

# <sup>1</sup>Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA

Rafael Pérez y Pérez\*  
Mike Sharples<sup>+</sup>

\* Laboratorio de Cibernética Aplicada.  
Centro de Instrumentos  
Universidad Nacional Autónoma de México  
Circuito Exterior s/n-Ciudad Universitaria, México D.F. 04510  
Tel (52) 5622-8602 ext 110, e-mail: [rpyp@servidor.unam.mx](mailto:rpyp@servidor.unam.mx)

<sup>+</sup> Educational Technology Research Group  
School of Engineering  
University of Birmingham  
Edgbaston-Birmingham B15 2TT, UK  
e-mail: [m.sharples@bham.ac.uk](mailto:m.sharples@bham.ac.uk)

## Key-words

BRUTUS, MINSTREL, MEXICA, computer-based storytelling, creativity, c-creativity, predefined structures, story-predictability, structure-predictability, content-predictability, engagement, reflection, TRAMs, creative distance.

## ABSTRACT

This paper attempts to establish criteria to analyse and evaluate computer models of creativity in writing. The paper provides a brief review of the antecedents of automatic story-generation and offers a proposal for the analysis and evaluation of computer models of creativity in writing. It reviews three major projects to develop computer-based storywriters published between 1993 and 2000 and analyses their approach, similarities, differences and contributions. It compares the three approaches and discusses implications for the modelling of creativity in writing and the design of future story generation systems.

## 1. INTRODUCTION.

Computer-based text generation has been an area of great interest for many AI researchers. Although there were attempts in the 1960s and 70s to generate stories and poems by computer [1-3], the origin of AI-based computerised storytellers can be traced to TALE-SPIN and the story-grammar approach. TALE-SPIN [4] was a program that produced stories by setting goals for characters and then recording their attempts to reach the goal. It demonstrated how computer techniques in problem-solving can be applied to story-telling. The problem-solving approach introduced by TALE-SPIN became the model to follow for other AI researchers working within story-telling and some related areas. Although it was successful in showing the ability of computers to generate coherent very short stories, it was only able to generate a limited range of stories within a rigid pre-defined structure. Another important limitation, which has been employed to criticise the system, is its lack of differentiation between the goals of characters and the author. Because TALE-SPIN was driven solely by the goals of its characters to solve particular problems, uninteresting stories can arise, such as: "John Bear is hungry, John Bear gets some berries, John Bear eats the berries, John Bear is not hungry anymore, the end". Story grammars were developed with the objective of creating a theory of story understanding. They represent stories as linguistic objects which have a constituent structure that can be represented by a grammar, e.g. [5-7]<sup>2</sup>.

---

<sup>1</sup> This article was published as Pérez y Pérez, R. and Sharples, M. (2004) Three Computer-Based Models of Storytelling: BRUTUS, MINSTREL and MEXICA. *Knowledge-Based Systems*, 17, 1, 15-29 and is posted with permission from Elsevier

<sup>2</sup> Story grammars have been the centre of a polemic. While some authors believe that story grammars are important for story understanding, others argue that they are incapable of generating all and only valid stories. For example, see [8-10].

Although this approach was originally developed in the context of story understanding, some researchers have employed story-grammars to produce automatic storytellers. For example, GESTER [11] is a program that generates story outlines based on a story-grammar derived from medieval French epics. Its main shortcoming (which is shared by other programs based on this approach) is its rigidity. GESTER is only able to produce stories that satisfy its grammar and is not able to modify its knowledge to generate different outcomes.

The main contribution of these pioneer works was in showing how a computer program can produce coherent and well structured texts that, in some limited respects, resemble human stories. Programs based on story-grammars indicate how grammars could be employed to generate interesting plots, a characteristic that TALE-SPIN lacks. What the early programs revealed is that generating well-formed text is necessary but not sufficient. Other characteristics of successful human stories must be explicitly addressed, such as novelty and interestingness. MINSTREL in 1993, was the first attempt to implement an explicit computer model of creativity in writing. After MINSTREL, two significant models of computerised storytellers are MEXICA in 1999 and BRUTUS in 2000. These three models are analysed in this paper.

### 1.1 Evaluation.

This section discusses the elements that a computer model of creativity in writing should have, and how they can be evaluated. An essential requirement for such computerised storytellers is that they should be based on a clearly explicated analysis of creativity. It is generally accepted that the creative process consists in generating appropriate and novel outputs, e.g. [12-14]. A text is considered appropriate when its content and style satisfy the constraints of the task, in particular the characteristics of the audience and the purpose of the text [15: 41]. As found by the first automatic storytellers, to be appropriate a computer-generated story should be interesting, coherent and well structured.

Boden defines two types of creativity. The first is psychological creativity (p-creativity) and “concerns ideas .. that are fundamentally novel with respect to *the individual mind* which had the idea.” [16: 32]. The second is historical creativity (h-creativity) and is related to “ideas that are fundamentally novel with respect to *the whole of human history*.” [16: 32]. We suggest that computer models of creativity should initially be evaluated from the perspective of an adaptation of p-creativity. A computer model might be considered as representing a creative process if it generates knowledge that does not explicitly exist in the original knowledge-base of the system and which is relevant to (i.e. is an important element of the) produced output. Note that this definition involves both inspection of the output of the program and its initial data structures. Following Boden, we refer to this type of creativity as computerised creativity (c-creativity). The definition of c-creativity is wide since a computer model of writing might include several types of knowledge, for example knowledge about the story-structure and knowledge about the content. Two programs cannot be considered equally c-creative if one of the systems generates new knowledge about the structure and content of texts, and the second only produces new knowledge about the content. This point must be considered when evaluating c-creativity. When the system cannot produce new and relevant knowledge of a specific type represented in the system, we say that the structures representing those knowledge are predefined. *Predefined-structures* are established by the designer, cannot be modified and are developed to produce a particular effect in the output.

We shall introduce the term *story-predictability* to indicate the degree to which the output of a computerised-storyteller can be predicted when the content of the system’s knowledge-structures are known. Since predefined structures imply known output features, we propose that the output’s predictability depends strongly on the amount of predefined knowledge structures. The notion of story-predictability can be split into at least two elements: the structure and the content. Structure-predictability indicates the degree to which the story-structure can be predicted, and content-predictability indicates the degree to which a sequence of events in a story in progress can be foretold. Thus, story-predictability depends on the values of structure-predictability and content-predictability. We suggest that if the story-predictability is low, the system is more likely to be evaluated as representing a creative process. As a consequence, we propose some issues to be considered while studying a computer model of story writing. It is not suggested that these are the only important aspects to consider, but they provide a good starting point for an analysis.

Knowledge data-bases are at the heart of story-development. The content and style of stories depends on the type of knowledge stored by the system, including rhetoric-knowledge, story-world knowledge and common-sense knowledge. Thus, when analysing a computerised storyteller it is important to determine:

- What type of knowledge is represented, how it is represented and the percentage of predefined structures employed by the system. All this information helps to clarify the story predictability.
- How knowledge is updated (in order to keep on producing novel outputs it is necessary to constantly update the system’s knowledge).

Regarding the processes involved in story-development, it is necessary to establish:

- How events that form a story are retrieved from memory, i.e. how the system progress the story.
- How the system maintains the coherence of the whole story.

- How the system ensures interestingness.
- How the system ensures novelty.
- How the system generates natural language structure and meaning.

Interestingness is an important characteristic and central to a critique of computer-generated stories. It can either be created explicitly by the human designer and encoded into a database of story elements, or it can be constructed by computer through an appropriate combination of content, coherence, suspense and originality.

Lastly, it is necessary to evaluate the story output. To be acceptable as a tellable story, a piece of text should satisfy certain minimum requirements. It must introduce characters and setting and advance one or more key characters through activities within the setting to produce a plot. It must show a coherence and narrative flow between activities and settings over time. It should display an overall integrity and closure, for example with a problem posed in an early part of the text being resolved by the conclusion. It should generate suspense by setting up difficulties faced by the characters or tension in the plot. It should display originality, with successive stories being substantially different. These elements, of interestingness, appropriate content, narrative flow, coherence, suspense and originality cannot be captured by objective measures (for example of syntactic complexity). They are the elements of literary criticism and in order to provide a comparative measure they need to be assessed, either singly or in combination, by human readers. This does not preclude attempts to program story generation systems to assess and improve the interestingness and novelty of their outputs (see sections describing MINSTREL and MEXICA in this paper).

## 2. BRUTUS

BRUTUS [17] is a program that writes short stories about pre-defined themes like betrayal. The system is built on a Prolog-based system called FLEX that allows complete access to Prolog, as well as defining frame-structures, relations between frames and production rules, all in an English-like format. Figure 2.1 shows a story developed by BRUTUS.

In BRUTUS the development of a story consists of the following core processes:

- The instantiation of a thematic-frame.
- A simulation-process where characters attempt to achieve a set of pre-defined goals (development of a plot).
- The expansion of story-grammars to produce the final output.

### 2.1 Instantiation of a thematic frame.

In BRUTUS, story elements like characters, characters' goals, events, etc. are represented by structures that we refer to as story-frames. The slots that constitute story-frames contain explicit data employed by BRUTUS to develop a story. For example, for the story in figure 2.1, the system has defined a story-frame that represents how a character named Hart has an evil goal of thwarting Dave Striver's purpose of obtaining his PhD. This frame can be represented as it is shown in figure 2.2.

The frame *evilGoal* includes a slot with an explicit list of actions to perform in order to achieve Hart's evil goal. In this case, the plan consists in Hart letting Striver to think that he will sign his thesis (lie to Striver) and then, at the end of the defense, refuse to sign it. In this way, the essence of the story is explicitly codified in the frame. A second type of structure employed by BRUTUS during the instantiation process is known as story-theme. This structure is designed to group all the elements (story-frames) that conform a specific story-theme. Figure 2.3 is an adaptation of an example given by Bringsjord and Ferrucci [17: 190] to illustrate how a story-theme is specified. Thematic-structures are defined in an English-like syntax where logic variables are represented by strings starting with an uppercase letter, and story-frames (or attributes of story-frames) are represented by strings like *lie*, *thwart*, *character goal*, *agent*. As it is possible to observe, in the story-theme there exist explicit references to story-frames as well as to their attributes. In other words, the story-frame explicitly establishes which are the elements that conform the story and how they are organised. All these structures defined in English-like syntax are transformed into standard Prolog programs.

### 2.2 Development of a plot.

Plot development consists of performing a set of character-actions that might modify the story-world. Such modifications can result in new actions being performed by the same or different characters, and so on. The process ends when no more actions can be executed. The development of a plot is formed by two sub-processes:

- Activation of Proactive behaviour: Performance of characters' plans to achieve their goals.
- Activation of Reactive behaviour: Performance of character-actions as result of triggering a set of production rules stored in the knowledge database.

Activation of Proactive behaviour. - In BRUTUS, characters have associated a set of explicit actions, referred to as plans, to achieve a particular goal. Each action includes a set of preconditions and a set of consequences that

might modify the story-world. When action's preconditions are satisfied, the actions associated with characters' plans are performed and the story-world might be modified.

Activation of Reactive behaviour.- Reactive behaviour consists of actions performed by characters when the requirements of a production rule in the knowledge database are satisfied and therefore the rule is triggered. It is possible to include production rules as concrete or as general as the designer decides. For example, one can define a rule that represents the typical behaviour of a committee's chair when a thesis defense is concluded. This rule indicates that, when all requirements are satisfied (e.g. the requirement that the defense is completed), the chair will request their signatures to the other members of the committee. In the same way, one can define a rule that represents the typical behaviour of a whole thesis committee after the chair has requested their signatures. This rule indicates that, when all requirements are satisfied (e.g. the requirement that the chair has requested the signatures) each member of the committee will answer the request and will sign the thesis.

Thus, during the development of a plot, the system attempts to perform characters' plans (proactive behaviour). Then, it attempts to trigger rules in the knowledge database (reactive behaviour). If any action modifies the story-world, producing that some preconditions might now be satisfied, BRUTUS attempts to perform those actions associated with characters' plans, and then it tries to fire the production rules. As a result of the development of the plot BRUTUS creates what is known as scenario. A scenario is formed by an instantiated story-theme and a group of actions performed by characters.

### **2.3 Expansion of the story-grammar and generation of the final output.**

BRUTUS employs story-grammars (which include descriptions of how to create phrases and sentences called paragraph-grammars and sentence-grammars) to represent the structure of the stories it produces. The outputs produced during the instantiation of the theme and the development of a plot processes are employed by the grammars during its expansion to generate the sentences and words that form the final output of the system. To achieve this goal BRUTUS makes use of, *inter alia*:

- A group of previously defined words to instantiate the grammars.
- Structures that explicitly link words representing objects with other words that are related, e.g. the word university is typically related to clocktowers, brick, ivy, youth, architecture, books, knowledge, scholar, sports. In this way, the system can create descriptions like "He loved its ivy-covered clocktowers, its ancient and sturdy brick, and its sun-splashed verdant greens and eager youth".
- Structures that explicitly link words representing objects with other words that modify the object (e.g. 'green' can modify 'football pitch').
- Structures that explicitly represents analogies.

### **2.4 Discussion about BRUTUS.**

BRUTUS is a system that does not represent any creative process. Although BRUTUS is the most recent of the three systems to be published, it does not incorporate innovative elements in its architecture. Instead, it merges methods employed in other programs to generate its outputs. For example, it employs frames to represent characters, places, events, goals, actions, etc., which are used to instantiate a theme-frame (c.f. MINSTREL, see Section 3); depending on the selected theme, the system develops a plot through planning and simulation (c.f. TALE-SPIN); the system employs story-grammars to establish the structure of the stories that it produces (c.f. GESTER); reactive behaviour fires rules until quiescence is reached (c.f. SOAR [18]). Thus, BRUTUS' contribution consists of merging different known methodologies into one program.

The way the authors compare BRUTUS with other similar systems is not very clear. As it is explained later (see Section 4), MEXICA was evaluated by means of a questionnaire. An interesting feature observed from the questionnaire's results is a possible correlation between those stories rated as the best and their length in lines and words. This type of information is relevant when comparing stories produced by different systems. For example, Bringsjord and Ferrucci [17: 122,124] compare the interestingness of a 36 lines story developed by BRUTUS against a 2 lines narrative found in an introductory book to AI [19: 592] and against 9 lines of a larger story developed by TALE-SPIN (it is not clear why they only present a fragment of the story. The original story has 37 lines. See [4: 200]). This type of comparison seems uneven.

## **3. MINSTREL.**

MINSTREL is a computer program that writes short stories about King Arthur and his Knights of the round table. MINSTREL is a very complex program and this paper only presents a brief summary of the core elements of the system. A detailed explanation of the model can be found in [20] and particularly in [21]. The system is a case-based problem-solver where past cases are stored in an episodic memory. Figure 3.1 shows an example of a story created by MINSTREL:

### 3.1 Schemas in MINSTREL

In MINSTREL, all the elements that conform a story are represented as schemas. In this paper, schemas are divided in two classes: 1) those employed by the system to satisfy rhetoric constraints, referred to as author-schemas; 2) those employed by the system to represent events in a story, referred to as character-schemas.

Examples of author-schemas are author-level goals, e.g. the goal of including suspense in a story. MINSTREL has 21 author-level goals divided in four main groups: thematic goals (whose purpose is to select and develop a theme), drama goals (whose purpose is to create scenes to include in the story suspense, tragedy, foreshadowing and characterisation), consistency goals (whose purpose is to keep the consistency of the story in progress) and presentation goals (whose purpose is to present the story in English). Author-level goals have associated instructions to achieve their goals. These instructions are independent blocks of Lisp code, and are referred to as Author-Level Plan (ALP). MINSTREL counts with 34 ALPs which explicitly indicates how to create scenes to include revenge, deception, beliefs, how to create the introduction of a story, its denouement, to check its consistency, and so on.

Examples of character-schemas are character-level goals (MINSTREL employs 13 of these goals, like satisfying one's hunger, changing location, causing fear in someone, finding a romantic love, etcetera), representations of humans, monsters, animals, physical objects, beliefs, emotions, actions, states (a state is a representation of a fact that is true, e.g. "Lancelot is in the city"), etcetera. Character-schemas can be linked to establish relationships between them. In this way, it is possible to construct elaborated scenes.

MINSTREL develops stories about six predefined schemas-themes known as Planning Advice Themes (PATs). PATs, and therefore stories in MINSTREL, have the following structure:

- Introduction scenes.
- Theme scenes
  - World-facts (preconditions)
  - Decision (plan to achieve a goal)
  - Connection
  - Consequence (consequences of performing the plan)
- Denouement scenes.

World-facts scenes are facts about the world, represented as character state-schemas, that must be true in order a PAT-schema can be applied (i.e. they are preconditions). Decision scenes are formed by character-schemas that represent a plan to achieve a goal. Consequence scenes are formed by character-schemas that represent the consequences of performing the plan. Connections are scenes employed to make the transition from the Decision scenes to the Consequence scenes.

### 3.3 Main processes in MINSTREL

MINSTREL performs two main processes: the planning process, which controls the author-level goals, and the problem-solving process, which focuses in achieving these goals. When MINSTREL starts the planning process is executed and an author-level goal to tell a story is triggered; it breaks down into sub-goals like selecting a story-theme, illustrating the theme and presenting the story to the reader. All these goals are kept in an author-level goal agenda. Each goal is assigned a priority value. MINSTREL selects the goal with the highest priority and passes it to the problem solving process, which finds and executes a plan to achieve the goal. If the goal cannot be achieved it is sent back to the agenda. Once a goal has been achieved, the planning process takes over again and MINSTREL selects a new goal to achieve and the cycle starts again. The cycle is repeated until the author-level goal agenda is empty.

The problem-solving process focuses in achieving author-level goals. MINSTREL's main author-level goals during the developing of a story are:

1. Selecting and illustrating a story-theme schema.
2. Generating the introduction and denouement of the story.
3. Linking the schemas that conform the story with predefined words and phrases in English to produce the final output.

One of MINSTREL's first author-level goals is to select a theme (i.e. a PAT) to develop a new story. Once the PAT has been selected it is necessary to satisfy the author-level goal of illustrating the theme. PATs are formed by a set of character-schemas representing incomplete scenes: one, two or even all the slots in each of the structures pointed by the PAT might not have assigned a value. Thus, the goal of illustrating a theme consists in employing those partially filled schemas as a search-specification or index to retrieve scenes from episodic memory and in this way instantiate the schemas that make up the theme. As part of the process of developing a new story, MINSTREL employs what is referred to as opportunistic goals. Each time a new scene is created the system verifies if an opportunistic goal can be triggered. These type of goals are used:

a) To verify the consistency of the story in progress (e.g. when the system attempts to use a PAT but its preconditions are not satisfied a consistency opportunistic goal is triggered in order for MINSTREL to insert scenes that fulfil these requirements).

b) To find opportunities to include dramatic elements in the story (e.g. when a new scene is created MINSTREL verifies if it is possible to trigger an opportunistic author-level goal to include suspense. The system knows two ways of producing suspense: including a scene where a character feels fear or including a scene of a failed escape).

Once the theme has been selected and instantiated, MINSTREL attempts to satisfy the author-level goals of producing an introduction and denouement for the story in progress. MINSTREL only can build one type of introduction. Denouements only involve characters with a tragic fate, e.g. the buried of a death character. Finally, MINSTREL attempts to satisfy the goal of presenting the story in English. The system has several author-level plans to present to the reader in a specific order all the character-schemas that make up the story: the introduction, world facts, decision, connection, etcetera. MINSTREL counts with a phrasal generator to produce English-like outputs. Character-schemas are associated with linguistic schemas representing a word or a phrase. In this way, if the phrasal generator receives as input a set of character-schemas representing a complex scene, it looks for the corresponding linguistic-schemas to produce its output.

### 3.4 Creativity in MINSTREL: TRAMS

MINSTREL's main feature is its representation of the creative process. Writing consists of instantiating all the schemas that make up the theme. When MINSTREL cannot find events in episodic memory to instantiate the theme or all available events have been employed twice in previous stories, a set of heuristics called Transform Recall Adapt Methods (TRAMS) are employed to create novel scenes (in MINSTREL, a scene that has been used more than once in a story does not satisfy novelty requirements).

TRAMS are divided in two groups. The first group only includes one member: TRAM:Standard-Problem-Solving. This is the first TRAM the system always employs during the searching process. Its purpose is to retrieve schemas from episodic memory that can be employed to instantiate a scene. Only when this TRAM fails, the second group of TRAMS —formed by the rest of the TRAMS in the system— are employed.

The second group of heuristics are grouped in memory conveniently according to their characteristics, forming pools of available TRAMs. They work in the following way: TRAMs have explicit instructions of how to make small modifications (or transformations) into the schemas specifications (or search-specifications) used as index to recall from episodic memory (i.e. how to modify the value of one or more schemas' slots). Those instructions are executed creating a new slightly different index. A recursive call to the searching process is made, TRAM:Standard-Problem-Solving is performed and MINSTREL searches memory to find an event which matches the new specifications. If no scene is found, a new TRAM is selected from the pool producing a new slightly different specifications, and a recursive call to the searching process is made to try a new attempt to match schemas in memory. This process continues until an event is matched or a processing limit is reached. Once an event is matched, MINSTREL goes back layer by layer, adapting the event found to the original specifications in each layer, until it finally reaches the top level.

The following example illustrates how TRAMS work. The goal is to create a scene where *a knight kills himself* (known as the original description). The original episodes in memory are: 1) *a Knight fights and kills a troll* and 2) *a Princess makes herself intentionally ill by drinking a potion*. One of the solutions MINSTREL comes across when solving the problem is that a Knight intentionally drinks a potion in order to kill himself. To achieve this result MINSTREL employs two TRAMS as follows. The searching process starts and TRAM:Standard-Problem-Solving is triggered. It fails and the system calls TRAM:Similar-Outcomes-Partial-Change which “recognizes that being killed is similar to being injured ... [that is, it] ‘guesses’ that an action which is known to result in injury might also result in death” [21: 116]. So, the TRAM substitutes the original description “a knight kills himself” for “a knight makes himself ill”. A recursive call to the searching process is triggered. TRAM:Standard-Problem-Solving fails so a second TRAM is applied. TRAM:Generalize-Constraint substitutes the character “knight” by “anyone”; in this way, the new specification is “someone does something to injure them self”. A recursive call to the searching process is made, TRAM:Standard-Problem-Solving is triggered and the episode where the Princess makes herself ill is recalled. The system goes back layer by layer and princess is substituted by knight and a potion to make someone ill by a potion to kill someone. This produces the scene where the knight kills himself by drinking the potion.

### 3.5 Evaluation.

A story produced by MINSTREL was evaluated by means of an Internet questionnaire. Nine subjects responded to it. They did not know that the story they were evaluating was written by a computer program. The subjects were asked to answer questions related to the age, education and sex of the hypothetical author

of the story, as well as questions regarding the quality of the story. The following lines show the mean scores for some of the answers regarding the author: author-age 15.8; author-education: 0.9 (0=grade school, 1=high school, 2=college, 3=Graduate school); sex: 0.4 (0=female, 1=male). The following lines show the results obtained for the story (in a scale from 1 to 5 where 5 indicates best): Overall rating of story: 1.5; Clever plot: 2.5; Attention to details: 2.8; Coherency: 3.6; Use of language: 2.1 (the story employed to evaluate MINSTREL as well as the whole questionnaire can be consulted in Chapter 15 in [21]). In a different questionnaire, where seven computer-created stories were evaluated by fifty persons from twelve different countries (see table 4.2), MINSTREL obtained a better evaluation rate with the story in figure 2.1. (As an interesting anecdote, one of the subjects that answered that questionnaire asked if MINSTREL's story was really created by a computer. See Chapter 6 in [22]).

### 3.6 Discussion about MINSTREL.

MINSTREL is a powerful and complex system capable of producing interesting and novel outputs. It is a major advance on TALE-SPIN in terms of the complexity of the program and sophistication of the output:

- MINSTREL is one of the first computerised storytellers (probably the first) that explicitly represents a computer model of the creative process of writing.
- MINSTREL's main contribution to the computerised creative process in writing is the concept of TRAMs, which has demonstrated the potential power of small changes.
- MINSTREL is one of the first computerised storytellers that explicitly employs author-level goals, combined with character-level goals, to generate adequate and interesting outputs.
- MINSTREL generates novel stories by transforming "old" episodes loaded in memory.

However, there are important aspects to be improved. Although TRAMs seem to be a powerful tool, sometimes they appear to be written for the special purpose of achieving a specific scene. For example, in the case of the Knight who wants to commit suicide (see Section 3.4), it is difficult to picture how TRAM:Similar-Outcomes-Partial-Change, which "guesses" or "recognizes that being killed is similar to being injured" [21: 116], can work in a different context. To illustrate this situation, the reader can imagine a Knight who is sewing his socks and pricked himself by accident; in this case, because the action of sewing produced an injury in the Knight, MINSTREL would treat sewing as a method to kill someone. Something similar occurs with TRAM:Similar-Outcomes-Implicit, another heuristic to interchange story-elements. This heuristic suggests that "two outcomes are interchangeable in *every* situation if it [the system] can recall [retrieve] *any* situation in which they are interchangeable" [21: 120]. Thus, if in episodic memory there is a scene where a knight kills a monster, and another scene where a knight kills a dragon, MINSTREL "guesses" that monsters and dragons are interchangeable. One wonders what happens if in memory there is a scene where a knight kisses his horse after a battle, and in the following scene the knight kisses his girlfriend. Would MINSTREL "guess" that girlfriends and horses are interchangeable? Turner is aware of the possibility of producing strange outputs and explains it saying that "A creative problem-solver *should* make errors." [21: 120]. However, this does not justify the fact that some TRAMs only work in very concrete situations and clearly seems to be designed to produce a particular scene.

TRAMs are selected using the search-specifications as an index, i.e., episodic memory contains past problems and associated TRAMs; thus, the programmer decides which TRAMs go together with particular episodes (MINSTREL does avoid the issue of how TRAMs are acquired). This association TRAM-episode might lead to programmer-influences in the way MINSTREL develops a tale. That is, when developing a story MINSTREL is not really selecting which TRAMs to use but trying all the TRAMs previously selected by the programmer for a specific episode.

One of the most impressive characteristics of MINSTREL is its capacity to create stories where revenge, deception, mistaken beliefs, etc. take place. However, these situations are not really created by the system. MINSTREL has ten ALPs heuristics that explicitly indicate the structure that such scenes have and the way to construct them. So, again procedures with precise information on how to achieve particular events play a considerable role in MINSTREL's outcomes. Another important limitation comes from its lack of flexibility in the organisation of its tales: MINSTREL can only produce stories with four different themes, which are structurally predefined and which only can turn around a planning process.

MINSTREL is a very complex program which has pointed out the utility of small modifications as a way to solve problems. Although TRAMs in MINSTREL seem at times to be too specific, they can be a powerful tool during problem solving. Also, MINSTREL indicates the importance of author-level goals in storytelling, particularly theme, consistency, drama and presentation's goals. On the other hand, TRAMs tailored to the specific genre of the story, inflexibility in story's structure, and rigidity in the structure of some scenes are the most important limitations of MINSTREL.

## 4. MEXICA.

MEXICA is a computer model based on the engagement-reflection cognitive account of creative writing that produces stories about the Mexicas (the old inhabitants of what today is México city, also wrongly known as Aztecs). During the engagement-mode the system produces material driven by content and rhetorical constraints avoiding the use of explicit goal-states or story-structure information. During the reflection-mode the system breaks impasses generated during engagement, satisfies coherence requirements, and evaluates the novelty and interestingness of the story in progress. If the results of the evaluation are not satisfactory, MEXICA can modify the constraints that drive the production of material during engagement. In this way, the stories produced by the program are the result of the interaction between engagement and reflection. Figure 4.1 shows a story developed by MEXICA. For reasons of space it is impossible to describe all the system. A detailed description of the whole MEXICA system can be found in [22]; a detailed description of the core processes in MEXICA can be found in [23]. A detailed description of the engagement-reflection account of writing can be found in [15].

### 4.1 Story-World Contexts.

MEXICA builds its knowledge mainly from stories provided by the user. In MEXICA, stories are formed by sequences of actions. Thus, first it is necessary that the user defines a set of valid story-actions, and then write a group of stories named Previous Stories. MEXICA processes the Previous Stories to build structures in memory representing content and rhetorical knowledge. Story-actions and Previous Stories are defined in a text file through a set of syntactic rules known as definition languages designed for that purpose. Thus, the steps to build knowledge in memory are:

- The user defines a set of story-actions.
- The user defines a set of previous stories.
- MEXICA builds in memory content and rhetoric knowledge through information obtained from the previous stories.

One of the main problems in story generation is the production of a coherent sequence of actions. In previous models this problem has been solved by predefining the structure of the story to be developed, or by predefining the actions a character can perform (or a combination of both), assuring in this way that events in the story in progress follow a logical path. MEXICA presents an alternative approach. It is assumed that characters' reactions to any particular event are conditioned by the story-world context at the moment the action occurred. For example, if a knight walking in the forest suddenly perceives a dangerous situation, let us say the presence of a bear, probably this character will react by walking fast in order to avoid it. The context in the forest conditioned the behaviour of this hypothetical walker. In the MEXICA system, it is assumed that a coherent sequence of actions can be produced by linking events through the story-world context surrounding them, avoiding in this way the use of explicit goal states or pre-defined story structures.

In MEXICA, stories are formed by explicit and tacit information: the former consists of actions performed by characters that are registered in structures representing story-events, and which later are transformed into the words that form the written story; the latter is formed by consequences or post-conditions of story-actions that modifies the story-world, which are registered in structures referred to as story-world context (SWC) and which are never shown as part of the narrative. In MEXICA, consequences of actions produce emotional links between characters, modify the dramatic tension in the story (see section 4.2 for an explanation of the tension), or produces changes of location for the characters. Post-conditions are specified by the user as part of action's definition. An example of explicit information is the action *Princess cured Jaguar Knight's wounds after the battle*. As a consequence of the performance of this explicit action the knight felt an enormous gratitude towards the princess. This consequences is a tacit information classified as an emotional link between characters in the story. Since each character in the story has its own SWC, a single action might produce several different SWCs, which are all stored in memory. SWCs are not only an accumulation of stories' consequences, they are dynamic structures which are constantly modified by 'garbage-collection' routines and by what is known as inferred post-conditions. For example, the following lines illustrate how the inferred post-condition due to love competition works. If the system detects that two characters are in love with the same third character, the system modifies the characters' SWCs incrementing the dramatic tension due to love competition. If any of the three characters die, the system decrements the dramatic tension in the SWCs (it is not possible to have a tension due to love competition if any of the three characters die).

MEXICA performs two core processes: creation of knowledge structures in memory; production of stories through the engagement-reflection cycle.

### 4.2 Knowledge Structures: Abstract, Concrete and Tensional Representations.

MEXICA obtains from the previous stories its content and rhetoric knowledge. It is assumed that previous stories represent well-constructed stories, therefore the structures obtained from them represent correct



knowledge. In order to load content and rhetoric knowledge in memory, MEXICA executes each of the story-actions that form the previous stories, building SWC structures (with the consequences of each action) and linking to each structure the following action in the previous story. In other words, the system associates groups of following next actions to perform to specific story-world contexts. In this way, during the engagement-reflection cycle MEXICA can employ the SWCs of the story in progress as cue to probe memory and retrieve a next event. The group of SWCs and associated following next actions built from the Previous Stories are known as the Abstract Representation. None of the characters in the Abstract Representation is instantiated. This is important because the instantiation of characters during the development of a new story proved to be relevant to produce novel and coherent sequences of actions (a detailed explanation of the characters-instantiation process can be found in [22]).

The Concrete Representation is a copy in memory of the previous stories. It is employed to make explicit analysis of the previous stories during the breaking of an impasse.

In MEXICA, stories are formed by actions that might lead characters to a degraded or improved state [24]. During a degradation process the dramatic tension in the story increases due to the presence of obstacles that prevent a character from achieving a more satisfactory state. Conversely, an improvement process is characterised by actions that eliminate those conditions that stand against a more satisfactory state. In MEXICA, a story is classified as interesting when it includes degradation-improvement processes. The system includes a variable known as Tension to the Reader that represents numerically the dramatic tension in the story at any specific moment. The Tension to the Reader can increase or decrement its value as a result of actions' consequences or inferred post-conditions. A structure referred to as Tensional Representation captures the value of the Tension to the Reader over time. In this way, it is possible to visualize the Tensional Representation as a graphic (see figure 4.2). MEXICA compares the Tensional Representation of the story in progress against the Tensional Representation of the Previous Stories during the evaluation of interestingness in the Reflective State (see Section 4.3).

In summary, MEXICA represents content and rhetoric knowledge at different levels of abstraction: Tensional Representation, Abstract Representation and Concrete Representation. The Tensional Representation is a vector that records the different values of the variable tension to the reader over time. The abstract representation are non-instantiated schemas that links story-world contexts to sets of logical possible next actions to be performed in the story in progress. The Concrete Representations are schemas that represent in memory the previous stories. During the engagement-reflection cycle the system employs the three types of representation to generate a story.

#### **4.3 The engagement-reflection cycle.**

During engagement MEXICA retrieves actions from memory avoiding the use of explicit goal-states or predefined story-structures. The engagement cycle is formed by the following steps:

1. An action is performed by a character (the first action in the story is given by the user).
2. The consequences of this action modify characters' SWC.
3. MEXICA employs the SWCs as cue to probe memory.
4. When an SWC matches a structure in memory (in the abstract representation), the system retrieves the set of possible next actions associated to it.
5. Routines, known as Filters, eliminate all those possible next actions that do not satisfy a group of constraints known as guidelines (see below for an explanation of filters and guidelines).
6. One of the retrieved actions is selected at random as the next action in the story.
7. Such an action is performed by a character, modifying the SWCs, and the cycle starts again in step 3.

Filters are a set of routines which have a very important role in the production of appropriate material during engagement. Its main function is to remove those possible next actions retrieved from memory which do not satisfy the requirements imposed by the guidelines. Through them MEXICA assures that the recommendations set by the Reflective State are attended during engagement. In this way, requirements of novelty and interest are fulfilled. Also, filters guarantee that all possible next actions retrieved from memory will help the story in progress to flow. Filters do not perform any activity that could be considered as part of the Reflective State. They do not evaluate the closeness of an action to a goal state, explicitly explore a problem space, etc. Filters only constrain the universe of possible next actions to those that are appropriate for the story.

If no action can be retrieved from memory or all retrieved actions are eliminated by the Filters, MEXICA employs different strategies to try to match structures in memory. Basically, they consist in:

- Modifying the SWC used as cue and probing memory again.
- Instantiating the characters of the retrieved possible next actions in surprising ways (a detailed explanation of how cues are modified and how character-instantiation routines work can be found in [22]).

In this way the system is able to produce novel sequences of actions. The engagement cycle ends when a fixed number of actions have been retrieved and performed or when no more actions can be retrieved from memory provoking an impasse.

During reflection MEXICA performs three main processes:

- breaks impasses
- verifies the coherence of the story in progress
- evaluates the novelty and interest of the story in progress.

An impasse is declared when a story-action produces SWCs that cannot be matched with any structure in memory, or when all the actions retrieved from memory do not satisfy a set of requirements known as guidelines. The system attempts to break it by employing a method similar to those used by case-based models: it searches in the previous stories (Concrete Representation) for a tale that contains the same action that triggered the impasse and tries to apply the same “formula” in the story in progress.

In order to explain how MEXICA verifies the coherence of the story in progress, it is necessary to introduce a new element: action’s preconditions. When a story-action is defined, the user can include a set of preconditions as part of its definition. They represent requirements that must be satisfied before a story-action can be performed. Since during engagement preconditions are ignored, it is during reflection that MEXICA verifies coherence by checking that preconditions in all actions forming the story in progress are fulfilled. Thus, if during reflection an action with unsatisfied preconditions is detected, the system inserts an event in the story in progress whose consequences satisfy the unfulfilled preconditions.

Before describing the way MEXICA evaluates the story in progress it is necessary to introduce the concept of guidelines. After an evaluation is performed, MEXICA determines if the story in progress is developing adequately. When the evaluation is not satisfactory MEXICA employs a group of structures, known as the guidelines, to constrain the type of actions retrieved from memory during engagement (see above how filters work during engagement). MEXICA evaluates originality by comparing the content of the current story with the previous stories. If they are similar, the system sets the guidelines to constrain the retrieval of actions from memory during engagement to more original events. MEXICA evaluates interestingness by comparing the changes in the dramatic tension (Tensional Representation) of the current story with those of the Previous Stories. If, for example, as result of this comparison it is detected that the story is boring (i.e. there is not an increment of the tension) the system sets the guidelines to constrain the retrieval of actions from memory during engagement to those that increment the tension to the reader (guidelines can also be set to decrement the tension or keep the same value). Guidelines are only updated during reflection and they only affect the retrieval of actions during engagement. In this way, the evaluation process affects the retrieval of actions during engagement.

Stories in MEXICA are the result of the interaction between engagement and reflection. During this process actions are not produced linearly. Some actions are generated during engagement; then, during reflection, the guidelines are updated and the story is modified to satisfy coherence requirements inserting new actions along the material produced during engagement. Thus, when the system switches back to engagement, the SWCs and the constraints that drive the production of material have been modified: new actions are retrieved from memory, and the system switches to reflection where the guidelines are updated, the story is evaluated and if necessary modified, and so on.

Once a story is finished —i.e. when the engagement-reflection cycle ends— the system performs a process called the final analysis. The purpose of this analysis is to make clear the motivation that characters in the story have to act in a particular way. Thus, the story becomes more coherence. To achieve this goal, MEXICA examines the story to determine where it is possible to insert either actions representing characters’ goals or actions that explicitly represent tensions between characters. Thus, this process examines the story and inserts actions where required. Some of these actions are represented as explicit goals to be reached. In this way, although a story produced by MEXICA might seem to be a goal-oriented product, the system never employs explicit goals to produce it. We suggest that, although the generation of some creative products might be explained in terms of explicit goals, they do not necessarily have to be the result of a goal-directed planning, but rather that goals can be inserted retrospectively to explain the product.

In MEXICA, all the story-actions have associated with them one or more phrases that are employed to produce the final output in English-like format. These phrases are established by the user at the moment of specifying the list of available story-actions. MEXICA includes different operation modes that allow it to produce stories that follow different writing strategies (see table 4.1).

MEXICA was evaluated by means of an Internet questionnaire. Fifty subjects from twelve different countries answered it. They were asked to rate computer-generated stories on a 5 point scale (from ‘very poor’ to ‘very good’) for narrative flow and coherence, narrative structure, content, suspense and overall quality. Four stories

developed by MEXICA under different operations mode formed the questionnaire. One each of the stories was generated under the E2 operation mode (engaged mode only with filters), E2-boring (engaged mode only with filters and with the variable “tension to the reader” forced to have low values), ER2 (the complete model with engaged and reflective mode and the full set of constraints, see figure 4.1), and ER2-boring (engaged and reflective mode with the variable “tension to the reader” forced to have low values). It was hypothesised that ER2 would gain the highest rates while E2-boring would gain the lowest rates. No prediction was made about the order of the ratings for E2 and ER2 boring. For comparison reasons, a story generated by MINSTREL (see figure 3.1) and a story outline produced by GESTER [11] were included. Finally, a story written by a human in a computer-like language but aimed to satisfy requirements of originality and interest was also included in the questionnaire. The results can be observed in table 4.2. ER2 obtained the highest rates in all categories. The ER2 story got the highest rate in the suspense category while the ER2-boring and E2-boring stories obtained two of the lowest rates. These results suggest that the degradation-improvement process employed by MEXICA to produce interesting stories works correctly. It is interesting to notice how the ER2 story was rated higher than the human-generated story. From these results it is not possible to conclude that MEXICA is a better storyteller than a human author, but perhaps that the human was less effective in accomplishing the task of creating a somewhat unnatural story.

#### **4.4 Discussion about MEXICA.**

MEXICA is one of the first storyteller based on a cognitive account of writing. The system achieves two important goals: it embeds a general cognitive account of writing into a computer model and it demonstrates the plausibility of the engagement-reflection cognitive account of writing.

The system demonstrates several innovations:

- It employs a novel methodology to generate novel outputs from previous stories.
- It employs three different structures to represent content and rhetoric knowledge at different level of abstraction. This knowledge-structure allows a good level of flexibility at the moment of developing new stories.
- It avoids the use of explicit goals or predefined structures to drive the generation of material. This point is important since many AI and cognitive sciences researchers have seen the creative process as a problem solving activity where explicit initial and final-goal states drive the generation of material.
- It is the first system that avoids the use of predefined structures to assure interestingness. Therefore, MEXICA employs routines that not only evaluate the novelty but also the interestingness of the story in progress.
- It has an explicit management of story’s characters, i.e. based on the characteristics of the story in progress, the system decides when to introduce a new character, when to employ an old one, etc.
- It is simple to modify the group of available story-actions and previous stories loaded in memory, allowing the user to have control of the system’s knowledge.

The system also has limitations. A feature of this research is the ability of the computer model to adapt its store of previous stories to create novel ones but, as Ben du Boulay has pointed out (personal communication) a weak point is the partial analysis of the theoretical limits of this methodology. It is necessary to have a deeper understanding of its possibilities to generate novel stories.

Predefined elements, which usually imply rigidity, are always a delicate point in any automatic story-teller. In MEXICA, inferred post-conditions (e.g. love competition) are specified as part of the code. Therefore, they are fixed and lack flexibility. This can be a problem, since one of MEXICA’s important characteristics is its facility to modify story-actions. If new events are added to the system which generate new and obvious inferred post-conditions, MEXICA would not be able to handle this situation. In the same way, although most of the filters are very flexible (they have the capacity of adapting to the story in progress), the number and type of filters are fixed in the code. Since filters are an essential part of engagement, this situation might prevent the system from generating stories in a different genre due to unforeseen problems solved by filters.

### **5. GENERAL DISCUSSION.**

This paper has analysed three large projects to develop computer-based models of storytelling. In this section the three systems are compared and evaluated according to requirements proposed in section 1.2: knowledge representation and story predictability, knowledge updating, retrieval of adequate and coherent events, interestingness, originality, and natural language production. Sometimes, two or more of these aspects can be found in a single process; therefore, they are not necessarily presented in this discussion in the same order as they are listed in section 1.1.

It is known the importance of designing systems that allow updating their knowledge data-bases in the simplest possible way. This seems particularly relevant in computerised storytellers, which depends on their knowledge to develop new stories. The following lines illustrate how knowledge structures are created and updated in the three programs that are analysed in this paper. Knowledge structures in BRUTUS are constructed through a program called FLEX that allows, in an English-like format, defining frame-structures, relations between frames and production rules. The steps that the designers follow to load knowledge into the system are: a) they create a story; b) they represent the story in terms of frames, rules and grammars in an English-like format; c) they employ FLEX to load that knowledge into the system. Knowledge structures in MINSTREL are largely entered by a graphical interface that allows dropping frames from a menu and interconnecting them with relationships. Although scenes might be formed by complex nets of linked schemas, once they have been designed it is easy to build and load them into the system. Scenes created by MINSTREL during the developing of a new story can be stored as part of the episodic memory and used in the future to create novel material. In MEXICA, knowledge structures are constructed by taking the sequence of actions represented in the previous stories and story-actions and creating the abstract, tensional and concrete representations. Any user of the system can add, remove or modify the group of story-actions and previous stories. This information is stored in text files and follows a simple set of syntax-rules. Any new story created by the system can be incorporated into the previous stories file and therefore into the system knowledge structures. Thus, MEXICA and MINSTREL are able to feedback the stories they create as part of their knowledge data-bases.

The production of interesting and coherent sequences of actions represents one of the main problems in story-generation. TALE-SPIN has been criticized for producing boring stories. MINSTREL and BRUTUS address this problem by predefining story-structures and assuring in this way the production of interesting material. Turner explains how “by structuring the stories it tells around themes, MINSTREL assures that they will have the purpose that was missing from stories told by TALESPIN” [21: 213]. Since the structure of stories are explicitly encoded in the themes, structure-predictability in BRUTUS and MINSTREL can be classified as high. MEXICA employs a different tactic to solve this problem. It uses story-world contexts to link actions in a logical succession of events and the tensional representation to generate interesting stories, avoiding in this way the use of explicit predefined structures. Therefore, its structure-predictability can be classified as lower. The implementation of routines to assess the interestingness of the story in progress indicates that it has not been explicitly codified as part of the knowledge structures by the designer (see table 5.1).

Notice that to employ pre-defined structures assures that the story in progress will satisfy the desired level of interestingness. However, the high structure-predictability of this approach brings as a consequence an important lack of flexibility at the moment of developing a new story. This is an important limitation since a computer model of creativity in writing must be able to produce different types of interesting stories. The improvement-degradation method employed by MEXICA is much more flexibility in this respect. However, these improvement-degradation processes are only one of several possible ways of evaluating the adequacy of stories. For example, a story which does not include any tension to the reader can catch the interest of an audience by describing a novel perspective of a social fact. MEXICA cannot be asked to produce a story with this characteristic. Evidently, it is necessary to continue studying different options to generate interesting stories.

The following aspect to consider is how the systems retrieve appropriate and coherent material from its knowledge-base so as to progress the story. MINSTREL does not develop a story by retrieving a succession of actions. Instead, it employs a set of partially instantiated episodic schemas as indexes to retrieve episodes from memory. MEXICA employs a different strategy. It builds in memory, from the set of previous stories, a group of story-world contexts and associated logical next actions to perform. Then, it calls on the story-world contexts of the story in progress to probe memory, find a similar context and retrieve an associated and adequate next action. MINSTREL and MEXICA employ a set of preconditions to ensure the coherence of the story in progress. In MINSTREL, preconditions are associated with PATs while in MEXICA preconditions are coupled with story-actions. In both cases, events are inserted into the story in progress to satisfy the preconditions and assure the consistency of the output. In this way, MEXICA and MINSTREL represent writing as a no linear process; e.g. in MEXICA the first event generated by the system might be the last one in the story. This contrast with BRUTUS, which produces its material in an ordered way. BRUTUS selects a frame-theme, searches for elements to instantiate the theme, and builds a stage (which is the instantiation of the theme with characters and actions). The system then expands its story-grammar to allocate each element of the story in the right place. In this way, it is assured to employ adequate and coherent actions.

How does the system ensure originality of output? As with TALE-SPIN, BRUTUS does not include any process that can be considered as representing a creative process. It does not incorporate any evaluation routine to assess novelty, since all elements that comprise its output are previously predefined. MINSTREL evaluates novelty by registering the number of times an episode has been employed in previous stories. When the counter exceeds a threshold, MINSTREL employs TRAMs to transform episodes in memory to generate novel outputs. MEXICA evaluates novelty by comparing the content of the story in progress with all the previous stories. When it detects

that they are too similar (and therefore it is necessary to improve the novelty), guidelines are set to obtain novel actions during engagement. MEXICA then employs different methods to produce novel stories, such as modifying SWCs to retrieve original actions from memory and attempting to employ characters in ways that might produce novel situations. In general terms, BRUTUS has a high story predictability, since both the structure of the story and the actions that characters perform are predefined. In MINSTREL, predefined-schemas establish the structure of the story in progress. Thus, it has a high structure predictability, but there is novelty in the content because the actions that characters perform are not predefined. In MEXICA, neither the structure nor the events are cast as predefined structures (see table 5.2).

There is an important aspect related to content-predictability in MINSTREL and MEXICA that must be pointed out. Both systems are able to produce novel scenes. However, the schemas that represent those scenes are fixed and neither MINSTREL nor MEXICA can modify them to create novel structures. For example, in MINSTREL a character's plan is always represented by a goal-schema, act-schema and state-schema. MINSTREL cannot invent new types of schemas, novel ways of link them or new values to fill the schemas' slots. In MEXICA, story-actions have a fixed organisation and they cannot be altered: e.g. a story-action involves only a maximum of three characters. The system cannot create new story-actions or produce novel consequences of actions (this aspect is important because story-world actions are employed to link sequences of events, and they are formed by actions' consequences). How much does this situation affect the content-predictability? There may come a point where the content of the outputs produced by these systems would no longer be original. This leads to the idea that, in order to evaluate a computerised storyteller, it might be necessary to subdivide content-predictability into three new divisions: content-structure predictability (how predictable —static or flexible— are the structures that represent the events in a story and how they affect the content of the text?), content-value predictability (how predictable are the possible values employed to instantiate a schema, that is, can the system produce novel values to fill schema's slots?) and content-action predictability (how predictable are the actions that a character might perform in a given situation?). Thus, MINSTREL and MEXICA have a high content-structure predictability and content-value predictability, and a low content-actions predictability. This type of analysis could result in endless subdivisions and requires a deeper study, which is outside the scope of this paper. In this work we keep the original content-predictability definition and leave the study of the subdivisions for later works.

Bringsjord and Ferrucci suggest a different way to evaluate computer creativity. They introduce the term *creative distance* as the difference between a program's initial data and the output it can generate. In this way, they suggest that "if we look 'under the hood' of a program and find it trivial for a human to transform the program's initial data to the program's output, we are less likely to consider the program creative" [17: 161]. Can the creative distance be useful to evaluate how creative is a computerised storyteller? We believe the answer is no and we use an analogy to explain why. Imagine that you paint a beautiful landscape and transform it into a 1,000 piece jigsaw-puzzle. Now, imagine that each of these pieces is marked with a special code that indicates the order in which each of them must be assembled. If someone mixes all the pieces and, by following the code, re-creates the landscape, would you consider it a creative process? Probably no, even when the initial data (the 1,000 mixed small pieces) is very different to the final result (the beautiful landscape). The reason is that the result of the process has been previously established (the painting was made by you) and therefore one knows the final output. That is the way BRUTUS works and that is why we do not consider it to be a model that represents the creative process. In an analogous way, all the story-elements that comprise BRUTUS have been marked with a special code and the system follows that code to assemble those elements and re-create a story. Thus, it can be described as a system focused in assembling predefined story-elements. In this way, BRUTUS belongs to the group of computerised storytellers like TALE-SPIN and GESTER which are not focused on representing the creative process but on generating well-structured narrative texts.

But, do not MINSTREL and MEXICA also assemble predefined schemas or story-actions to produce their outputs? The answer is yes. The main difference consists in the number of predefined elements employed by each system (see the definition of predefined structures in section 1.1). In BRUTUS, all the story elements are predefined. By contrast, MINSTREL and MEXICA are able to produce some relevant knowledge for the stories in progress which originally was not present in their knowledge-base. Therefore, their process not only consists in assembling predefined elements but in generating novel situations which are relevant for the new story.

The final consideration is the production of natural language. In all the models of writing mentioned in this paper, the words and phrases output by the programs have been predefined by the designers. They are canned words. MINSTREL and BRUTUS employ complex phrasal generators to produce the final version of their stories. MEXICA associates with each story-action a set of phrases that are employed by the system to produce the final output. None of the systems analyzed in this paper generate as part of its creative process the words and sentences that form the story. We suggest that the production of well formed language is an integral part of the creative process of writing and it has a strong interaction with the generation of coherent and interesting

sequences of events during the development of a story. This is a major limitation of all systems analysed in this paper and it is important to address this aspect in future designs of computer models of writing.

In order to evaluate the stories Turner employs a questionnaire, where a story by MINSTREL is presented to nine subjects who are asked to evaluate it. The subjects are not told that the story was written by a computer program and they are asked to answer different questions regarding the age, education and sex of the hypothetical author of the story, as well as questions about the quality of the story. Pérez y Pérez designed a questionnaire where 50 subjects were asked to compare different aspects of computer-made stories, such as narrative flow and coherence, structure, content and suspense. The authors of BRUTUS do not report any kind of evaluation. Questionnaires like those employed by Turner and Pérez y Pérez seem to be a good option.

## 6. CONCLUSIONS.

This section discusses what we can learn from these programs with regard to creativity in writing. The development of computer models of writing allow us to explore possible mechanisms for creativity in human beings, as well as enabling a study of alternative computer systems to generate novel outputs. If a program component contributes towards generating an interesting or novel story, that alone is not proof of its existence in human cognition but it does point to a possible mechanism for creativity and also provides a demonstration that aspects of creativity can be modelled by computer. At present, it is only possible to simulate a small part of the creative process of writing and as a result computerized storytellers are far from producing human-like stories. However, the possibility of observing how the models work and how their different parts interact with each other to generate a story, is a valuable resource to test the plausibility of writing models and to gain a clear picture of their limitations.

It is interesting noticing that, although MINSTREL and MEXICA represent the creative process in very different ways, they share some features:

- Predefined opportunistic goals (MINSTREL), inferred post-conditions (MEXICA);
- Predefined preconditions (PATs in MINSTREL and story-actions in MEXICA);
- When necessary, both systems insert events in the story in progress to satisfy preconditions;
- Evaluation of story coherence;
- Evaluation of story novelty;
- Similar content-predictability;
- The generation of natural language based on predefined words and phrases.

Meehan notes that “Stories are certainly *more* than problem-solving narratives, but they may not be less” [4: 203]. Pemberton states that “We need a much richer model of story processing than that which could be offered by either the story grammar or the goal directed model alone” (Pemberton cited in [25: 149]). Thus, much more research is needed in this area. We suggest that these tentative findings are a more accurate depiction of the current state of research than the more conclusive pronouncements. Turner, for example, claims that MINSTREL is a “prima facie evidence that artistic ability can be explained in terms of problem-solving, and that *no further or different cognitive process need be stipulated*” [emphasis added] [21: 244]. This claim would appear to be precipitate. Furthermore, this statement is later contradicted when Turner claims that his phrasal generator

reflects the intuitive difference between explicit, reasoned decisions about how a story will be presented to the reader and the implicit, unconscious use of language... [T]he author’s language generation capability is at least partly at the conscious level and at least partly below the conscious level. Explicit author-level presentation goals [which are an essential part of MINSTREL] capture conscious reasoning ...while knowledge [i.e. predefined knowledge-structures] in the phrasal lexicon captures the author’s fluent, automatic language generation capability [21: 447].

Thus, Turner appears to conclude that other cognitive processes than those explicitly addressed by MINSTREL are involved in writing. Bringsjord and Ferrucci assert that “human (literary) creativity is beyond computation” [17: 149]. It would appear that BRUTUS is not intended to model a creative process: “Brutus did not and does not originate ... stories.” [17: 102]. However, systems like MINSTREL and MEXICA suggest that, at least in part, creativity can be represented in computer terms. Bringsjord and Ferrucci also assert that “... our ability to decide whether stories are interesting is one that mere computing machines must forever lack” [17: 106]. However, the results obtained from MEXICA’s evaluation challenge this claim. Table 4.2 shows how stories with a low tensional representation are evaluated as lacking suspense and boring. We do not claim that the method of the tensional representation can accurately conclude if a story is interesting, but that it suggest that at least part of the process to evaluate the interestingness of a story can be represented in computational terms.

In general, it is possible to conclude the following. BRUTUS is built on the assumption that creativity cannot be represented in computer terms and it produces stories by assembly of pre-defined story elements. MINSTREL is

built on the assumption that problem solving is sufficient to explain creativity, and the method it employs is to produce stories by the achievement of explicit goals. MEXICA is built on the assumption that problem solving is only a part of the creative process and that engagement plays an important role in writing; the method employed to produce stories is the engagement-reflection cycle.

The analysis of MINSTREL, MEXICA and BRUTUS point out the importance of:

1. How events are represented in memory: as episodic memory, individual actions, etc.
2. How events that form a story are chosen and how the system ensures that they fit together (i.e. that they produce coherent sequences of events) in the story in progress.
3. How the system ensures interestingness in the story in progress.
4. How the system ensures novelty in the story in progress.
5. How the system ensures an adequate story-structure.
6. How much the system depends on predefined structures to produce adequate outputs.
7. The kind of evaluation that the system performs on its output.

One significant difficulty in assessing computer models of story generation is that there is no generally agreed basis for comparison. To assist future research, we would propose a set of basic benchmarks, as follows:

1. The program's knowledge base should be available for inspection, in human readable form.
2. The designers should state those aspects of creativity that the program is intended to model.
3. The designers should state the intended audience for the stories.
4. The program should be capable of generating at least ten stories from the same knowledge base, without human alteration of the program.
5. The designers may then select three outputs from the ten for evaluation.
6. The three stories should be rated for novelty with respect to the original knowledge base (either by the designers or an independent assessor).
7. The same three stories should be judged for overall quality, originality and interestingness by independent raters who might be unaware that the output is machine-generated.

Creativity is a fascinating area where there are too many things waiting to be discovered. Hopefully, this paper will encourage further work in computer models of creativity.

## **7. ACKNOWLEDGEMENTS.**

We want to thank Scott Turner for very useful comments on this paper. The MEXICA project was sponsored by the National Council of Science and Technology (CONACYT) in México.

## **8. REFERENCES.**

- [1] Klein S, Control of Style with a Generative Grammar, *Language* 41 (1965) 619-631.
- [2] Meehan J R, *The Metanovel: Writing Stories by Computer*, PhD Thesis, Yale University, 1976.
- [3] Sharples M, *Poetry from LOGO*, DAI working paper no 30, University of Edinburgh, 1978.
- [4] Meehan J, *TALE-SPIN*, in: Shank R C and Riesbeck C K, Eds., *Inside Computer Understanding: Five Programs plus Miniatures*, (Erlbaum, Hillsdale NJ, 1981) 197-226.
- [5] Lakoff G P, Structural complexity in fairy tales, *The study of man* 1 (1972) 128-190.
- [6] Rumelhart D E, Notes on a schema for stories, in: Bobrow D G and Collins A.M, Eds., *Representation and Understanding: Studies in Cognitive Science* (Academic Press, New York, 1975).
- [7] Mandler J M and Johnson N S, Remembrance of things parsed: Story structure and recall, *Cognitive Psychology* 9 (1977) 111-151.
- [8] Black J B and Wilensky R, An evaluation of story grammars, *Cognitive Science* 3 (1979) 213-230.
- [9] Wilensky R, Story grammar versus story points, *The Behavioral and Brain Sciences* 6 (1983) 579-623.
- [10] Garnham A, What's wrong with story grammars, *Cognition* 15 (1983) 145-154.

- [11] Pemberton L, A modular approach to story generation, in: 4<sup>th</sup> European ACL (Manchester UK, 1989) 217-224.
- [12] Kant I, Genius Gives the Rules, in: Rothenberg A and Hausman C R, Eds., *The Creative Question*, (Duke University Press, Durham NC, 1976) 37-42.
- [13] Rothenberg A and Hausman C R, *The Creative Question* (Duke University Press, Durham NC, 1976).
- [14] Boden M A, Could a robot be creative—and would we know?, in: Ford K M, Glymour C. and Hayes P J, Eds., *Android Epistemology*, (AAAI Press/The MIT Press, 1995) 51-72.
- [15] Sharples M, *How we write: Writing as Creative Design*, Routledge, London, 1999.
- [16] Boden M A, *The creative mind*, Abacus, London, 1992.
- [17] Bringsjord S, and Ferrucci D A, *Artificial Intelligence and Literary Creativity. Inside the Mind of BRUTUS, a Storytelling Machine*, Lawrence Erlbaum Associates, Hillsdale NJ, 2000.
- [18] Laird J, Newell A and Rosenbloom P, SOAR: An architecture for general intelligence, *Artificial Intelligence* 33 (1987) 1-64.
- [19] Charniak E, and McDermott D, *Introduction to Artificial Intelligence*, Addison-Wesley, Reading MA, 1985.
- [20] Turner S R, *The Creative Process: A Computer Model of Storytelling*, Lawrence Erlbaum Associates, Hillsdale NJ, 1994.
- [21] Turner S R, *MINSTREL: A computer model of creativity and storytelling*, PhD Dissertation, University of California LA, 1993.
- [22] Pérez y Pérez R, *MEXICA: a Computer Model of Creativity in Writing*, PhD thesis, University of Sussex, 1999.
- [23] Pérez y Pérez R and Sharples M, MEXICA: A computer model of a cognitive account of creative writing, *Journal of Experimental and Theoretical Artificial Intelligence* 13 (2001) 119-139.
- [24] Bremond C, La lógica de los posibles narrativos (trad.), in: *Análisis Estructural del Relato*, (Ediciones Coyoacán, México DF, 1996) 99-122. (Originally published as *La logique des possible narratifs*, *Communications* 8, 1966, 60-76).
- [25] Casebourne I, The Grandmother Program: A Hybrid System for Automated Story Generation, in: Candy L and Edmonds E, Eds., *Proceedings of the Second International Symposium of Creativity and Cognition* (Loughborough, England, 1996) 146-155.



<b>Operation mode</b>	<b>Filtering Process active</b>	<b>Guidelines active</b>
Engaged State 1 (E1)	no	no
Engaged State 2 (E2)	yes	yes (guidelines are set with pre-defined values that never change)
Engaged and Reflective States 1 (ER1)	no	no
Engaged and Reflective States 2 (ER2)	yes	yes (guidelines are always updated during the Reflective State)

**Table 4.1. Operation modes in MEXICA.**

	<b>ER2</b>	<b>Human</b>	<b>MINSTREL</b>	<b>ER2-boring</b>	<b>E2</b>	<b>E2-boring</b>	<b>GESTER</b>
<b>Narrative flow</b>	3.8	3.5	2.9	3.5	2.2	2.8	2.1
<b>Narrative structure</b>	3.7	3.7	3.2	3.2	2.6	2.7	2.1
<b>Content</b>	4.1	3.7	3.6	2.8	2.8	2.4	2.6
<b>Suspense</b>	3.8	3.8	3.3	2.3	2.3	2.0	2.1
<b>Overall quality</b>	3.8	3.6	3.3	2.9	2.6	2.5	2.4

**Table 4.2. Results of MEXICA’s evaluation on a 5 point scale from ‘very poor’ to ‘very good’.**

	<b>MINSTREL</b>	<b>MEXICA</b>	<b>BRUTUS</b>
<b>Evaluates novelty</b>	√	√	X
<b>Evaluates interestingness</b>	X	√	X

**Table 5.1. Comparison of evaluation routines.**

	<b>MINSTREL</b>	<b>MEXICA</b>	<b>BRUTUS</b>
<b>Structure-predictability</b>	yes	no	yes
<b>Content-predictability</b>	no	no	yes

**Table 5.2.** Story predictability in the three systems.

**Figure 2.1 A story developed by BRUTUS.**

**Figure 2.2 A representation of a story-frame.**

**Figure 2.3 A representation of a story-theme.**

**Figure 3.1. A story developed by MINSTREL.**

**Figure 4.1. A story generated by MEXICA (text in italics shows those parts produced in ‘reflection’ mode).**

**Figure 4.2. Tensional Representation of a given story.**

**Figure: 2.1**  
**Journal: Knowledge-Based Systems.**  
**Authors: Rafael Pérez y Pérez and Mike Sharples.**

Dave Striver loved the university. He loved its ivy-covered clocktowers, its ancient and sturdy brick, and its sun-splashed verdant greens and eager youth. He also loved the fact that the university is free of the stark unforgiving trials of the business world —only this isn't a fact: academia has its own tests, and some are as merciless as any in the marketplace. A prime example is the dissertation defense: to earn the PhD, to become a doctor, one must pass an oral examination on one's dissertation.

Dave wanted desperately to be a doctor. But he needed the signatures of three people on the first page of his dissertation, the priceless inscription which, together, would certify that he had passed his defense. One the signatures had to come from Professor Hart. Well before the defense, Striver gave Hart a penultimate copy of his thesis. Hart read it and told Striver that it was absolutely first-rate, and that he would gladly sign it at the defense. They even shook hands in Hart's book-lined office. Dave noticed that Hart's eyes were bright and trustful, and his bearing paternal.

At the defense, Dave thought that he eloquently summarized Chapter 3 of his dissertation. There were two questions, one from Professor Rodman and one from Dr. Teer; Dave answered both, apparently to everyone's satisfaction. There were no further objections. Professor Rodman signed. He slid the tome to Teer; she too signed, and then slid it in front of Hart. Hart didn't move. "Ed?" Rodman said. Hart still sat motionless. Dave felt slightly dizzy. "Edward, are you going to sign?" Later, Hart sat alone in his office, in his big leather chair, underneath his framed PhD diploma.

**Figure: 2.2**  
**Journal: Knowledge-Based Systems.**  
**Authors: Rafael Pérez y Pérez and Mike Sharples.**

```
story-frame
  name:  evilGoal
  type:  character goal
  agent: Hart
  plan:  {lie_to_Striver, refuse_to_sign_his_thesis}
```



**Figure: 2.3**  
**Journal: Knowledge-Based Systems.**  
**Authors: Rafael Pérez y Pérez and Mike Sharples.**

Evil\_Goal is a structure of type *character goal*, whose slot *plan* is employed to instantiate Evil\_Plan and whose slot *agent* is employed to instantiate Betrayer, and Lying is a structure of type *lie* included in Evil\_Plan, and Thwarting is a structure of type *thwart* included in Evil\_Plan, and Betrayed\_Goal is instantiated with the value of the slot *goal\_to\_prevent* from the structure Thwarting, and Betrayers\_lie is instantiated with the value of the slot *theme of the lie* from the structure Lying, and Betrayed is a structure of type *person*, whose slot *goals\_to\_achieve* includes Betrayed\_goal and whose slot *beliefs* includes Betrayers\_lie.



**Figure: 3.1**  
**Journal: Knowledge-Based Systems.**  
**Authors: Rafael Pérez y Pérez and Mike Sharples.**

Once upon a time there was a lady of the court named Jennifer. Jennifer loved a knight named Grunfeld. Grunfeld loved Jennifer.

Jennifer wanted revenge on a lady of the court named Darlene because she had the berries which she picked in the woods and Jennifer wanted to have the berries. Jennifer wanted to scare Darlene. Jennifer wanted a dragon to move towards Darlene so that Darlene believed it would eat her. Jennifer wanted to appear to be a dragon so that a dragon would move towards Darlene. Jennifer drank a magic potion. Jennifer transformed into a dragon. A dragon moved towards Darlene. A dragon was near Darlene.

Grunfeld wanted to impress the king. Grunfeld wanted to move towards the woods so that he could fight a dragon. Grunfeld moved towards the woods. Grunfeld was near the woods. Grunfeld fought a dragon. The dragon died. The dragon was Jennifer. Jennifer wanted to live. Jennifer tried to drink a magic potion but failed. Grunfeld was filled with grief.

Jennifer was buried in the woods. Grunfel became a hermit.

**Figure: 4.1**

**Journal: Knowledge-Based Systems.**

**Authors: Rafael Pérez y Pérez and Mike Sharples.**

*Jaguar\_knight was an inhabitant of the great Tenochtitlan. Princess was an inhabitant of the great Tenochtitlan. From the first day they met, Princess felt a special affection for Jaguar\_knight. Although at the beginning Princess did not want to admit it, Princess fell in love with Jaguar\_knight. Princess respected and admired Artist because Artist's heroic and intrepid behaviour during the last Flowery-war. For long time Jaguar\_knight and Princess had been flirting. Now, openly they accepted the mutual attraction they felt for each other. Jaguar\_knight was an ambitious person and wanted to be rich and powerful. So, Jaguar\_knight kidnapped Artist and went to Chapultepec forest. Jaguar\_knight's plan was to ask for an important amount of cacauatl (cacao beans) and quetzalli (quetzal) feathers to liberate Artist. Princess had ambivalent thoughts towards Jaguar\_knight. On one hand princess had strong feelings towards Jaguar\_knight but on the other hand Princess abominated what Jaguar\_knight did. Suddenly, the day turned into night and after seconds the sun shone again. Princess was scared. The Shaman explained to Princess that Tonatiuh (the divinity representing the sun) was demanding Princess to rescue Artist and punish the criminal. Otherwise Princess's family would die. Early in the Morning Princess went to Chapultepec forest. Princess thoroughly observed Jaguar\_knight. Then, Princess took a dagger, jumped towards Jaguar\_knight and attacked Jaguar\_knight. Jaguar\_knight was shocked by Princess's actions and for some seconds Jaguar\_knight did not know what to do. Suddenly, Princess and Jaguar\_knight were involved in a violent fight. In a fast movement, Jaguar\_knight wounded Princess. An intense haemorrhage arose which weakened Princess. Jaguar\_knight felt panic and ran away. Thus, while Tlahuizcalpantecuhtli (the god who affected people's fate with his lance) observed, Princess cut the rope which bound Artist. Finally, Artist was free again! Princess was emotionally affected and was not sure if what Princess did was right. Princess was really confused. The injuries that Princess received were very serious. So, while praying to Mictlantecuhtli (the lord of the land of the dead) Princess died.*

**Figure: 4.2**  
**Journal: Knowledge-Based Systems.**  
**Authors: Rafael Pérez y Pérez and Mike Sharples.**

