Chapter IV

How MEXICA Works

The system can be described in terms of two main processes: the first creates all data-structures in long-term memory and the second, based on these structures, creates new stories.
The first structure created in long-term memory is called Primitive Actions Structure; it encodes all the information defined in the file of Primitive Actions. The rest of the structures are created by transforming the file of Previous Stories into the Concrete, Abstract and Tensional Representations. The Concrete Representation can be seen as a copy in memory of the file of Previous Stories. The Abstract Representation encodes the knowledge necessary to bring into working memory plausible actions to continue the story in progress. Finally, the Tensional Representation registers the behaviour of a variable called Tension to the Reader, which is used to produce interesting stories. Both, the Primitive Actions and the Previous Stories are defined (i.e. built) by the user.

The process of developing new stories consists of a cycle between the Engaged and Reflective States. During engagement, an action is performed producing a specific story-world context. Such a context is used to match in long-term memory structures representing similar situations. These structures have associated a set of possible next actions, which are brought into working memory. There, one of them is selected as the next action in the story. This action is performed in the story producing a new story-world context and the cycle starts again.
If the cycle is interrupted (e.g. by an impasse) the system switches to the Reflective State. During reflection all preconditions are verified, impasses broken, and the material produced is evaluated either to set the guidelines and return to the Engaged State or to finish the story. When the story is finished, MEXICA performs a final analysis to add a few finishing touches to the tale.
In this chapter, all these processes are explained in detail.

4.1 Primitive Actions.

The design of the system is based on operators, preconditions and effects. In MEXICA operators are called Primitive Actions; thus, Primitive Actions are a set of actions that any character can perform in the story and whose consequences produce some change in the story-world. Primitive Actions are defined by the user in a text file by specifying the name of the action, the number of characters who participate in it and the set of preconditions and effects associated with the action. A complete explanation of the syntax to define Primitive Actions can be found in Appendix A The maximum number of characters allowed in one action are three, and most of them follow the structure *actor1 acts-on actor2*, i.e. actor1 executes an action towards actor2, e.g. EAGLE_KNIGHT ATTACKED ENEMY or PRINCESS CURED JAGUAR_KNIGHT. However, actions in which actors act on each

other can also be defined, e.g. EAGLE_KNIGHT FOUGHT ENEMY (both characters are fighting to each other). Actions involving three characters are special cases and they are explained in section 4.2. When defining actions, letters are used to represent characters. For example, the following string represents the action of character A kissing B:

ACTION
A KISSES B

All the primitive actions used to test the present model have been defined in the past tense, e.g. A LOVED B, A WENT_TEXCOCO_LAKE, (a list with some of the Primitive Actions used to test the prototype can be found in Appendix C). The user is free to define the primitive actions in the most convenient way since the internal representation and instantiation process are independent of such definition.

Each time the program starts MEXICA creates in long-term memory the Primitive Actions Structure. Such a structure contains all the information the user defined in the file of Primitive Actions.

4.1.1 Postconditions.

The effects that Primitive Actions produce in the story-world are known as postconditions. There are three types of possible postconditions that can be used in MEXICA:
1) Emotional Links between characters.
2) Tensions in the story.
3) Changes in the physical position of the characters in the story-world.
Each of these is now described in detail.

4.1.1.1 Emotional Links.

It is beyond the scope of this work to make a detailed analysis of emotions. However, they are essential elements in any short story. For the purpose of implementing the present model, a simple representation of Emotional Links between characters has been incorporated in the system. Dyer (1987) describes some computer models of emotions. He affirms that “all affects can be represented simply in terms of a positive or negative state of arousal...” Based on that idea in MEXICA three types of Emotional Links have been defined.

For practical reasons all types of emotions are implemented in discrete terms with a value in the range of -3 to +3. Type 1 represents a continuum between love and hate. In this way, an emotion of intensity -3 and type 1 represents hate, while an emotion of intensity +3 and type 1 represents love. The remainders are possible values between these two poles.

To hate						To love
-3	-2	-1	0	+1	+2	+3

Emotions of type 2 represents a continuum between being in love with and feeling hatred towards. In the present work, love is interpreted as brotherly love, while to be in love is interpreted as amorous love. Finally, type 3 has been defined with the intention of allowing the user to include another Emotional Link according to his/her necessities. That is, when defining Primitive Actions the user can include an Emotional Link of type 3; it will represent what the user requires.

The following example illustrates the use of Emotional Links as postconditions. If a warrior is ill or wounded and a lady cures him, it is natural that he will develop very positive emotions towards the lady. In this way, when defining the action A CURED B, postconditions might include that B develops an Emotional Link of type 1 intensity +3 towards A.

ACTION
A CURED B
POSTCONDITIONS
B(+3,1): A

Notice in the last example the notation used to represent the postcondition B developed an Emotional Link of intensity +3 and type 1 towards A. From now onwards, the same notation will be used in all the examples involving Emotional Links: first it is specified the character who develops the Emotional Link, followed in brackets and separated by a coma by the intensity and type of Emotional Link, followed by a colon and ending with the character towards whom the link is developed[1].

Postconditions also allow establishing Emotional Links between characters participating in an action and their friends or family in the story, i.e. between characters participating in an action and others characters who previously have established a positive Emotional Link towards them. This second group of characters (friends, family, etc.) is referred as Linked Characters. In order to establish such an Emotional Link it is necessary that the Linked Characters are located in the same position as the actors in the event. For example, an action like PRIEST CURED JAGUAR_KNIGHT causes not just jaguar knight but all his friends (i.e. all Linked Characters) to develop a positive Emotional Link towards the priest. This situation is defined in the declaration of primitive actions as follows:

ACTION
A CURED B
POSTCONDITIONS
B(+3,1): A
LB(+3,1):A

Where LB represents all those characters linked previously to character B. In this way, the second postcondition can be read as all those characters who have a positive Emotional Link towards character B and are in the same location as character B, also establish an Emotional Link of intensity +3 and type

1 towards character A. Finally, MEXICA allows the user to introduce a variation of this situation. If the priest cures jaguar knight, the intensity of the emotional response of jaguar knight's friends towards the priest depends on how strong their relationship is. That is, jaguar knight's fiancé would establish a stronger emotional response than someone who is just fond of him. This situation can be represented in MEXICA. If instead of specifying a concrete intensity for Linked Characters the symbol of percentage is used, e.g. LB (%,1):A, then the intensity of the Emotional Link triggered by the postcondition between LB (Linked Characters) and the character A depends on the intensity of the Emotional Link between LB and the character B, and the intensity of the Emotional Link between character B and character A. In this way, if character B is very grateful towards character A, someone who is just fond of character B establishes a less intense Emotional Link towards character A that someone who is in love with character B. This situation is defined in terms of primitive actions as follows:

ACTION
A CURED B
POSTCONDITIONS
B(+3,1): A
LB(%,1):A

MEXICA uses a table, called Percentage Table, to obtain the value of the intensity of the Emotional Link when it is defined in terms of the percentage symbol.

LB->B	**B->A**	**LB->A**
+3	-3	-3
+3	-2	-2
+3	-1	-1
+3	0	0
+3	+1,+2+3	(B->A)-1
+2	-3,-2	-2
+2	-1	-1
+2	0,+1	0
+2	+2,+3	(B->A)-1
+1	-3,-2,-1	-1
+1	+3	+1
+1	0,+1,+2	0

(B->A)-1 means to decrement by one the value of the intensity between character B and character A.

The table shows how the value of the intensity of the Emotional Links between LB and character A depends on the value of the intensity of the Emotional Link between LB and character B, and the value of the intensity of the Emotional Link between the character B and the character A. Also, it is important to notice that the value of such intensity does not depend on the type of Emotional Link established.

[1] Although the notation used to define Emotional Links in the prototype differs from the one in the example, for reasons of clarity the second has been adopted through this text.

As an example to illustrate the use of the Percentage Table the reader can imagine a princess is in love with jaguar knight, i.e. she has established the following Emotional Link: princess(+3, 2):Jaguar_Knight. Now, if the jaguar knight has an accident and the priest cures him, according to the primitive action defined above, two new Emotional Links are established: Jaguar_Knight(+3,1):priest, and princess(%,1):priest. Then, ensuing from the Percentage Table the following value is obtained:

princess->Jaguar_Knight	Jaguar_Knight->priest	princess->priest
+3	+3	(+3)-1 = +2

In this way the princess establishes the Emotional Link princess(+2,1):priest which makes sense since probably she is very grateful to the priest for saving the life of her lover.

MEXICA allows characters to establish different Emotional Links towards each other; e.g., character B can have an Emotional Link of intensity +3 type 2 towards character A and at the same time have an Emotional Link of intensity -2 type 1 towards such a character A. In this case, if MEXICA is calculating the intensity of an Emotional Link defined in terms of the percentage symbol, and either the previously Linked Characters to B or character B have different Emotional Links towards A, the system needs to decide which of such Emotional Links is the one that will be used to obtain the value of the intensity from the table. The rule is simple: the one with the highest absolute value of intensity is selected. This was decided on the assumption that strong feelings overcome weak ones. If both Emotional Links have the same absolute value, one is selected randomly.
Linked Characters are only allowed in postconditions of type Emotional Links. Notice that Linked Characters are just those characters who have established a positive Emotional Links towards other one.

4.1.1.2 Tensions.

One of the key elements in any short story is the tension produced by the author to the reader through the use of different tools or techniques. It is beyond the scope of this work to analyse tensions and fiction, but it is assumed that a tension in a short story arises when a character is murdered, when the life of a character is at risk, when the health of a character is at risk (e.g. when a character has been wounded), when a character is made a prisoner, when a character feels clashing emotions towards other one, and when two different characters are in love with a third one producing a love competition. Six different types of tension represent these situations: Actor Dead, Life at risk, Health at risk, Prisoner, Clashing Emotions and Love Competition. It is not suggested that those are the only situations that can produce tension in a reader, or even that always these situations will produce it. They have just been defined in order to be able to implement and test the present computer model.
There are two different ways to trigger tensions. Actor Dead, Life at risk, Health at risk and Prisoner are activated directly by postconditions. For example, if in a story the enemy attacks the tlatoani, a

tension will be produced because the life of the tlatoani is at risk. This situation can be defined in terms of Primitive Actions as follows:

ACTION
A attacked B
POSTCONDITIONS
Lr (B):A+

Notice again the way the postcondition indicating that the life of character B is at risk due to the presence of character A has been defined. From now onwards this notation will be used in all examples involving tensions: first a mnemonic representing the tension to be triggered is specified, followed in brackets by the character who is the victim, followed optionally by a colon and the character who is responsible of triggering the tension, followed optionally by the addition symbol. The addition symbol indicates that the tension will be active only while character A and B are in the same location; it would not make sense in a short story to keep the tension Life at Risk active if character A attempts to attack character B and then s/he runs away.

When the addition symbol is omitted the tension triggered by the postcondition is kept active, whether or not A and B are in the same location, until other postcondition deactivate it. For example, if character A pushes character B to a river, the life of character B is at risk regardless if character A runs away or not. Finally, when the second character is excluded from the definition of the postcondition, e.g. Lr(B), it is assumed that nobody is responsible that the life of the character B is at risk and the tension is kept active until other postcondition deactivates it. To illustrate this case the reader can picture an event where character B falls accidentally in a river. The mnemonics used to represent tensions are: Ad for Actor Dead, Lr for Life at Risk, Hr for Health at Risk and Pr for Prisoner. Appendix C shows examples of declarations of Primitive Actions.

The second method to trigger tensions is now explained. Clashing Emotions and Love Competition belong to a special class of postconditions known as Inferred Postconditions. They arise when MEXICA detects that the conditions that activate them are present in the story. Thus, each time an action is performed in the story MEXICA verifies if a tension must be triggered. A tension due to Clashing Emotions is produced when a character establishes two Emotional Links of any type but opposite intensity towards other character. To illustrate this situation, the reader can imagine a princess who falls in love with an enemy. A tension due to love competition is produced when two different characters are in love with a third one. For example, consider a story with the following actions:

EAGLE_KNIGHT WAS_IN_LOVE_WITH PRINCESS
JAGUAR_KNIGHT WAS_IN_LOVE_WITH PRINCESS

The first event triggers the Emotional Link Eagle_Knight (+3,2):princes and the second one activates Jaguar_Knight (+3,2):princess. Then, MEXICA detects that two different characters are in love with the princess and it triggers a tension of Love Competition.

Finally, there are three other tensions: Life Normal, Health Normal and Prisoner Free. They are used to deactivate those tensions triggered directly by postconditions. The mnemonics to identify them trough the examples are Ln, Hn, Pf respectively. In the same way as the previous tensions, the user defines them as consequences in the primitive actions. For example, to deactivate the tension produced when the health of a character is in risk (Hr), the following action can be defined:

ACTION
A CURED B
POSTCONDITIONS
Hn(B):A

4.1.1.3 Positions.

The story-world in MEXICA has eight different positions or locations: Texcoco Lake, Popocatepetl Volcano, Tlatelolco Market, Palace, Tenochtitlan city, Chapultepec Forest, Temple and Jail. Two other special locations have also been defined: Nowhere which is used to locate dead characters, and b_position that is used to situate a character in the same position as another character. In MEXICA all characters that participate in an action must have the same position in the story-world in order to perform such an action.

The set of active postconditions that a character has at any specific time in the story is known as the Context of the character. During the developing of the story each time an action is executed, each character's context is updated. In MEXICA, the Context determines what Emotional Links and Tensions characters are aware of in the story, and plays a crucial role in the generation of material during the Engaged State.

4.1.2 Preconditions.

Preconditions specify the requirements that need to be fulfilled in order that a character can perform an action. However, as explained in Chapter III, preconditions have a very particular characteristic in the present model: they are just checked during the Reflective State. During the Engaged State MEXICA produces material without taking them into account. The user specifies preconditions when Primitive Actions are declared; they can be of two types: Tensions or Emotional Links. There is a predefined or implicit precondition for all Primitive Actions; it requires that when an action is performed, all characters participating in it must be in the same place in the story-world. The following is an example of a declaration of a Primitive Action including preconditions.

ACTION
A CURED B
PRECONDITIONS
Hr(B):*
A(+1,*):B
POSTCONDITIONS
Hn(B):A
B(+3,1):A

Notice that the syntax used to define the tension in the precondition is slightly different to the one used previously. The reason is that preconditions allow using the start symbol ('*') which means "any character in the tale". In this way, the precondition in the example can be read as: In order for A to cure B it was necessary that the health of B was previously put at risk by any other character in the story (this also includes the possibility of a none-character.) There is one important restriction to point out: when a tension is used as a precondition, the star can only be used to represent the character who is responsible for triggering the tension (i.e. the second character in the definition); the victim of the tension (i.e. the first character in the definition) cannot be represented by a star.
The star symbol has also another function. When it is used in the definition of an Emotional Links as a preconditions, the star stands for "any type of Emotional Link"; so, the precondition in the example can be read as: In order for A to cure B it was necessary that A has an Emotional Link of any type towards B.

Preconditions represent the requirements that must be satisfied in order an action can be performed. In the present research it is assumed that if an Emotional Link is defined as a precondition, the same Emotional Link but with a stronger intensity also matches the precondition. That is, if the precondition for an action is A(+1,*):B, the precondition is satisfied if the character A has an Emotional Link of intensity +1, +2 or +3 of any type towards B. If the precondition for the action is A(-2,*):B, it is fulfilled if the character A has an Emotional Link of intensity -2 or -3 of any type towards B. In this way, the range of valid values goes from the value specified in the precondition to +3 or -3.
So, in the example of the definition of the action A cured B, the range of valid values goes from +1 to +3. In this way, the precondition can finally be read as: In order for A to cure B it was necessary that A has an Emotional Link of any type and intensity greater or equal to +1 towards B.
It is important to stress that this interpretation of the intensity just applies for Emotional Links as preconditions; under other circumstances it would make no sense.

During the Reflective State MEXICA verifies that all preconditions in the new story have been fulfilled, and, if necessary, actions are inserted in the new story in order to fulfil them. Only those characters participating in the action are allowed intervening in the preconditions; so, Linked Characters are not permitted.

4.1.3 Text Generation.

The words and phrases that form a story are a vital part of it. However, it is out of the scope of this work to make any research related to natural language generation. Thus, in order to produce better outputs, MEXICA includes a routine that permits linking text to Primitive Actions. In this way, it is possible to embellish the stories produced by the system. The method works as follows. The user defines as part of the Primitive Actions a text related to the event it represents. When the story is finished, all Primitive Actions are substituted by their texts to generate the final version of the tale. The following is an example of a Primitive Action including the text option:

ACTION
A CURED B
PRECONDITIONS
Hr(B):*
A(+1,*):B
POSTCONDITIONS
Hn(B):A
B(+3,1):A
TEXT
With the help of some magic plants @A cured @B.

The text is formed by a string of characters and three possible variables symbolised as @A, @B and @C. These variables represent characters A. B and C in the Primitive Action. Thus, if the action in this example is instantiated as PRINCESS CURED JAGUAR_KNIGHT, the variable @A is substituted by PRINCESS and the variable @B by JAGUAR_KNIGHT resulting in the following text: “With the help of some magic plants princess cured Jaguar_knight.” MEXICA permits defining any number of texts in a single Primitive Action and when the story is finished, the system selects the text to be used at random between the available options. If no text is defined MEXICA employs the name of the Primitive Action; e.g., if no text is defined for A CURED B, MEXICA includes in the final version of the story the phrase “princes CURED Jaguar_knight”.

4.2 Previous Stories.

Previous stories provide the basic material from which all new stories are created. Consequently, any outcome from MEXICA depends on their number and the way they are organised. In MEXICA, the Previous Stories are represented as a set of short stories. The following is an example:

STO
EAGLE_KNIGHT ACTOR
JAGUAR_KNIGHT ACTOR
EAGLE_KNIGHT WAS_IN_LOVE_WITH PRINCESS
JAGUAR_KNIGHT WAS_IN_LOVE_WITH PRINCESS
EAGLE_KNIGHT WERE_FRIENDS HUNTER
HUNTER HATED PRINCESS
PRINCESS WENT_TEXCOCO_LAKE

HUNTER FOLLOWED PRINCESS
HUNTER KILLED PRINCESS
JAGUAR_KNIGHT WAS_TOLD EAGLE_KNIGHT KILLED PRINCESS
JAGUAR_KNIGHT KILLED EAGLE_KNIGHT
HUNTER REALISED JAGUAR_KNIGHT KILLED EAGLE_KNIGHT
HUNTER LOOKED_FOR_AND_FOUND JAGUAR_KNIGHT
HUNTER KILLED JAGUAR_KNIGHT
END

The user defines in a text file all previous stories. The syntax to define them is explained in Appendix B. Notice from this example how a story can be understood as a chain of primitive actions, and how the first two actions have neither preconditions nor postconditions; they are used just to introduce the characters in the story. Their real use will become evident through the following sections. Notice as well the events JAGUAR_KNIGHT WAS_TOLD EAGLE_KNIGHT KILLED PRINCESS and HUNTER REALISED JAGUAR_KNIGHT KILLED EAGLE_KNIGHT. These are special kind of primitive actions called Compound Actions; their main purpose is, for the former, to make characters to believe things which have not occurred in the tale, and for the latter, to allow characters to realise about other events happening in the story.

Previous Stories are transformed into structures in long-term memory, which are used to produce new tales. This process is done action by action; i.e. MEXICA reads an action from the text file, updates all characters' contexts in working memory, updates long-term memory, and then reads the following action in the file and repeats the same cycle until all pervious stories are processed.
Although all actions in the file are read in sequence, when MEXICA detects that a new story is starting it cleans all characters' contexts and resets variables, i.e., each of the Previous Stories is processed independently.

4.2.1 Introducing New Characters in the Story.

Characters become alive in the story the first time they are mentioned, i.e. the first time they participate in a Primitive Action. Their location is set according to the following rule: when all characters participating in an action are introduced for the first time in the story (i.e. they have just becoming alive), they are placed in a default location chosen previously by the user. When at least one of the characters in the action is already alive in the tale, new characters are situated in the same position as the one alive.

In this way, if for example the first action in a story is WARRIOR WAS_IN_LOVE_WITH LADY and the default location is set by the user to Texcoco Lake, MEXICA assigns a context to the warrior and to the lady and sets their location to the default value Texcoco Lake. If the next action is EAGLE_KNIGHT WAS_IN_LOVE_WITH LADY, a context is assigned to the new character eagle knight and he is located in the same place as lady, i.e. at the lake.

Although the three characters are in the same location, because a character becomes alive in the story the first time s/he is mentioned, eagle knight is not aware of the warrior's Emotional Link towards the princess. However, the warrior and princess do know about eagle knight's Emotional Links towards her, and that triggers in their context as an inferred postcondition a tension due to love competition (see Section 4.2.2.2 for inferred postconditions). This situation is represented in the following lines in terms of characters' contexts.

CONTEXT ***
***Time=>1 Action: warrior WAS_IN_LOVE_WITH lady
Character:WARRIOR. Pos:Texcoco Lake. Status:Alive
Emotional Links => Warrior(+3,2):princess.
Tensions=> none.

Character:LADY. Pos: Texcoco Lake. Status:Alive
Emotional Links=> Warrior(+3,2):princess.
Tensions=> none.

CONTEXT ***
***Time=>2 Action: eagle_knight WAS_IN_LOVE_WITH lady
Character:WARRIOR. Pos:Texcoco Lake Status:Alive
Emotional Links=> Warrior(+3,2):princess
Eagle_Knight(+3,2):princess
Tensions=> Lc(Warrior):Eagle_Knight

Character: LADY Pos: Texcoco Lake Status:Alive
Emotional Links=> Warrior(+3,2):princess
Eagle_Knight(+3,2):princess
Tensions=> Lc (Warrior):Eagle_Knight

Character:EAGLE_KNIGHT Pos:Texcoco Lake Status:Alive
Emotional Links=> Eagle_Knight(+3,2):princess.
Tensions=> none.

If this situation is not convenient for the story, it can be avoided by including as a first action EAGLE_KNIGHT ACTOR. The Primitive Action ACTOR has neither preconditions nor postconditions; its only function is to introduce a character to the tale. Thus, what it produces is that eagle knight is assigned a context when the story starts and then from the beginning he is able to become aware of all the events occurring around him.

4.2.2 Updating Working Memory.

Working memory is updated in two main stages: 1)The preconditions triggered by the action performed are joined to characters' contexts; 2)The consequences that such preconditions produce in characters' contexts (i.e. inferred postconditions) are analysed and updated.

4.2.2.1 Joining Postconditions.

The process of joining postconditions to characters' contexts involves three steps: 1)Getting the characters participating in the action, 2)Getting the postconditions that the action triggers and 3)Joining such postconditions to the characters' context. So, each time MEXICA reads an action from previous experiences, it obtains and transforms to a suitable representation the name of the action and the characters participating in it. These characters are denominated the Actors in the Action. If any of these characters does not exist in the story, a context is assigned to them. After this, MEXICA obtains a copy of the postconditions and instantiates them with the actors in the action. Finally, the postconditions are joined to the characters' contexts.

Postconditions always have a main or principal actor:

-When the postcondition is of type Position, the main actor is the character who changes location.

-When the postcondition is of type Emotional Link, the main actor is the character who develops or establishes the link towards other one (e.g. in the case of princess(+3,2):Eagle_Knight, the main character is the princess).

-When the postcondition is of type Tension, the main actor is the character who is the victim in the situation (e.g. for Lr(princess):enemy, the main actor is the princess).

This concept of main actor is important when updating characters' contexts.

Thus, depending on the type of postcondition to be added to the context, MEXICA runs one of three possible routines: Joining Position (change in the location), Joining Emotional Links or Joining Tensions.

A. Joining Position.

The field Position in the main actor's context is set either to a specific location (e.g., Texcoco Lake, Tenochtitlan City, etc.) or to the location of the second character in the action (b_Pos) depending on what was specified in the postcondition. The second option (b_Pos) is useful for actions like A followed B, where the postcondition specifies that the main actor (character A) must change his/her position to the position of the second character in the action (character B).

B. Joining Emotional Links.

This process consists of joining the list of Emotional Links triggered by the action to the context of all those characters in the story who are in the same location that the characters in the action. For example, if a tale has three characters, e.g. jaguar knight, princess and eagle knight, they all have the same location, and the action JAGUAR_KNIGHT FELL_IN_LOVE_WITH PRINCESS occurs, the postcondition Jaguar_Knight(+3,2):princess is joined to the context of the three of them (including Eagle_Knight who did not participate in the action).

There are two possible variants to this process. When the action performed is compound (compound actions are REALISED or WAS_TOLD, see section 4.2), the Emotional Links are added just to the

context of the character who is performing it; e.g. with the action JAGUAR_KNIGHT REALISED PRIEST CURED PRINCESS, all the Emotional Links triggered by the action CURED are added just to the context of the character jaguar knight.

The second variation of the process occurs when the main actor in the postcondition is a Linked Character (see Linked Characters in Emotional Links in section 4.1.1.1). In this case, MEXICA looks in the context of all those characters who have the same location that the actors in the action, for the Linked Characters. When MEXICA finds a context with such Linked Characters, it joins the postcondition to that context. For instance, in the example some lines above where JAGUAR_KNIGHT FELL_IN_LOVE_WITH PRINCESS, jaguar knight is a Linked Character to the princess, and three of them (jaguar knight, princess and eagle knight) are aware of this link. Let us suppose that the action WARRIOR MUGGED PRINCESS occurs. One of the postconditions for such an action is that all characters linked to the princess hate the warrior; so, the Emotional Link Jaguar_Knight(-3,1):Warrior is triggered. This Emotional Link is joined as a postcondition to the context of the jaguar knight, the princess and the eagle knight. That is, because three of them were in the same location, and three of them knew about the link between jaguar knight and the princess, three of them become aware that jaguar knight hates the warrior. However, the warrior does not get that postcondition, i.e. he is not aware that jaguar knight hates him because he did not know about the previous link.

C. Joining Tensions.

This process works as the same as for joining Emotional Links but it does not include Linked Characters. That is, those tensions triggered by the action as postconditions are joined to the context of the characters located in the same place that the actors in the action; for compound actions, tensions are joined to the context of the character performing the compound action.

4.2.2.2 Analysing Consequences (Inference Procedures).

Characters' contexts are very dynamic structures that are integrated, not just for the postconditions triggered by actions, but also for postconditions triggered and/or eliminated by inference procedures. That is, through these procedures MEXICA adds or deletes Emotional Links and/or Tensions to characters' contexts. Members added through this process are known as inferred postconditions. So, after characters' contexts have been updated with the last action in the tale, MEXICA analyses the consequences of such an action (runs inference procedures) in all characters' contexts. This section explains this analysis.

A. Analysing Emotional Links.

There are three potential situations related to Emotional Links that can trigger inferred postconditions. They are Clashing Emotions, Love Competition and Potential Danger. A clashing emotion is produced when a character establishes two or more Emotional Links of any type but opposite sign towards other character. For example, if an enemy kidnaps the princess and later he risks his life to save her from the attack of a tiger, she will probably develop a very negative feeling from the first action, and at the same time a very positive from the second one. When MEXICA detects that both feelings are present, an inference process triggers a tension due to clashing emotions which is joined to the context of all characters who are aware of the dual emotion.

A love competition is produced when MEXICA detects that two different characters have an Emotional Link of type 2 intensity +3 (which represents to be in love with) towards the same character. For example, if the tlatoani and the eagle knight fall in love with the same lady, a tension due to love competition is triggered and joined to the context of those characters that are aware of this situation. As explained in the example in section 4.2.1, if (for example) the tlatoani is not aware that the eagle knight is also in love with the lady, the tension due to love competition is just joined to the context of the eagle knight.

A tension due to potential danger is triggered when a character establishes an Emotional Link of any type and intensity -3 towards other one, and both of them are in the same location. Again, the tension is just joined to the context of those characters who are aware of the conditions to trigger the tension.

As part of the inference processes and in order to avoid an unmanageable amount of Emotional Links between characters, MEXICA simplifies them. It works as follows. An Emotional Link is classified by its type and by the sign of its intensity (positive or negative). Thus, if two characters have established two or more Emotional Links of the same type and the same sign, just the one with the highest intensity is kept and the rest are eliminated. This is done under the assumption that strong feelings override weak ones when they belong to the same type, e.g. if character A is fond of B but later character A falls in love with character B, the second Emotional Link overrides the first one.

B. Analysing Tensions.

For every action performed in a tale, MEXICA corroborates that the conditions which keep tensions active are still present in the context; if an action modifies such conditions, the tensions are deleted. There are four situations that need to be verified:

1)Presence-Conditioned Tensions.- As indicated in Section 4.1.1.2, during the declaration of primitive actions tensions can be conditioned through the use of the symbol '+' to be active while the characters participating in the action are in the same location. If one (or both) of the characters changes his/her position or one or both of them die, the tension is deleted from the context of all those characters who knew about it and who are aware that they have changed their location. For example, for the story

PRINCESS WENT_POPOCATEPTL_VOLCANO_WITH JAGUAR_KNIGHT

ENEMY ATTACKED JAGUAR_KNIGHT
PRINCESS RAN_BACK_TO_CITY
JAGUAR_KNIGHT FOUGHT ENEMY
ENEMY RAN_AWAY

the princess was aware that the life of the jaguar knight was at risk due to the presence of the enemy because she was with them during the attack (see the postconditions for A ATTACKED B in Appendix C). However, because she changed her position, when the enemy ran away her context is not updated (the tension triggered by the action ATTACKED is not deleted) and she keeps on thinking that the life of jaguar knight is at risk due to the presence of the enemy. The contexts of jaguar knight and enemy are updated and such a tension is eliminated from the contexts.

2)Normal Tensions.- There is a special group of postconditions used to deactivate tensions triggered previously by actions in the tale (see Section 4.1.1.2). They are called Normal Tensions although they are not really tensions, i.e. instead of incrementing the value of the variable Tension to the Reader they decrement it by deactivating tensions previously triggered by other actions. Normal Tensions are essential in order to produce those processes of degradation and improvement described in section 3.2.1-B. There are three kinds of Normal Tensions: Life-normal (Ln), Health-normal (Hn) and Prisoner-free (Pf). The syntax used to define them as postconditions in a Primitive Action is the same as the one explained before to define tensions. For example, in the action

ACTION
A cured B
POS
Hn(B):A

the postcondition can be read as the health of character B is normal again thanks to character A. This implies that the tension Hr(B):* was active before the action A CURED B was executed. So, when a Normal Tension is triggered MEXICA looks in all characters' contexts (who are in the same position as the characters in the action) for a tension to deactivate. A Normal Tension of type Hn always deactivates a tension of type Hr; Ln always deactivates a tension of type Lr; and Pf always deactivates a tension of type Pr. Once a Normal Tension has deactivated the old tension it also deactivates itself, that is, they are not kept in the characters' context. For this reason Normal Tensions can be used just as postconditions.

3)Dead-character Tensions.- Once this tension is triggered it always remains active; however, it can provoke other tensions to be eliminated. The purpose of Dead-Character Tension is to make others characters aware that someone in the tale died. When a character dies, all the Emotional Links where s/he was the main actor and all tensions where s/he participates are deleted from the context of all characters who are aware of his dead. For example, if jaguar knight and eagle knight were in love with a lady causing a Love Competition tension, and one of them dies, the tension is eliminated. Notice that MEXICA does not eliminate any of the Emotional Links that other characters have towards the dead

actor; that is, although a character is dead her/his friends and family keep their Emotional Link towards such a character.
When a character dies MEXICA also cleans his/her context, i.e. it deletes all Emotional Links and Tensions, sets the status to Dead and position to Nowhere.

4)Love Competition Tensions.- When MEXICA detects a Love Competition tension, it is necessary to corroborate two situations: 1)the characters engaged in the love competition are still alive and both are still in love with the same character; 2)the loved character is still alive, i.e. it is not possible that an Emotional Link towards a dead character produces a Love Competition tension.

5)Clashing Emotions Tensions.- MEXICA verifies that the main character in a tension of Clashing Emotions is still alive and that s/he is still having these opposite Emotional Links towards other character.

6)Potential Danger Tensions.-MEXICA verifies that the main character in a tension of Potential Danger is still having an Emotional Link of any type and intensity -3 towards other character, and that both of them are still in the same location.

MEXICA also has a way of simplifying tensions: If two different actions trigger the same tension between two characters, the system keeps just one of them.

4.2.3 Loading Long-Term Memory.

When MEXICA starts it creates in long-term memory the Primitive Actions Structures and transforms Previous Stories into three different structures: Concrete Representation, Abstract Representation and Tensional Representation.

4.2.3.1 Primitive Actions Structures and Concrete Representation.

The Primitive Actions Structures can be described as a copy in memory of the file of the Primitive Actions created by the user. The Concrete Representation is formed by linking the Primitive Actions Structures in the same way that Previous Stories are organised; i.e., the Concrete Representation can be described as a copy in memory of the text file of Previous Stories.

4.2.3.2 Abstract Representation.

The Abstract Representation is built in two steps. The first consists of generalising the context of the first character participating in the action, i.e. all concrete characters are substituted by variables; this new representation is called atom. During the second step MEXICA searches memory to find out if the

features represented by the new atom have been already encoded by another atom created previously. If that structure already exists the new atom is deleted; otherwise, it is allocated in the memory-area reserved for the Abstract Representation.
The main purpose behind the Abstract Representation is to gather together all those actions in Previous Stories that are preceded by the same context. In this way, using characters' context as a guide, MEXICA can link actions in a coherent way. Different contexts can lead to the same actions; or in other words, there are different paths to arrive to the same action. This seems to be in accordance with the studies of Cognitive Psychology in Elaborations and the alternative retrieval routes they produce (Anderson 1990 p.181).

When MEXICA generalises a context, it replaces all concrete characters in the Emotional Links and Tensions with variables, obtaining a representation very similar to the one used for specifying postconditions. For example, having the tension Lr(Tlatoani):Enemy after it has been generalised it is represented as Lr (A):B, where A stands for tlatoani and B for enemy. So, each character in the story is represented in the atom by a different variable, and the same variable is used through the whole atom to represent the same character.

The following is an example that shows how characters' contexts change through a story and specially how atoms are created. The reader must imagine that MEXICA is processing the file of Previous Stories which just contains the following story.

STO
PRINCE WENT_TEXCOCO_LAKE
PRINCE HAD_AN_ACCIDENT
PRIEST FOUND_BY_ACCIDENT PRINCE
PRIEST REALISED PRINCE HAD_AN_ACCIDENT
PRIEST CURED PRINCE
PRINCE WENT_PALACE
FISHERMAN MUGGED PRIEST
PRINCE REALISED FISHERMAN MUGGED PRIEST
PRINCE LOOKED_FOR_AND_FOUND FISHERMAN
PRINCE MADE_PRISIONER FISHERMAN
END.

The characters' contexts for the first three actions are[2]:

*** CONTEXT`S REPORT STORY No.1
*** Time => 1 Action: prince WENT_TEXCOCO_LAKE
Charac: PRINCE Pos: Texcoco Lake Status: Alive

*** Time => 2 Action: prince HAD_AN_ACCIDENT
Charac: PRINCE Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

[2] For reasons of clarity, some of the information produced by the Context's Report has been suppressed or modified in this example.

*** Time => 3 Action: priest FOUND_BY_ACCIDENT prince
Charac: PRINCE Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

Charac: PRIEST Pos: Texcoco Lake Status: Alive

The first action just introduces the character prince in the story; his context includes neither Emotional Links nor tensions. The second action's postcondition triggers a tension of type Life at risk which is registered in the prince's context. The third action introduces the priest in the tale. Notice that, because the priest was not in the same location as the prince was when he had his accident (as a matter of fact the priest did not exist in the tale at that moment), he is not aware that the life of the prince is at risk. However, the compound action at time 4 makes the priest to realise about this situation and it is registered in his context

*** Time => 4 Action: priest REALISED prince HAD_AN_ACCIDENT
Charac: PRINCE Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

Charac: PRIEST Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

Now, let us examine the atoms. Because the prince's context in action one does not include any Emotional Link or Tension, no atom is created from it. Prince's context in action number two produces the following atom:

** Atom
Tensions=> Hr(a)
Emotional Links=> NIL
Possible Next Actions=> NIL

Notice that the character “a” substitutes the character prince. The Abstract Memory is empty and MEXICA allocates the atom in it. Notice that the list of possible next actions points to nil; that is, MEXICA has not read yet action three from the file of Previous Stories, so it cannot assign a value to such a list. To solve the problem, MEXICA keeps the address in memory of such an atom; then, when it reads action three, the list of possible next actions in the atom is updated with the right action.

** Atom
Tensions=> Hr(a)
Emotional Links=> NIL
Possible Next Actions=> b FOUND_BY_ACCIDENT a

Notice how the character prince has been substituted by the letter "a" in the representation of the tension and in the possible next actions. So, there is a correlation between the characters in the Emotional Links and Tensions, and the characters in the list of Possible Next Actions. This correlation is important in order to obtain the correct interpretation of the atom. For instance, the atom in this

example encodes that when the health of a character in a story is at risk, a coherent possible next action to happen is that someone else finds by accident such a character.

Action number three introduces the priest into the story and locates him by the prince. Notice that this action does not alter prince's context. Now, MEXICA follows the process described earlier to update the Abstract Representation: it generalises prince's context and looks in memory for an atom equal to it. Because the context did not change between actions two and three, MEXICA finds that the only atom in memory is equal to the one just created. So, it deletes the new atom and again keeps the address in memory of the old atom to updated it when the next action is read.
Because the context of the priest does not include any Emotional Link or Tension, no atom is created from it. When action four is read, MEXICA updates the list of possible next actions in the atom.

*** Atom
Tensions=> Hr(a)
Emotional Links=> NIL
Possible Next Action=>
b FOUND_BY_ACCIDENT a
b REALISED a HAD_AN_ACCIDENT

In this way the atom encodes that when the health of a character in a story is at risk a coherent possible next action to happen is, either that someone finds by accident the wounded character or that someone realises that the wounded character had an accident.
After action four is executed, the context of the prince is not modified and the context of the priest becomes equal to the prince's context. So, no atom is added to the Abstract Representation and after action five is read the Abstract Representation looks as follows:

*** Atom
Tensions=> Hr(a)
Emotional Links=> NIL
Possible Next Action=>
b FOUND_BY_ACCIDENT a
b REALISED a HAD_AN_ACCIDENT
b CURED a

Finally action five produces a change in the characters' context, a new atom is added to memory, and the process continues until the story ends.

In order to make it easy to understand, the atom presented in the example is very simple. However, atoms can become very complicated with large numbers of Emotional Links, Tensions and characters involved. In MEXICA, special attention is given to the correlation between characters in the atoms, as was mentioned above. If a new atom which has the same Emotional Links and Tensions as an old atom in memory is created, but the correlation between characters is different in such atoms, they are not considered as equals and the new atom is allocated in memory. This distinction allows encoding actions with semantic differences as different structures. For example, if the first action in a story is

WARRIOR TRIED_TO_KISS PRINCESS, the tale changes completely if the second actions is PRINCESS PUNCHED WARRIOR or if the second action is WARRIOR PUNCHED PRINCESS. MEXICA produces different atoms when such sequences are found in different tales.

Atoms are the core element for the production of material during the Engaged State They allow MEXICA to write stories without the use of specific-goals or predefined story-structures as the main way to drive development of the story.

4.2.3.3 Tensional Representation.

For every action in a tale MEXICA calculates and stores the tension produced in the story in a variable called Tension to the Reader. The Tensional Representation is a vector that records the different values over time of this variable. In other words, each time an action is executed a variable named Tension to the Reader is updated with the value of the tension accumulated in the tale, and such a value is stored in a vector called Tensional Representation. The Tensional Representation permits representing graphically a story in terms of the tension produced in the reader. When MEXICA writes a new story it can use the Tensional Representation of the Previous Stories as frames to evaluate the story in progress (see Section 4.3.2.3-B). This evaluation is used to set some guidelines. From now onwards, all situations that produce tension in the reader are referred to as Tensions.

The value of Tension to the Reader at time=t depends on the kind and number of tensions active at that moment. The user can assign a value between zero and fifty to each of the situations that produce tension. By default the system sets a value of 10 to Love Competition. A value of 20 to Health at risk, Life at risk, Prisoner, Clashing Emotions and Potential Danger. And a value of 30 to Dead Characters.

MEXICA examines all characters' contexts to determine which tensions must be taken into account when calculating the value of the Tension to the Reader. That is, the fact that a tension is found in a character's context does not imply that it must be included in such a calculation. For example, if characters A, B and C are in the same location and character B has an accident, the tension Hr(B) is joined to the context of three of them. However, it does not mean that the Tension to the Reader will be incremented three times. The characters' contexts represent what actors are aware of. In this way, the Tension to the Reader must not be incremented just because a character is aware of a particular tension that involves other characters. In MEXICA, such a variable is incremented only when the characters involved in the tension are aware of it. In the previous example MEXICA increments the Tension to the Reader only when it finds in the context of character B the tension Hr(B).

So, the tension Health at Risk is triggered when a character (e.g. character B) has an accident or is wounded by someone. When MEXICA finds in the context of character B the tension Hr(B):* the value of the Tension to the Reader is incremented by 20. If any other character is aware of this

situation, the Tension to the Reader is not further incremented. The same process is used for Life at risk, Prisoner and Clashing Emotions.

To illustrate this situation, the characters' contexts after action four from the example in last section are examined again. In this case, the tension Lr(Prince) is found in the context of the two characters participating in the tale

*** Time => 4 Action: priest REALISED prince HAD_AN_ACCIDENT
Charac: PRINCE Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

Charac: PRIEST Pos: Texcoco Lake Status: Alive
Tensions => Hr(prince)

After MEXICA examines the prince's context, because the main character in the tension is the prince, the Tension to the Reader is assigned a value of 20. However, when MEXICA examines the priest's context, the Tension to the Reader is not further incremented because the main character in the tension is not the priest but the prince; that is, the priest is just aware that the health of the prince is at risk. So, the final value of the Tension to the Reader at time=4 remains 20.

A similar process is followed to calculate the tension due to Love Competition. If character A and character B are in love with character C, a tension is triggered due to the competition between them. If MEXICA finds in the context of character A such a Love Competition (i.e. if character A is aware of it), the Tension to the Reader is incremented by ten; the same happens when examining B's context. So, if both characters are aware of the competition the Tension to the Reader is incremented by 20; if one of them is aware of it, the tension is just incremented by 10. If any other character is aware of this situation, the Tension to the Reader is not further incremented.

Potential Danger arises when character A has an Emotional Link of any type and intensity of -3 towards character B, i.e. A(-3,*):B, and both of them are in the same location. Thus, when MEXICA finds in the context of character A the tension due to Potential Danger, the Tension to the Reader is incremented by 20. If any other character is aware of this situation, the Tension to the Reader is not incremented.

Finally, if character A kills another character, MEXICA increments the Tension to the Reader when it finds in the context of character A the tension Character Dead. If any other character is aware of this situation, the Tension to the Reader is not incremented. There is an exception to this process; if enemy is the dead character the tension is not incremented. This is under the assumption that the enemy’s death does not produce a problem to anybody.

To summarise, to calculate the Tension to the Reader MEXICA examines all characters' contexts. When a tension is found in the context of a character that participates in it, in any of the ways described

in this section, the Tension to the Reader is incremented. Once it has been calculated, MEXICA stores the value of the Tension to the Reader in the Tensional Representation of the story. This process if repeated for each action in the tale.

4.3 Creating a New Story.

All new stories in MEXICA start with the user specifying the initial action, the characters participating in it and a starting location that becomes the default location for the story. When developing a new tale MEXICA can work in four Operation Modes (see Section 3.2.5).

1)Engaged State 1 (E1).
2)Engaged State 2 (E2).
3)Engaged and Reflective States 1 (ER1).
4)Engaged and Reflective States 2 (ER2).

When running under E1 or E2 operation modes, MEXICA works exclusively within the Engaged State. By contrast, when running under ER1 or ER2, the Engaged and Reflective States work together in order to produce a new story. When running under E2 or ER2 operation mode the guidelines and filters are active. When MEXICA is running under E2 operation mode run-time requirements are set by default values, while when running under the ER2 such requirements are set by the guidelines produced during the Reflective State. In this way, the guidelines set during the Reflective State influence the production of material during the Engaged State.

Engaged State 1 (E1)	No Filtering Process	No guidelines.
Engaged State 2 (E2)	Filtering Process	Guidelines set by default.
Engaged and Reflective States 1 (ER1)	No Filtering Process	No guidelines.
Engaged and Reflective States 2 (ER2)	Filtering Process	Guidelines set during the Reflective State

The interaction between the Engaged and Reflective States is controlled by three factors. The first one is a parameter modifiable by the user that establishes the initial state (either Engaged or Reflective) under which the system starts to run. (From now onwards this parameter is referred as Initial State.) Depending on the content of characters' contexts and the knowledge in memory, the Initial State can have an important influence in the developing process (see examples in Chapter V).
The second factor is a constant, also modifiable by the user, which indicates the number of actions that MEXICA must generate under the Engaged State before switching to the Reflective State to evaluate the material produced. From now onwards this parameter is referred as CtEg-Rf (Constant Engaged-Reflective). When the constant is set to one MEXICA works with a strong tendency towards the Reflective State, i.e. it analyses and evaluates every action produced in the tale. On the other hand, if

the constant is set with a high value (e.g. 10) MEXICA works with a strong tendency towards the Engaged State, i.e. it produces a lot of material before switching to the Reflective State. Since MEXICA is a system developed to study the interaction between the Engaged and Reflective States, the possibility to manipulate the cycle in this way is an important characteristic of the system.
The third factor which influences the interaction between the two states arises when no more material can be produced, i.e. when an impasse is declared. In this case MEXICA switches to the Reflective State to try to break the impasse. Once it is broken MEXICA switches back to the Engaged State.

During the Engaged State, possible next actions are retrieved using characters' contexts to construct the structures to match atoms in memory. In MEXICA all characters' context participate in this process; i.e., MEXICA retrieves a set of possible next actions using the context of the first characters in the tale, retrieves another set of possible next actions using the context of the second character in the tale and so on. Then, after all options are in working memory one is selected as the next event in the story.

During the Reflective State impasses are broken, the material produced is evaluated, guidelines prepared and preconditions verified and fulfilled. When the story is finished, the final analysis takes place.

4.3.1 Engaged State.

The retrieval of information from memory and selection of the next action constitutes the basic cycle during the Engaged State. MEXICA uses dynamic structures —known as Associative Structures— as samples (or probes) to match atoms in memory and retrieve the set of possible next actions associated to them. Associative Structures are built from characters' contexts. These structure are referred as dynamics because, if no atom can be matched, they modify their own organisation (i.e. they transform themselves) to try to match another atom.
The logic behind dynamic structures works as follows. When no match is possible between an Associative Structure and an atom in memory, MEXICA attempts to find other alternatives by modifying the structure. That is, by modifying the Associative Structure MEXICA is changing the requirements to match an atom. So, new options no available in the previous search are accessible now. As long as the Associative Structure conserves after each transformation the core characteristics of the characters' contexts, MEXICA is able to retrieve coherent possible next actions following this procedure.
The maximum number of transformations allowed is three; thus, Associative Structures can be of three types. The first is identical to the character's context from where it was built and is referred as Associative Structure I (ASI). The second is the result of the first transformation and is referred as Associative Structure II (ASII). Finally, the third one is the result of the last transformation and is referred as Associative Structure III (ASIII).

In this way, instead of recalling equal episodes from previous stories and adapting them to the actual story, MEXICA recognises similar context-situations; this characteristic gives it a greater flexibility. Associative Structures are not indexes to access a particular area in memory but structures that can be matched against different patterns in memory. Associative Structures do not relax constraints but modify its organisation.

There is another important process related to retrieving possible next actions. When scanning long-term memory MEXICA tries to match the Associative Structure against an atom. If it fails, before transforming the Associative Structure an attempt is made to find another atom which contains the Associative Structure; that is, MEXICA tries to find an atom in which is possible to find as part of its organisation the Associative Structure. From now onwards this process is referred as ACAS-Process (Atom Containing the Associative Structure).

To explain it with a simple example, the reader can picture a person who is looking in a library (the memory) for an article that describes the sacred rituals of the Mexicas (the Associative Structure). If this person does not find it but instead come across a small book (an atom) which describes not just the Mexicas' rituals but also the Mayas and Olmecas' ones, i.e. a book which contains as part of its structure the article, probably this book will satisfy all the requirements the person was expecting the article to satisfy. However, it is necessary to establish some limits to this process. If instead of finding such a book this person gets a twenty volume encyclopaedia that describes all the pre-Hispanic civilisations (Mexicas, Mayas, Zapotecas, Chichimecas, Olmecas, etc.) where the Mexicas occupies volumes I and II, probably it will be too much.

To avoid this situation MEXICA requires the matched atom to be equal to the Associative Structure in a minimum percentage. From now onwards this percentage is referred as ACAS-Constant. So, in order an Associative Structure can match an atom, the atom must be equal to the Associative Structure in at least ACAS-Constant percent. MEXICA has three different ACAS-Constants, one for each type of Associative Structure; i.e., ACAS-Constant I is used with the ASI, the ACAS-Constant II with the ASII and the ACAS-Constant III with the ASIII. They are definable parameters; in this way the user can decide which percentage is associated to each of the Associative Structures.

It is expected that when the ACAS-Constant is set to a low value unusual actions are brought to working memory increasing the originality of the story. However, unusual actions also increase the risk of producing sequences of events with a lack of coherence, i.e. events with a weak link to the actual context in the story. On the other hand, a high value of the ACAS-Constant leads MEXICA to retrieve actions well connected with the story in progress, but which are very predictable. So, it is necessary to find a value that balances properly this situation.

The whole process of retrieving possible next actions can be summarised as follows:

1)Create from characters' contexts an Associative Structure I.

2)Scan memory (Abstract Representation) to match the Associative Structure against an atom.

3)If no match is possible, try to find an atom that contains as part of its organisation the Associative Structure (ACAS-Process).

4)If no match is possible the Associative Structure is transformed (to ASII or ASIII) and the cycle starts in step two until an atom is matched or two transformations are made.
This process is repeated for each of the characters in the story; that is, all actors in a tale contribute (or at least try to contribute) with a set of possible next actions.

The retrieval of possible next actions through dynamic structures and the ACAS-Process is one of the most important characteristics in MEXICA. The following sections describes in detail how the Associative Structures are created and transformed, how the search in long-term memory is performed, how actions are instantiated and selected.

4.3.1.1 Bringing Possible Next Actions to Working Memory.

The process to create the Associative Structure works as follows. When MEXICA is analysing the context of a particular character, e.g. character A:
1) All Emotional Links and Tensions in the context of character A are generalised and joined together to form what is known as Associative Structure I (ASI). A search in long-term memory is launched to try to match the ASI with an atom.
2) If it is not possible to match an atom the first transformation occurs. It consists of eliminating from the ASI all Emotional Links where character A does not participate, and all Tensions —except Life at Risk and Health at risk— where character A does not participate. In other words, this new structure contains all Emotional Links where character A participates, all Tensions where character A participates, and all Life at Risk and Health at Risk Tensions (independently of who participates in them) found in ASI. The new structure created is named Associative Structure II (ASII).
3) If it is not possible to match an atom the second transformation occurs. It consists of eliminating from the Associative Structure II all Emotional Links; i.e., just those Tensions where character A participates and all Life at Risk and Health at Risk Tensions (independently of who participates in them) are kept. The new structure created is named Associative Structure III (ASIII).

The same process is repeated for all characters in the tale. For example, having the following context:

CONTEXT ***
Charac: WARRIOR Pos: Texcoco Lake Status: Alive
Emotional Links=>
Warrior(-3,1):enemy
Eagle_Knight(-3,1):enemy
Tensions => Lr(Warrior):enemy

The first Associative Structure is created by copying all the Emotional Links and tensions which form the character's context.

Associative Structure I
Emotional Links=>

Warrior(-3,1):enemy
Eagle_Knight(-3,1):enemy
Tensions => Lr(Warrior):enemy

If no possible next actions are retrieved from memory, the first transformation occurs:

Associative Structure II
Emotional Links=> Warrior(-3,1):enemy
Tensions => Lr(Warrior):enemy

If no possible actions are retrieved from memory, the second transformation is made:

Associative Structure III
Tensions => Lr(Warrior):enemy

There is a requirement to fulfil when the search is done using an ASIII. All retrieved possible next actions must be fully instantiated; those that do not fulfil this condition are eliminated (see Section 4.3.1.2 for examples of uninstantiated possible next actions brought to working memory). This restriction attempts to minimise the possibility of bringing actions that are unrelated to the story in progress to working memory.

Once the Associative Structure has been created, MEXICA starts the search in long-term memory. It works as follows. Atoms in memory are organised according to their attributes (i.e. number and type of Emotional Links and Tensions); so, MEXICA tries to find one that shares with the Associative Structure the same properties. If it succeeds, MEXICA attempts to correlate the characters in the atom and the characters in the Associative Structure; if the atom and the Associative Structure share the same attributes but the characters cannot be correlated, MEXICA looks for another atom which can fulfil both requirements. This correlation between characters is important to link properly the actors in the tale in progress and the characters in the set of possible next actions in the atom (see also the explanation of the importance in the relation between characters in the atom and the possible next actions in Section 4.2.2.1).
If the process succeeds, the actors in the list of possible next actions pointed by the atom are substituted with the characters in the Associative Structure and then transferred to working memory, where finally they are instantiated. For example, the reader can imagine that a story in progress has produced the following context:

CONTEXT ***
Charac: PRINCESS Pos: Texcoco Lake Status: Alive
Emotional Links=> princess(+3,2):Jaguar_Knight
Tensions => NIL

MEXICA generalises the context to create the Associative Structure

Associative Structure:
Emotional Link=> A(+3,2):B
Tension=> NIL

and a map which correlates the characters in the context and the Associative Structure is formed (this map had not been mentioned before to avoid complicating previous explanations).

princess <-> A
Jaguar_Knight <-> B

Now, MEXICA matches the Associative Structure with an atom in memory

Atom
Emotional Link=> C(+3,2):D
Tension=> NIL
Possible Next Action=> D married C

and based in the correlation between characters described earlier it creates a map between the actors in the Associative Structure and the atom.

A <-> C
B <-> D

With the help of this map, the characters in the possible next actions are substituted with the characters in the Associative Structure.

Possible Next Actions: (D married C) -> Possible Next Actions: (B married A)

Finally, the possible next action is transferred to working memory where, with the help of the map between the characters in the context and the Associative Structure, the possible next action is instantiated.

Possible Next Action: (Jaguar_Knight married princess)

If no atom is found which can match the Associative Structure, MEXICA changes tactic; this time it looks for an atom where the Associative Structure can be found as part of the atom's organisation. That is, it tries to match an atom using the ACAS-Process. For example, having the following Associative Structure

Associative Structure I:
Emotional Link=> A(+3,2):B
Tension=> NIL

an atom that contains as part of its organisation the Associative Structure is

Atom
 Emotional Link=> C(+3,2):D
 Tension=> Hr(D)
 Possible Next Action=> C cured D

If such an atom is found, MEXICA extracts from it all those elements equal to the ones in the Associative Structure and attempts to correlate the characters between them. Again, if it succeeds the characters in the list of possible next actions pointed by the atom are substituted by the characters in the Associative Structure and transferred to working memory; once there, the possible next actions are instantiated with the characters in the tale.

When MEXICA is trying to match an atom using an Associative Structure I, by default it requires that such an ASI constitutes at least 50% of the atom. In other words, at least 50% of the atom must be equal to the Associative Structure I. This percentage is known as the ACAS-Constant I. When MEXICA is trying to match an atom using an ASII, such percentage is set by default to 70%; this percentage is known as ACAS-Constant II. And when MEXICA is trying to match an atom using an ASIII the percentage is set to 90%; this percentage is known as ACAS-Constant III. If this minimal percentage is not reached or the characters cannot be correlated, MEXICA looks for another atom that can fulfil the requirements. (All these percentages are modifiable by the user.)

When a set of possible next actions has been brought to working memory, through a process described in following sections MEXICA eliminates those actions which are not useful for the story. If all possible next actions are left out, MEXICA starts a new search in long-term memory to try to find another atom that can offer more possible next actions.

The ability to transform the Associative Structure (dynamic structures) and the ACAS-Process are two of the most important characteristics in MEXICA; they provide the flexibility required to create unexpected sequences of actions. Notice that in the case of the dynamic structures MEXICA is not relaxing constrains but transforming the structure. This transformation can be done in different manners without affecting the way the procedure to retrieve possible next actions from memory works. In MEXICA's prototype the three heuristics just explained are implemented; however, other ways to build such structures are also suggested (see Future Work in Chapter VI).

This section can be summarised in the following sentences: an Associative Structure is formed and a search in memory executed. If the search fails MEXICA tries to match an atom using the ACAS-Process. If it also fails a second Associative Structure is formed and a new search launched. If the search is not successful MEXICA tries to match an atom through the ACAS-Process. If it fails again a third Associative Structure is formed and another attempt is made. If it does not work a final attempt is tried using the ACAS-Process. A search fails when an atom which shares with the Associative Structure the same characteristics cannot be found in memory. A search also fails when the characters in an atom cannot be correlated with the characters in the Associative Structure. When some material is obtained, MEXICA transfers it to working memory.

A last comment. MEXICA performs the process just described for each of the characters in the tale. That is, each character's context is used to match an atom. In order to bring different possible next actions to working memory MEXICA never matches the same atom twice. That is, if two characters' contexts are equal they are forced to match different atoms.

4.3.1.2 Selecting the next action.

When MEXICA is running under E1 operation mode, the next action in the story is chosen randomly between the options in the set of possible next actions brought from long-term memory. Only a simple process to avoid repeating the same action twice in a row in a tale is carried out.
During the E2 operation mode, before selecting the next action all options brought from long-term memory are filtered in order to eliminate those that not contribute to do story development. Then, the next action is selected randomly between the remaining alternatives.
Before an action can be filtered and/or selected it must be completely instantiated.

A. Instantiating Characters.

Although MEXICA creates the maps described in the previous section to establish correlation between characters, sometimes possible next actions can arrive into working memory without being instantiated. For example, the action VIRGIN HAD_AN_ACCIDENT produces the following context.

CONTEXT ***
Charac: VIRGIN Pos: Texcoco Lake Status: Alive
Emotional Link => NIL
Tensions=> Hr(Virgin)

The Associative Structure that arises from it is

Associative Structure
 Emotional Links=> NIL.
 Tensions=> Hr(A)

and a map that correlates Virgin and A is created. Now, let us imagine that the following is the atom matched in memory.

Atom.
 Emotional Links=> NIL
 Tensions=> Hr(C)
 Possible Next Actions=> D cured C

Thus, MEXICA creates another map which correlates A and C. However, when the characters in the possible next action are substituted, MEXICA does not find and a equivalent actor for character D.

Possible Next Actions: (D cured C) -> Possible Next Actions (? cured A).

And the same happens when MEXICA tries to instantiate in working memory the possible next action using the first map.

Possible Next Actions: (? cured A) -> Possible Next Actions: (? cured Virgin).

That is, an uninstantiated action arrives in working memory.

Under the E1 or ER1 operation mode, to solve this problem MEXICA creates a set of potential candidates to instantiate the action and selects one of them at random. The way this set is created is now described. All characters in MEXICA except those participating in the context from where the Associative Structure used to retrieve the actions was formed, are used to build the set. For instance, in the last example all characters in MEXICA would be included in the set of candidates except the Virgin.

The reason why those characters are excluded (or banned) for participating in the set is now explained. All characters in the context are included in the first map; so, all characters in the context are represented when the second map is created. An uninstantiated action arrives when there is no correlation between the actors in the second map and the actors in the possible next actions, which implies a lack of correlation between the actors in the context and the actors in the possible next actions. So, if the actors in the context have already failed to be instantiated, they are considered as inadequate and banned to be included in the set of candidates to instantiate the possible next actions.

However, in order to make the system more flexible MEXICA allows some variations to this rule. A parameter modifiable by the user known as Forbidden Characters prevents some of the characters in the context from being banned. It works as follows. When Forbidden Characters is set to Active MEXICA bans all characters in the context (as just explained).

When Forbidden Characters is set to Half-Active MEXICA just bans the character who holds the context from where the Associative Structure which matched the atom was created. In the last example, the only character banned would be Virgin.

When Forbidden Characters is set to No-Active MEXICA does not ban characters, i.e. all characters are allowed being included in the set of potential candidates to instantiate the action.

Finally, in order to instantiate the action there is one last requirement. If the action includes the postcondition A(+3,2):B (i.e. that A fells in love with B) the system verifies that characters A and B are not the same, and that both of they are not males or females. If this requirement is not satisfied the system looks for other characters (following the same procedure) to instantiate the action.

Thus, under the E1 or ER1 operation modes MEXICA selects randomly a character from the set of candidates. When MEXICA is running under the E2 or ER2 operation mode the problem is approached in a different way. The system tries to avoid the random procedure by using other three alternatives.

First, as was commented before, each character in the story contributes with a set of possible next actions. In another words, each characters' context is used to retrieve from long-term memory possible next actions. So, MEXICA tries to replace those uninstantiated actors with the character who holds the context. For example, if the context of character A retrieves uninstantiated actions, MEXICA tries to instantiate them using character A.
However, there are two situations that can prevent this procedure from success: when character A has been banned (as explained in the previous paragraph) or when there is more than one character not instantiated.

In this case MEXICA tries a second option: it tries to use the rest of the characters in the tale to instantiate the actions. With this heuristic MEXICA is trying to apply that rule taught in courses of theatre and improvisation, that Johnstone (1989) refers to as reincorporating. To avoid lack of structure and coherence when improvising a story, characters used previously in the tale are reincorporated.
In order to give all characters in the tale an opportunity to be used to instantiate the action, each time MEXICA reincorporates a character it verifies if the resulting action makes the story flow (see part B in this section for a definition of an action which flows). If it does not flow MEXICA tries again to instantiate the action using a different character.
This procedure fails if the rest of the characters in the tale are banned, if all instantiated actions do not flow, or if some characters remain uninstantiated in the action.

Before explaining the third option, the concept of group class is introduced. In MEXICA, as a support for the procedures to instantiate characters, all characters are divided in what has been called group class. Group classes are organised as follows:

[tlatoani, prince]
[princess]
[Eagle_Knight, Jaguar_Knight, priest]
[lady, Virgin,]
[Farmer, Hunter, Fisherman, Trader, Warrior, Artist]
[Slave]
[enemy]

The main idea behind group class is to be able to substitute one character for an equivalent one. That is, in MEXICA two characters are considered as equivalent when they belong to the same group class. This classification is arbitrary and just useful during the instantiation process.

Now, the third alternative is explained. It requires that at least one of the characters in the action has already been instantiated and it works as follows. MEXICA studies (in the Concrete Representation) the way the uninstantiated action has been used in Previous Stories; it tries to find an instance where an actor with the same group class as the known character participates. If it is found, the other character in such an instance is used to instantiate the action.

For example, jaguar knight and priest belong to the same group class. So, if the uninstantiated action X CURED PRIEST arrives in working memory, and in the Concrete Representation the action LADY CURED JAGUAR_KNIGHT is found, MEXICA detects that jaguar knight and priest belongs to the same group class and uses lady to instantiate the action resulting in LADY CURED PRIEST.
This alternative fails when none of the characters in the action are instantiated, or if none equivalent actors are found in the Concrete Representation.

When all previous alternatives fail, MEXICA uses the random procedure described earlier to instantiate the action. A final comment. If after an action is instantiated it results that the characters participating in it are located in different positions, automatically MEXICA situates both of them in character A's position. Later, during reflection, the system justifies how character B moves to character A's position.

B. Filters.

MEXICA is provided with a group of routines called Filters that eliminate all those actions not useful to the story. Filters are divided in two groups according to the way they establish what actions are not useful. In the first group, the characteristics of not useful actions are encoded in the program; i.e. the programmer defines them. In the second group, those characteristics are established by MEXICA at run time; i.e. depending on the features of the story in progress MEXICA decides which actions are not useful during the developing of the story.

The first group is constituted by routines that:

1. Delete all those possible actions that are equal to the last action performed in the story.
2. Verify that just those actors who are alive in the story participate in a possible next action.
3. Check that an action in a story flows. By definition an action flows if, after performing it, the Emotional Links or Tensions in any of the characters' context changes or if the Tension to the Reader changes. In this way, boring sequences of actions like JAGUAR_KNIGHT WENT_POPOCATEPTL_VOLCANO, JAGUAR_KNIGHT WENT_TLATELOLCO_MARKET, JAGUAR_KNIGHT WENT_TEXCOCO_LAKE, END, are avoided.
4. Eliminate no representative actions. The concept of no representative action is now explained. When MEXICA is developing a story a new character can be introduced in the tale at any moment. The context of this new character will include just the postconditions of the action used to introduce it in the story. And this can be a problem if the remainder characters' contexts include long list of Emotional Links or Tensions. That is, when MEXICA retrieves possible next actions from long-term memory it uses all available characters' contexts to create the Associative Structures. The context of the new character lacks important information about previous actions in the story; so, the possible next actions retrieved from it might include events not connected at all with the story in progress. To avoid this situation MEXICA detects and eliminates them through the following process. MEXICA analyses all characters' contexts to determine which of them has the greater number of elements (the number of elements is obtained adding the number of

Emotional Links and the number of Tensions in a context). This number is used as a reference; so, all characters' context which number of elements is less than the reference divided by two are classified as no representative contexts. And all actions brought to working memory from a no representative context are classified as no representative actions, therefore eliminated.

5. Eliminate illogical actions. If character A has strong positive Emotional Links towards character B (i.e. A(+2,*):B or A(+3,*):B), and the action performed triggers strong negative Emotional Links in B towards A (i.e. B(-2,*):A or B(-3,*):A), that actions is considered as illogical and eliminated by the filters. In other words, if the PRINCESS is in love with JAGUAR_KNIGHT (strong positive Emotional Links) and in the next action PRINCESS performs an action that produces that JAGUAR_KNIGHT hates her (e.g. PRINCESS ATTACKS JAGUAR_KNIGHT) that actions is considered as illogical. By contrast with the previous routines in this group that always are active, the user can activate or deactivate this filter through a parameter known as Logical Action. This option is included because sometimes, with the help of the Reflective State, sequences of illogical actions can be transformed into good stories (see the story in Section 5.3).

The second group of filters is constituted by routines which verify that the story in progress is observing certain constrains; these constrains are denoted Guidelines for the Story. They are set either by some procedures during the Reflective State when MEXICA is running under the Engaged and Reflective States 2 operation mode, or by default values when MEXICA is running under the Engaged State 2 operation mode. Details of the process followed to assign their values to the guidelines is explained in Section 4.3.2.3

The guidelines try to produce an increment or decrement in the Tension to the Reader by restricting the requirements to select the next action in the tale. Also, they establish a minimum percentage of novelty that any action must satisfy. In MEXICA the novelty is defined by the number of times an action has been used in Previous Stories. If an action does not fulfil the guidelines' requirements, it is deleted.

So, the basic cycle during the Engaged State consists in creating Associative Structures from character's contexts and retrieving from long-term memory sets of possible next actions. A filtering and selecting process is applied to them. If none of the options is suitable as a possible next action MEXICA tries again to get a new sets of possible next actions. When is not possible to retrieve any new option, an impasse is declared.

4.3.2 Reflective State.

During the Reflective State MEXICA performs four main tasks: it checks (and solves) preconditions, breaks an impasse produced during the Engaged State, evaluates the story in progress and produces guidelines to the Engaged State. Once the story is completed, MEXICA performs the final analysis. Each of these tasks is now described.

4.3.2.1 Checking Preconditions.

MEXICA verifies that the preconditions of all the actions in the story in progress are satisfied. This procedure is necessary because during the Engaged State preconditions are not checked at all. The process consists of four steps:
1)Clear all characters' context in the story (i.e. delete all their Emotional Links and Tensions) and get the first action in the story ready for inspection.
2)Verify that all its preconditions are satisfied. If they are not satisfied, an action to fulfil them is inserted in the story and the cycle starts again in step number one.
3)When all the preconditions in the action are satisfied the characters' contexts is updated with the postconditions triggered by the action.
4)Take the following action in the story and repeat the cycle from step number two.

Step number two is divided in two parts:
a)Verify that the characters participating in the action are in the same location.
b)Verify that the characters' contexts satisfy the action's preconditions.

In other words, preconditions can be unsatisfied because the characters participating in the action are not in the same location or because characters' contexts do not include the Emotional Links or Tensions necessary to satisfy the action's preconditions. The first situation is known as Location Problem and the second as Context Problem.

The Location Problem is solved as follows. Most actions have two characters; MEXICA has established as a rule that all characters participating in an action must have the same location as the second one (the same applies for actions with three characters). If MEXICA detects an action with Location Problems, an event is inserted to move the first character to the location of the second. For example, if character A is located in the Palace and character B in the temple, the action A FOLLOWED B will present a location problem. In this case MEXICA inserts the action B WENT_PALACE before the action FOLLOWED in order to sort out the problem.

The Context Problem is solved as follows. When the actual context in the story does not satisfies the preconditions of the action, MEXICA checks two possible situations: either if an action which fulfils the preconditions has already been performed in the story but the right characters are not aware of it (remember that not all characters are aware of the same things), or if an action which satisfies the preconditions has not occurred at all in the story.
In the first case MEXICA solves the problem by inserting the compound action REALISE together with the action that satisfies the preconditions. For example, in the sequence PRINCESS HAD_AN_ACCIDENT, PRIEST CURED PRINCESS, MEXICA inserts the compound action

PRIEST REALISED PRINCESS HAD_AN_ACCIDENT to solve the problem (the priest must be aware that the princess is injured in order to cure her).

In the second case MEXICA looks for all actions whose consequences fulfils the preconditions and selects one at random; if no action can be found the story is abandoned. If one or more characters in the unsatisfied action do not exist in the story, MEXICA first introduces them in the tale through the action ACTOR; then, it inserts the action to satisfy the preconditions. For example, to satisfy the preconditions of WARRIOR ATTACKED HUNTER, MEXICA inserts the action HUNTER MUGGED WARRIOR; if the character Warrior does not exist, MEXICA first insert the action WARRIOR ACTOR and then HUNTER MUGGED WARRIOR.
MEXICA avoids inserting an action to justify itself. That is, if the postconditions of ACTION X satisfy its own preconditions, MEXICA avoids inserting ACTION X to justify itself. The reason for this restriction is that such situation can lead to an endless loop. For example, if the precondition of A PUNCHED B is A(-2,*):B and its postcondition B(-2,1):A, PUNCHED can satisfy itself endlessly. In other words, the precondition of PRINCESS PUNCHED PRIEST is princess(-2,*):priest which could be satisfied with the action PRIEST PUNCHED PRINCESS. Now, PRIEST PUNCHED PRINCESS can also be satisfied with PRINCESS PUNCHED PRIEST, and so on.

When MEXICA inserts an action to satisfy preconditions, it always checks that such an action flows and satisfies the Novelty Guideline; however, the Tensional Guidelines are not verified (see Section 4.3.2.3 for an explanation of the Novelty and Tensional Guidelines). The reason is now explained. When checking preconditions MEXICA examines the whole story in progress and inserts actions wherever they are necessary: at the beginning, in the middle or at the end of the story. The Tensional Guidelines indicates a desired tendency in the behaviour of the Tension to the Reader. But it is a desired behaviour for the future actions in the tale, not for the previous ones. So, it is not point to have active the Tensional Guidelines.

4.3.2.2 Breaking an Impasse.

MEXICA tries to break an impasse by "copying" the way actions have been used in Previous Stories. The process consists of the following steps:
1)Obtain from the Concrete Representation all those actions that have followed in Previous Stories the deed which triggered the impasse.
2)Eliminate all options that are inadequate to break the impasse and select one action between the remaining ones.
3)Join the action to the end of the tale in progress and switch back to the Engaged State.

When the actions which have followed the one which triggered the impasse are obtained, MEXICA creates a map where the relation between characters in such actions and the one which produced the impasse is kept; this map helps later to instantiate properly the event chosen to break the impasse.
For example, if the action TLATOANI KISSED LADY triggers an impasse, MEXICA looks for all actions which have followed SOMEONE KISSED SOMEONE-ELSE. Let us imagine that in the Previous Stories exists the sequence EAGLE_KNIGHT KISSED PRINCESS, PRINCESS PUNCHED EAGLE_KNIGHT; so, MEXICA chooses the action SOMEONE PUNCHED SOMEONE-ELSE. However, in order to preserve the coherence of the original sequence is necessary to maintain the relation between the characters; thus, MEXICA creates a map where it is registered that the character who performed the action of kissing is the one who is punched, and the character who received the action of being kissed is the one who punches. In this way MEXICA obtains LADY PUNCHED TLATOANI.

Sometimes, however, it is not possible to fully instantiate all characters. For instance, if in the last example the sequence in the Previous Stories is changed to EAGLE_KNIGHT KISSED PRINCESS, JAGUAR_KNIGHT GOT_JEALOUS_OF EAGLE_KNIGHT, MEXICA will obtain the uninstantiated action SOMEONE GOT_JEALOUS_OF TLATOANI. This situation arises because it is not possible to establish a map to link all characters as in the last example.
MEXICA solves this problem by instantiating the actors as follows. First, it tries to substitute the unknown character with one of the other actors in the tale (i.e. it tries to reincorporate characters); thus, it instantiates the action with the first actor in the tale. To check that it makes the story moves, MEXICA verifies that the instantiated action flows. If it does not satisfies this condition, MEXICA takes the next character in the tale and repeats the same steps until it finds an action that flows or there are no more characters to try with.
If it is not possible to instantiate the action following this procedure, MEXICA tries a second option. It checks if in the Previous Stories exists an event where such an uninstantiated action has as one of its actors the known character. That is, following the last example MEXICA verifies if in a previous story there is an action where someone gets jealous of tlatoani. If it is found, this other character is used to instantiate the action.
If it fails, the same process is repeated again but this time MEXICA looks for an action in which is possible to find an actor with the same Group Class as the known character. Following the previous example, MEXICA looks for an action where someone gets jealous of an actor with the same Group Class as tlatoani. If it is found, this other character is used to instantiate the action.
If this process also fails, MEXICA instantiates the action with any actor who has participated in such an uninstantiated action in Previous Stories.

In order to get rid of all the actions that are inadequate to break the impasse, an elimination process is run. In MEXICA an action is considered inadequate when it includes two uninstantiated characters, when it includes a dead character, when it does not flow or when it does not satisfy the Novelty

Guideline. The Tensional Guidelines are not included in this process because the Tension to the Reader is considered as no important when MEXICA is trying to break an impasse.
Actions with two uninstantiated characters are eliminated because they have a big chance to introduce incoherent actions in the tale. That is, the procedures described earlier in this section to instantiate characters cannot be used when more than one character is uninstantiated. Thus, the only option left is to select characters randomly, which probably will produce an incoherent event. So MEXICA prefers to eliminate this type of actions.
Notice that if an action which does not flow is used to break an impasse, a new impasse will be declared since the characters' contexts used to retrieve possible next actions from long-term memory remain without any change.

Once all non-useful actions have been eliminated a selective process starts. This process divides all the remaining options into two categories: those that were fully instantiated and those that needed to pass through the instantiation procedures. The former provides a set of actions that can be integrated easily in the tale in progress; i.e. because they were fully instantiated from the beginning, actions in this category just include characters that are already participating in the tale. This avoids introducing actors "out of the blue" who might produce inconsistencies. The latter category includes actions that might present such a problem.
In this way, the selective process chooses randomly an action from the set of options belonging to the first category. If such a set is empty, a random action is selected from the second group. If both sets are empty (i.e. if the elimination process eliminated all actions) MEXICA tries a last option: equivalent actions.
The heuristic Equivalent Action looks in the Concrete Representation for an action equivalent to the one that produced the impasse and then follows the same steps described for the previous heuristic. That is, it gets all those actions that in previous stories have followed the equivalent action, deletes all those which are not useful and selects one of the remaining ones. In MEXICA two actions are considered equivalent if they share a percentage of equal postconditions; this percentage (known as Equivalent Constant) is modifiable by the user although by default it is set to 50%. If all the alternatives brought through the equivalent action are not useful to break the impasse, MEXICA looks for another equivalent action until either finds an action to break the impasse or no more equivalent actions can be found.
If both heuristics fail an unbreakable impasse is declared and the story is abandoned. Otherwise, MEXICA joins the selected action to the end of the tale in progress and verifies preconditions. If the preconditions are not satisfied the story is abandoned.

4.3.2.3 Evaluating the story.

MEXICA performs two kinds of evaluations. One verifies that the material produced is not too similar to any of the Previous Stories. The other compares the Tensional Representation of the tale in progress

against the set of Tensional Representations produced by Previous Stories; then, it uses the most similar as a frame to evaluate the material produced. The results of these evaluations are used to set the guidelines.

Guidelines.
Novelty:
Permanent Tension:
Temporal Tension:

The guidelines are divided into two groups: the Novelty Guideline and the Tensional Guidelines. The latter group is formed by two elements: the Permanent Tension and the Temporal Tension. All them are explained in detail in the following paragraphs.

When MEXICA starts it assigns default values to all the guidelines. If it is running under the E2 operation mode, the guidelines are never modified. If it is running under the ER2 operation mode, each time MEXICA evaluates the story in progress the guidelines are updated according to the result of the evaluation.
This section explains how the value of the guidelines is calculated and assigned. The function of the guidelines is to establish a criteria to eliminate no useful actions during the filtering process in the Engaged State. So, references to that process are frequently made.

A. Novelty.

During the evaluation of novelty MEXICA verifies if the material produced during the Engaged State resembles too much any of the tales in the Previous Stories. The system has a parameter that determines the percentage of similarity authorised to exist between two tales; if that percentage is exceeded, a request is sent through the guidelines to the Engaged State for more original sequence of events. This percentage is called Novelty-Percentage, has a default value of 50% and is modifiable by the user. So, if more than 50% of the story in progress is equal to some of the Previous Stories, a request for more original actions is sent to the Engaged State through the Novelty Guideline.

To evaluate novelty MEXICA divides the story in progress into sequences; one sequence is defined as a group of two consecutive actions. For example, the following story in progress

STO
START
PRINCESS WENT_TEXCOCO_LAKE
PRINCESS HAD_AN_ACCIDENT
JAGUAR_KNIGHT REALISED PRINCESS HAD_AN_ACCIDENT
JAGUAR_KNIGHT CURED PRINCESS
PRINCESS FELL_IN_LOVE_WITH JAGUAR_KNIGHT

can be divided into five sequences.

Sequence 1	PRINCESS WENT_TEXCOCO_LAKE PRINCESS HAD_AN_ACCIDENT
Sequence 2	PRINCESS HAD_AN_ACCIDENT JAGUAR_KNIGHT REALISED PRINCESS HAD_AN_ACCIDENT
Sequence 3	JAGUAR_KNIGHT REALISED PRINCESS HAD_AN_ACCIDENT JAGUAR_KNIGHT CURED PRINCESS
Sequence 4	JAGUAR_KNIGHT CURED PRINCESS PRINCESS FELL_IN_LOVE_WITH JAGUAR_KNIGHT
Sequence 5	PRINCESS FELL_IN_LOVE_WITH JAGUAR_KNIGHT NIL

MEXICA analyses how many of these sequences can be found in Previous Stories. If one of such Previous Stories includes more than 50% of the sequences (the default value of the Novelty-Percentage), the story in progress is classified as lacking originality. If all the sequences can be found in a Previous Story, the story in progress is classified as a copy. This classification is used to assign a value to the Novelty Guideline.

MEXICA includes four parameters called Novelty Constants. Their function is to establish a criteria to evaluate the novelty of an action. Two of them, known as the Medium and Strict Novelty Constants, are modifiable by the user. By default they are set with the values of 75% and 15% respectively. The other two parameters, known as the Low and Medium Novelty Constants, have a fixed value of 0% and 50% respectively.

They work as follows. After MEXICA has created all the structures in long-term memory it verifies how many Primitive Actions have been used at least once, and how many times each of them has been employed in the Previous Stories. The relation between these two quantities provides the average number of times an action has been employed.

MEXICA links that average number with the High Novelty Constant. For example, let us imagine that such an average number is five. So, MEXICA considers that any action used in Previous Stories five or less times satisfies the requirement of high novelty.

Now, knowing that five is equal to 50% (the value of the High Constant Novelty) MEXICA calculates how many times an action can be used to satisfy the Strict Novelty Constant (15%). The result is obtained through the following equation:

x=(Strict Novelty Constant*average number) / Medium Novelty
x=(15% * 5) / 50%
x= 1.5 = 2
(Obviously the result is rounded)

So, any action used 2 or less times in the Previous Stories satisfies the requirements of strict novelty. The same process is followed for the Medium Novelty Constant. The Low Novelty Constant is set to zero, i.e. any action satisfies its requirement.

The Novelty Guideline can be set with four different values: Strict, High, Medium or Low. When it is set to Strict the filters eliminate all those actions which do not satisfy the criteria established by the strict novelty requirement. The same applies when the Novelty Guideline is set to High or Medium. When it is set to Low, no action is deleted. The default value assigned to the Novelty Guideline is Low.

If after the evaluation MEXICA classifies the story as a copy, the Novelty Guideline is set to Strict for one action. After that it is re-set to High. That is, just the first following next action in the story has to satisfy the Strict requirements; the rest have to satisfy the High requirements.
Something analogous occurs when the story is classified as lacking originality. The Novelty Guideline is set to High and after one action, it is reset to Medium.

This is the way MEXICA verifies novelty. It is a simple, effective and flexible process (e.g. the user can decide how strict the requirements are through the Novelty Constants).
Each time a story is developed MEXICA produces a report. In that reported it is indicated the number of times each sequence has been used previously, and the result of the evaluation. In this way, the user has access to a detailed information of the novelty of the story in progress.

A final point regarding novelty. MEXICA gives the user the option to decide if the system starts the development of the tale under the Engaged or Reflective State. The analysis of the evaluation of novelty allows foreseeing some of the possible consequences of choosing an initial state.
If MEXICA starts the development of the tale under the Engaged State, the Novelty Guideline is set with the default value (Low). So, the filters delete no actions due to novelty problems, even if MEXICA is copying a Previous Story. This situation lasts until MEXICA switches to the Reflective State and performs the evaluation process. By contrast, if MEXICA starts the story under the Reflective State, MEXICA performs immediately an evaluation process. If the initial action is included in one of the Previous Stories, the Novelty Guideline is set to Strict even if the story in progress is formed just by one event (the initial action). In some cases, this unnecessary rigidity can lead to an impasse.
In the following chapter this kind of situation is analysed.

B. Tensional Representation.

The evaluation of the Tensional Representation is divided into four stages: the first and second perform two different comparison processes. The third, based on such comparison, selects a Tensional Representation as a frame. Finally the fourth stage, with the help of such a frame, evaluates the story in progress and sends guidelines to the Engaged State.

The first process contrasts the transition of Tension between actions (qualitative comparison). The second, the value of the tension at each point in the vector of Tensional Representation (quantitative

comparison). The objective of these processes is to obtain the most similar Tensional Representation to the one of the tale in progress.

In order to explain the first comparison process the concept of Transitional Table is now introduced. In MEXICA, stories are sequence of events which occur at discrete times; that is, the first event occurs at time=1, the second at time=2, etc. A Transitional Table encodes the way the Tensional Representation changes between an action at time=t and the following action at time=t+1 for all events in a story. For example, if a tale has the following Tensional Representation [0, 5, 10, 10, 25, 15, 0] (i.e. the tension at t=1 is zero, at t=2 is 5, at t=3 is 10, etc.) its Transitional Table is [up up same up down down] (i.e. the tension from t=1 to t=2 goes up; the tension from t=2 to t=3 goes up, the tension from t=3 to t=4 is the same, etc.).

Thus, during the first process MEXICA compares the Transitional Table of the story in progress against the Transitional Table of each of the Previous Stories. This process works as follows. The Nth element of the Transitional Table of the story in progress is compared against the Nth element of one of the Transitional Tables in the Previous Stories, where N goes from 1 to the number of elements in the shortest of such tables. If in both tables the Nth elements have the same value one point is added to a variable called Result; this variable is used to calculate the similarity between them. If one of such elements has the value "same" and the other the value "up" or "down" zero points are added. Finally, if one has the value "up" and the other "down", one point is subtracted from Result. Thus, after comparing the story in progress against all the Previous Stories, the most similar tale will be that with the highest result.

Table 1	Table 2	Result
Up	Up	1
Down	Down	1
Same	Same	1
Up	Same	0
Same	Up	0
Down	Same	0
Same	Down	0
Up	Down	-1
Down	Up	-1

For example, let us suppose that the Previous Stories contains two tales with the following Tensional Representations: Story1 [0 0 5 10 15 25 25 0] and Story2 [0 15 10 15 25 25 0]. And that the Tensional Representation of the tale in progress is [0 5 10] (just three action have been produced for this new story). Now, the result of the first comparison between the story1 and the tale in progress is 1.

Story 1	New Tale	Result	Final Result
Same	Up	0	1
Up	Up	1	
Up			

Up Up Same Down			

and the result between the story 2 and the tale in progress is 0.

Story 2	New Tale	Result	Final Result
Up	Up	1	0
Down	Up	-1	
Up			
Up			
Same			
Down			

In this way, for this example the Tensional Representation of the tale in progress is more similar to the one in story1 than to the one in story2.

MEXICA performs a second comparison, this time to identify which of the Tensional Representations in the Previous Stories is the most alike to the one of the tale in progress. For this process MEXICA calculates the absolute difference in the value between the Nth element in the Tensional Representation of the story in progress and the Nth element of the Tensional Representation of one of the Previous Stories. The result of each operation is added to obtain a global result. The same process is repeated between the tale in progress and each of the tales in the Previous Stories. The comparison that produces the lowest global result indicates which Tensional Representation is the most similar to the one of the tale in progress. For instance, following with the previous example, after performing the second comparison process MEXICA obtains as a result that the Tensional Representation in story1 and story2 are equally similar to the one of the story in progress.

New Tale Story 1 Result	[0 5 10] [0 0 5 10 15 25 25 0] 0 5 5 = **10**
New Tale Story 2 Result	[0 5 10] [0 15 10 15 25 25 0] 0 10 0 = **10**

The third stage starts when MEXICA groups the Tensional Representations of those stories which obtained the highest value in the first comparison, and those which obtained the lowest value in the second comparison.

First comparison = [Story1]
Second comparison = [Story1 Story2]

Finally, the frame that is used to evaluate the story and produce the guidelines for the Engaged State is selected randomly, from a set resulting from the intersection between the elements obtained from the first comparison and those obtained from the second one.

Frame = [Story1] ^ [Story1 Story2]
Frame = Story1

If this set is empty, a frame is selected randomly from the group of Tensional Representations obtained from the first comparison.

MEXICA is now able to evaluate the tale in progress. Between all the options from the Previous Stories, the frame selected is just the most similar to the Tensional Representation of the story in progress; i.e., important differences can still exist between both of them. In MEXICA all knowledge about story-telling is represented in long-term memory; thus, MEXICA "taste" for a good story is based on the way Previous Stories were constructed. That is, for MEXICA an interesting story is that which resembles the Tensional Representation of a tale in the Previous Stories. Thus, although it does not attempt to copy the Tensional Representation selected as a frame, MEXICA uses it as a reference; when the story in progress strays too much from this reference MEXICA tries to push the story back. This procedure allows MEXICA to produce novel but well organised stories in terms of Tensional Representation.

Thus, the evaluation consists in comparing how similar the Transitional Table of the story in progress is to the Transitional Table of the frame. If it is not similar in at least 50% the tale in progress is classified as a story lacking interest; otherwise, the story is seen as producing enough interest to the reader. Of course, this percentage —known as Transitional Table Percentage— is modifiable by the user.
This classification is important for the filters. They eliminate actions based on the requirements established by the Novelty and Tensional guidelines. But in the case of the Tensional Guidelines is necessary to define which of them, the Permanent or the Temporal, will indicate to the filters the criteria to follow.
If the story in progress is classified as producing enough interest, the filters follow the requirements specified by the Permanent Tension until another evaluation process is executed. If the story in progress is classified as lacking interest, the filters follow the requirements specified by the Temporal Tension just during one selective process. After that, they follow the requirements established by the Permanent Tension until another evaluation process is executed.
The difference in the criteria established by each of the Tensional Guidelines will become evident in the following lines. However, it must be mentioned that under E2 operation mode just the Temporal Tension is utilised.

The Permanent Tension indicates if the intensity of the tension in the frame will raise, decrease or be the same the next time that MEXICA evaluates the story. Its function, in combination with the variable

Chances (see some paragraphs later), is to provide the Tensional Representation of the story in progress with a general direction to follow in order to produce interesting stories.

The possible values for the Permanent Tension are: tendency-up, tendency-down, neutral, end-story. When the Permanent Requirement is set to tendency-up, all those actions which produce a decrement in the value of the variable Tension to the Reader are eliminated by the filters from the group of possible next actions. When it is set to tendency-down, all those actions which produce an increment in the value of the variable Tension to the Reader are eliminated by the filters from the group of possible next actions. When it is set to neutral no action is eliminated. Finally, when it is set to end-story MEXICA ends the story in progress. By default the Permanent Tension is set to tendency-up.

The process of calculating the value of the Permanent Tension works as follows. Having T representing the number of actions produced so far in the story in progress and CtEg-Rf representing the constant which indicates the number of actions to be produced before switching from the Engaged to the Reflective State (i.e. before having another evaluation process again), MEXICA obtains the difference between the value of the tension in the frame at T + CtEg-Rf and at T; if the result is positive MEXICA sets the permanent requirement to tendency-up, if the result is negative the guideline is set to tendency-down, otherwise it is set to neutral. For instance, in the previous example the story1 was selected as a frame for the story in progress. Their Tensional Representations are

New Tale [0 5 10]
Story1 [0 0 5 10 15 25 25 0]

By default CtEg-Rf is set to 3 and for this example T is also set to 3. Thus, the value of the frame at T+CtEg-Rf is 25 and at T is 5

Story1(T+CtEg-Rf) = 25
Story1(T)= 5

In this way, the requirement is set to tendency-up. If the value of T+CtEg-Rf is bigger than the number of actions in the frame, the last element in such a frame is used to perform the operation. If the number of elements in the story in progress and the frame are equal, the guideline is set to end-story and MEXICA finishes the tale.

There is a special variable related to the Permanent Tension called Chances. Its function is to allow some actions, which do not satisfy the requirements of the Permanent Tension, being selected as the next event in the tale. It works as follows. When MEXICA retrieves possible next actions from long-term memory the filtering process is executed; so, all those possible next actions that do not fulfil the requirements of the Permanent Tension are deleted. However, if the variable Chances has a value greater than zero, e.g. 2, MEXICA does not delete two of those actions which do not satisfy the

Permanent Tension's requirements, giving them the opportunity of being selected as the next event in the story.
Each time MEXICA does not delete an action because there are some Chances left, the value of such a variable is decrement by one.
As commented earlier, MEXICA does not attempt to copy the Tensional Representation of Previous Stories; it just uses it as a general frame to guide the story. Through the use of the variable Chances new Tensional Representations (different to the ones in the Previous Stories) can be created.

The process of calculating the value of the variable Chances is now explained. The constant CtEg-Rf determines how frequently MEXICA switches to the Reflective State to evaluate the story in progress. MEXICA establishes that —as a maximum— half of the actions retrieved between two evaluation processes cannot fulfil the requirements of Permanent Tension. In this way, Chances is equal to the value of CtEg-Rf divided by two (Chances=CtEg-Rf/2). If CtEg-Rf is an odd number greater than one Chances is decrement by one, i.e. Chances = (CtEg-Rd/2) - 1.

The Temporal Tension is used when the story in progress is classified as lacking interest. Its function is to try to ensure that the Tensional Representation of the story in progress becomes more similar to the frame.
The possible values for the temporal requirement are: Up, Down or Hold. When it is set to Up the filters eliminate all those possible next actions which decrement the value of the Tension to the Reader. When it is set to Down the filters eliminate all those possible next actions which increment the value of the Tension to the Reader. When it is set to Hold the filters eliminate all those possible next actions which modify the value of the Tension to the Reader.
The process to calculate its value is now explained. Imagine that the story in progress has produced T actions. If the intensity of the tension in the frame at T+1 is greater than the value of the Tension to the Reader in the story in progress at T, MEXICA sets the Temporal Tension to Up. If the intensity of the tension in the frame at T+1 is less than the value of the Tension to the Reader in the story in progress at T, MEXICA sets the Temporal Tension to Down. If the intensity of the tension in the frame at T+1 is equal to the value of the Tension to the Reader at T, MEXICA sets the Temporal Tension to Hold. For example, if the story in progress and the frame has the following values:

Story in progress	[0 10 20]
Frame	[0 10 20 40 20]

MEXICA sets the Temporal Tension to Up since the intensity of the tension in the frame at T=4 is 40 while the intensity of the Tension to the Reader in the tale in progress at T=3 is just 20.
To avoid copying the Tensional Representation of the frame, the Temporal Tension is active just for one action; afterwards, the Permanent Tension turns active.

If the number of actions in the frame is equal to the number of actions developed so far, MEXICA finishes the story. In MEXICA, a Tensional Representation cannot operate as a frame if it does not contain at least the same number of actions as the story in progress; this condition is necessary in order to be able to produce the guidelines. If all Previous Stories are shorter than the story in progress, i.e. if all Tensional Representations in long-term memory are smaller than the one of the story in progress, MEXICA does not know how to continue the story, and the permanent guideline is set to end-story.

A final point about the evaluation process. Each time that MEXICA switches to the Reflective State the story in progress is evaluated, and each time it is evaluated a new process to select a frame is executed. Thus, if the tale in progress now has different characteristics a new frame is selected. That is, MEXICA might use more than one Tensional Representation from completely different stories as a frame during the development of a tale. This allows the system to create novel Tensional Representations.

4.3.3. Ending a Story.

The Engagement-Reflection cycle ends (i.e., the story is finished) when:

1. All characters in the story are dead.
2. When an unbreakable impasse is declared.
3. When the number of actions in the story in progress is bigger than the number of actions in all the frames (see Section 4.3.2.3-B).
4. When the maximum number of actions allowed in a story is reached. The maximum number of actions allowed in a story is a parameter definable by the user called Maximum Actions.

Another criterion consists in ending a tale when a degradation-improvement cycle is completed. This alternative permits the user to experiment with stories that include one or several of such cycles. The user activates this option through a parameter named Zero Tension (by default it is not active). However, there is a potential problem with it: stories with a very short degradation-improvement cycle are boring (see the boring stories in Section 6.1.3). A parameter named Tensional Increments is used to control this problem. It works as follows. Before ending the story the system verifies if the variable Tension to the Reader has increased its value N times in a row, where N is the value of the parameter Tensional-Increments (by default Tensional-Increments is set to 2 although it accepts values between 1 and 100). The logic behind it is that if the Tension to the Reader grows in multiple occasions, it is more difficult to produce a boring tale. Thus, if the Tension to the Reader in the story in progress increases its value Tensional-Increments times in a row and the parameter Zero-Tension is active, the next time that the value of the Tension to the Reader is equal to zero (i.e. when the degradation-improvement cycle is completed) the story is ended.

4.3.4 The Final Analysis.

Once a story is finished —i.e., when the engagement-reflection cycle ends due to any of the reasons explained above— the system performs a final analysis of it. That is, MEXICA revises the material produced to add a few finishing touches to the story.

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 coherent. 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. This process is divided in two parts.

A. Detecting Life at Risk, Health at Risk, or Prisoner.

The first part of the process consists in identifying if, at any moment in the story, there is a character whose life is at risk, or whose health is at risk or is a prisoner. If MEXICA finds a character that satisfies these characteristics, e.g. character B, the system keeps on examining the story to investigate if someone else saves, cures or rescues such a character B. If MEXICA finds this other character, the system has found a situation where it can insert a character's goal. For example, imagine the following story: *the princess went to the forest and the enemy kidnapped her. Jaguar knight realised about the kidnapping. He followed and killed the enemy. He rescued the princess and she fell in love with him.* In this story, the princess becomes a prisoner after the kidnapping. MEXICA detects this situation and finds that some actions later the knight rescues the princess. That is, MEXICA establishes a link between someone kidnapping the princess and jaguar knight rescuing the princess. Thus, the system inserts an action, just after the knight realises about the kidnapping, where the goal to rescue the princess is set. The final story looks as follows: *the princess went to the forest and the enemy kidnapped her. Jaguar knight realised about the kidnapping. Jaguar knight knew that princess' life was at risk and that jaguar knight had to do something about it. He followed and killed the enemy. He rescued the princess and she fell in love with him.*

The extra action introduces the goal of saving the princess, which makes clearer jaguar knight behaviour. It also permits introducing some suspense in the story by delaying the kidnapping resolution. In general, these kind of additional actions improve the story.

When MEXICA detects a situation like the one just explained, it inserts the following standard Primitive Action:

```
ACT
Life_Risk_1 2 ;  Character A is the  hero.  Character  B is the victim.
TEXT
@A knew that @B's life was at risk and had to do something about it.
```

Where character A saves B. This Primitive Action does not include preconditions or postconditions. Its function is to represent the moment when character A decides to save character B. As with all the other Primitive Actions, MEXICA substitutes this Primitive Action with its associated text to produce the final version of the story. The user can modify the given text or add new ones.

The same process occurs when the system detects that the health of a character (e.g. character B) is at risk MEXICA analyses the story and if some other character cures character B the following standard Primitive Action is inserted:

ACT
Health_Risk_1 2 ; Character A is the hero. Character B is the victim.
TEXT
@A knew that @B could die and that @A had to do something about it.

And finally, MEXICA uses the same method when it detects that character B is kidnapped and someone else rescues such a character B. In this case, the following standard Primitive Action is inserted:

ACT
Pr_Free_1 2 ; Character A is the hero. Character B is the victim.
TEXT
Although it was very dangerous @A promised to Quetzalcoatl —The God between the Gods— to rescue @B.

B. Detecting Love Competition and Clashing Emotions.

This part of the process detects if —in any of the characters in the story— the tension Love Competition or Clashing Emotions becomes active. When this occurs, MEXICA inserts an action to make explicit these tensions. In this way, the story gets a special drama. The following is an example of a story: *Eagle knight rescued princess. The tlatoani felt an enormous gratitude for the knight and rewarded him. Suddenly, tlatoani was informed that eagle knight was responsible for the murder of many children during the last expedition. Tlatoani decided to exile eagle knight forever.* After MEXICA performs the final analysis, the story looks as follows: *Eagle knight rescued princess. The tlatoani felt an enormous gratitude for the knight and rewarded him. Suddenly, tlatoani was informed that eagle knight was responsible for the murder of many children during the last expedition. Tlatoani was emotionally tied to eagle knight but tlatoani could not accept eagle night's behaviour. What should tlatoani do? Tlatoani decided to exile eagle knight forever.*
When MEXICA detects a Clashing Emotion it inserts a standard Primitive Action just after the event that triggered the tension:

ACT
CLASH_EMOTION_1 2
TEXT
@A was emotionally tied to @B but @A could not accept @B's behaviour. What should @A do?

MEXICA distinguishes three different types of Clashing Emotions. The first has been just described. The second occurs when character A has Clashing Emotions towards character B, but previously character B had already developed Clashing Emotions towards character A. For example, continuing with the story above, the tlatoani develops the first type of Clashing Emotions when he realises that the knight is a murder. When the knight realises that he has been exiled by the tlatoani (the man who rewarded him), the

knight develops the second type of Clashing Emotions. In the first type, the Clashing Emotion is triggered by the contradictory and disappointing knight's behaviour (he is a hero but also a murder). In the second type, the Clashing Emotion is triggered by tlatoani's reaction to knight's behaviour. Detecting and making explicit these subtle differences improve the quality of the stories.
When MEXICA detects the second type of Clashing Emotions it inserts the following standard Primitive Action:

ACT
CLASH_EMOTION_2 2
TEXT
@A was shocked by @B's actions and for some seconds @A did not know what to do.

The third case occurs when a character has Clashing Emotions towards itself. For example, imagine a warrior that has an accident. A man arrives and cures him. Suddenly the warrior realises that the man is an enemy and kills him. Thus, the warrior develops clashing emotions towards himself. In this case MEXICA inserts the following standard Primitive Action:

ACT
CLASH_EMOTION_3 1
TEXT
@A was emotionally devastated and confused, and was not sure if what @A did was right.

MEXICA uses the same process described above to insert actions when it detects a Love Competition between characters. Thus, in this case, it inserts the following standard Primitive Action:

ACT
Love_Competition_1 2
TEXT
@A and @B were in love with the same person, and none of them would give up.

Like in the case of Clashing Emotions, different types can be distinguished. For example, when one of the characters is aware of the Love Competition but the other not (e.g. when a knight is in love with a lady but he does not know that other knight also wants her). When the characters involved in the Love Competition belong to the same social class (it is not the same when two knights are competing for the love of a woman, and when a knight and a farmer are competing for the same woman), etc.

4.4 Learning in MEXICA.

The learning mechanism in MEXICA is simple. Knowledge in long-term memory mainly comes from the set of Previous Stories; so, when a new tale is developed by MEXICA the user can add it to the file of Previous Stories. In this way, all the information produced by the new tale is incorporated into the structures in long-term memory each time MEXICA loads the Previous Stories. This is important because an impasse occurs when an (unusual) context cannot match an atom; so, if the impasse is

overcome, the next time the file of Previous Stories is loaded into memory such a context is transformed in a new atom with its associated set of possible next actions. In this way, when in the future MEXICA faces a similar situation the impasse will be avoided. What happens when an impasse cannot be broken? Because Previous Stories are encoded in a text file the user has the opportunity to provide some examples which illustrate how to break the impasse.

4.5 User Interface.

The user interface consists of a set of windows, which either open menus, run processes, display the new tale on the screen or allow the user to modify different parameters in the system.

A. Setting up Variables.

MEXICA has twenty three parameters, which can be manipulated by the user. They are all assigned default values when the system starts, so they do not have to be modified. However, if the user decides to experiment with all the different options, the possibility of changing them is always present. Table 4.1 shows a list of all parameters modifiable by the user ordered alphabetically. Such table includes the name of the parameter, the section where the reader can find an explanation of their function, the default value assigned to them and a brief description of their role.

Parameter	Section	Default Value	Description
ACAS-Constant I, II, III	4.3.1	50%, 70% and 90%	Minimum percentage required to match an atom through the ACAS-Process.
CtEg-Rf	4.3	3	Number of actions produced during the Engaged State before switching to the Reflective State.
Equivalent Constant	4.3.2.2	50%	Minimum percentage required to consider an action equivalent to another one.
Forbidden Characters	4.3.1.2-A	Half-Active	Prevent some of the characters in the context from being banned.
Initial State	4.3	Engaged State	Indicates the state under which MEXICA starts to develop a tale.
Logical Action	4.3.1.2-B	Not active	Activates or deactivates the constraints for logical actions.
Maximum Actions	4.3.3	20	Maximum number of actions allowed in a new story.
Medium Novelty Constant	4.3.2.3-A	75%	Minimum percentage necessary to satisfy the requirement of Medium Novelty.
Novelty-Percentage	4.3.2.3-A	50%	Maximum percentage of similarity authorised to exist between two tales.
Operation Mode	4.3	ER2	Active Operation Mode for the developing of a tale.
Strict Novelty Constant	4.3.2.3-A	15%	Minimum percentage necessary to satisfy the requirement of Strict Novelty.
Tension Dc	4.2.3.3	30	Tension produced due to a Dead Character.
Tension Hr, Lr, Pr, Ce, Pd	4.2.3.3	20	Tension produced due to Health in Risk, Life in Risk, Prisoner, Clashing Emotions or Potential Danger.
Tension Lc	4.2.3.3	10	Tension produced due to Love Competition.
Tensional Increments	4.3.3	2	Number of times that the variable Tension to the Reader must increment its value in a row to activate the parameter Zero Tension.
Transitional Table Percentage	4.3.2.3-B	50%	Minimum percentage required to classify a story as producing enough interest.
Zero Tension	4.3.3	Not active	When this parameter is active the story ends the next time that the Tension to the Reader has a value equal to zero.

Table 4.1 Parameters definable by the user (notice that rows 1 and 13 include more than one parameter).

B. Displaying the new Story.

MEXICA has the option of displaying on the screen the latest story created.

C. Saving a New Story

The user can incorporate the latest story created by the system to the Previous Stories. In this way, new members are added to the Concrete, Abstract and Tensional Representations in long-term memory and MEXICA can utilise them to produce new tales. The process consists in appending the new story to the text file of Previous Stories.

D. Re-loading Previous Stories.

Each time MEXICA starts all structures in long-term memory are created. When the user adds new tales to the file of Previous Stories through the option of saving a new story, they are not immediately incorporated to such structures. It is necessary to reset the program or re-load the Previous Stories. That is, the option of re-loading the Previous Stories runs again the process that constructs all structures in long-term memory; in this way, without leaving the program, the structures produced by such new tales are included in memory.

E. Printed Reports.

MEXICA generates three reports (some examples can be found in Appendixes E and F). The first traces step by step the way Previous Stories are processed; it is very useful to study and analyse the way characters' contexts behave and therefore the way atoms are created in long-term memory. It also includes the graphic of the Tensional Representation of all Previous Stories. The second is a map of the Abstract Representation, which includes all atoms and their position. Finally, the third traces the development of a new tale. For questions of space, only examples of the second and third report are presented in thesis. However, notice that the first and third reports are the same; the only difference is that the former reports the Previous Stories and the latter the story in progress.