So, employing the same structures each actor during improvisation can have its own representation of the word. Experiments will tell us if that is enough to generate good improvisations. Concerning emergent contingencies, when tensions like love competition or clashing emotions arise in a story, there is a good opportunity to create interesting plots. MEXICA already is capable of detecting and exploiting these situations. With respect to shared representations of the world, MEXICA employs clusters of emotional links and tensions between characters, referred to as contexts, to represent the state of affairs of the story-world. As mentioned earlier, contexts are very dynamic structures that can be easily built and modified during improvisation. So, we believe they can nicely support the representation of the new story-world created during improvisation.

The architecture described in this paper constitutes a very interesting platform from which to address some of the more mistifying issues of improvisional creativity. It may seem that setting two programs against one another may reduce the interest of the experiment to whatever restricted capabilities the automatic storytellers can model, which need not be as many or as good as a human storyteller might have shown. The option of connecting them via a multiagent architecture leaves open the possibility of developing an interface module for humans to participate in the task. In a kind of Wizard of Oz experiment, the human would initially attempt to reproduce the behaviour of an automatic storyteller, as specified in their description. However, if the interaction is recorded, all departures by the human from the specified behaviour would be duly noted, and this record could then be carefully studied to identify functionalities that might improve the performance of the automated storytellers.

An important issue is how the system would scale if more agents are used. It seems aparent that the introduction of more than one agent may result in an improvement of the variety of the resulting stories, as a result of a greater range of possible contexts being taken into account when building the story. However, each increase in the number of agents also increases the risk of fruitless computations (those that give rise to tentative actions that are later rejected either due to conflict or redundancy). A balance must be sought between the added variety and the loss of efficiency introduced by collaboration.

6 Conclusions and Further Work

An interesting issue that may need to be considered in further work is whether the actors may also use preparation or planning at a more abstract level, to introduce in the story material that they hope to be able to use at later stages. This would correspond to extending the idea of a tentative context so that agents may maintain actions as tentative even while they are operating in speaker mode, in the hope of contributing them during a later turn. This opens interesting possibilities, both in terms of how that preparation might take place and how it interacts with short-term preparation. Additionally, there is a revision problem equivalent to the one occurring for preparation, in as much as the material introduced by one player with a particular aim in mind may be exploited by the other in a different, possibly conflicting way. This would force the original player to forsake his long term plan, or at least to modify it.

Existing academic work on oral literature may provide keys to techniques and resources that human storytellers have used in the past to solve problems of lack of inspiration. These include the use of formulaic constructions to resolve descriptions of characters, locations or events, the insertion of brief messages addressed to the listeners - possibly intended to build suspense or to draw attention to particular ingredients of the story -, or the introduction of parallel stories as subtexts. Such resources may be considered as possible expansions of the architecture presented here if it is considered that their addition may improve the quality of the resulting story. They may provide a good way of covering up any noticeable gaps in the sequence when radical revision of prior preparation forced by conflict leaves a speaker with no material to start contributing immediately.

Acknowledgements

This work is partially supported by the Spanish Ministry of Education and Science project TIN2006-14433-C02-01, and research group grant UCM-CAM-910494, jointly funded by Universidad Complutense de Madrid and the Comunidad Autónoma de Madrid (Dirección General de Universidades e Investigación).

References

Agre, P. (1997). *Computation and Human Experience*. Cambridge University Press, Cambridge.

Agre, P. and Chapman, D. (1987). Pengi, an implementation of a theory of activity. In *Proceedings of the Sixth National Conference on Artificial Intelligence*, Seattle.

Bailey, P. (1999). Searching for storiness: Story-generation from a reader's perspective. In *Working Notes of the Narrative Intelligence Symposium, AAAI Fall Symposium Series.*, Menlo Park, CA. AAAI Press.

Bellinger, M. F. (1927). *A Short History of the Drama*. Henry Holt and Company, New York.

Cheyer, A. and Martin, D. (2001). The Open Agent Architecture. *Journal of Autonomous Agents and Multi-Agent Systems*, 4(1):143–148.

Meehan, J. R. (1977). Tale-spin, an interactive program that writes stories. In *Proceedings of the 5th International Joint Conference on Artificial Intelligence*, Cambridge, Mass. Morgan Kaufmann.

Moraes, M. C. and da Rocha Costa, A. C. (2002). How planning becomes improvisation? - a constraint based approach for director agents in improvisational systems. In *Advances in Artificial Intelligence, 16th Brazilian Symposium on Artificial Intelligence, SBIA 2002*, volume LNCS 2507, pages 97–107.

Pérez y Pérez, R. and Sharples, M. (2001). Mexica: a computer model of a cognitive account of creative writing. *Journal of Experimental and Theoretical Artificial Intelligence*, 13(2).

Rumelhart, D. (1975). Notes on a schema for stories. In Bobrow, D. G., editor, *Representation and Understanding. Studies in Cognitive Science*, New York. Academic Press.

Sharples, M. (1999). *How We Write: An Account of Writing as Creative Design*. Routledge.

Trastoy, B. (2005). La improvisación: estrategias productivas de una práctica escénica alternativa. hablan los teatristas. *Telondefondo*, (1).

Turner, S. R. (1994). *The Creative Process: A Computer Model of Storytelling*. Lawrence Erlbaum, Hillsdale, NJ.