ERIC: A Generic Rule-based Framework for an Affective Embodied Commentary Agent

Martin Strauss[*]
DFKI, Embodied Agents Research Group
Campus D3 2, 66123 Saarbrücken, Germany
martinstrauss@google.com

Michael Kipp
DFKI, Embodied Agents Research Group
Campus D3 2, 66123 Saarbrücken, Germany
michael.kipp@dfki.de

ABSTRACT

We present ERIC, an affective embodied agent for realtime commentary in many domains. The underlying architecture is rule-based, generic, and lightweight – based on Java/Jess modules. Apart from reasoning about dynamically changing events, the system can produce coherent natural language and non-verbal behaviour, based on a layered model of affect (personality, mood, emotion). We show how reasoning, template-based natural language generation and affective appraisal can be implemented within the same rule-based paradigm. To make the system domain independent we worked on two different domains, a virtual horse race and a multiplayer tank battle game. We empirically evaluated the genericness of the system by measuring the effort it takes to change the domain, and discuss the results.

Categories and Subject Descriptors

I.2.1 [**Artificial Intelligence**]: Applications and Expert Systems;
I.2.0 [**Artificial Intelligence**]: General — *Cognitive Simulation*;
I.2.7 [**Artificial Intelligence**]: Natural Language Processing —
Language generation

General Terms

Design, Human Factors

Keywords

Commentary Agents, Affect, Event Recognition, Natural Language Generation, Discourse Coherence, Embodied Conversational Agents, Embodied Characters

1. INTRODUCTION

An embodied character that automatically commentates on real or virtual events, such as sports, multiplayer games or a tour of a city, is a future yet highly desirable vision. As well as describing events verbally, embodied commentators can point at locations non-verbally, highlight important moments with emotion, and both motivate and entertain users with their "personality" [14]. The wide

range of modalities of an embodied agent (speech, intonation, face, gesture, posture) allows them to smoothly transport information in parallel: either to convey complementary information, or to increase redundancy for the sake of robustness, making the information accessible to many people (including those with poor hearing or eyesight) and in many situations (for example, with or without video/audio of the original events) [13, 2].

Embodiment helps people accept new technology [15], gives them orientation by providing an 'anchorman' [25] and helps them in navigation when travelling in a 3D virtual world [23]. Endowing an embodied agent with emotions is especially important for making the character believable [22, 30].

Acknowledging the importance of embodied commentary agents, the GALA Race Reporter Challenge was established[1]. This is a competition where embodied agents created by teams of students commentate on the same horse race event, generated by a publicly available horse race simulator.

Commentating a sports event is a challenging task because many events of varying importance happen in a potentially short period of time. The "importance" of events is context-dependent. Also, sports events are usually very emotional, which must somehow be reflected in the commentator's behaviour. From a wider perspective, the challenge is to make a commentator system as generic (i.e. domain independent) as possible, to make it reusable for other scenarios (e.g., horse races, RoboCup, or multiplayer combat games).

We suggest the ERIC[2] framework for the design of such a commentary agent [26]. The framework consists of several Java modules that each encapsulate a Jess rule engine [8]. All major "cognitive" tasks (reasoning, affective appraisal, natural language generation) are processed within the rule-based expert system paradigm. This has several advantages: first, there is a single overall knowledge representation language compatible with the processing framework. Thus there is no engineering overhead of defining interfaces or connecting to middleware. Also, it allows the rapid assembly of early demonstrations for user studies in the development cycle, and makes authoring accessible to non-experts (cf. [11]).

To make the framework domain independent, ERIC was used in two domains: the horse race simulator used in the GALA Challenge, and the dTank multiplayer tank battle game [17].

A wide range of previous work has dealt with the application of virtual agents as a user interface [24]. COHIBIT [18, 9] is an ambient intelligence edutainment exhibit implemented in the VirtualHuman framework. Visitors interact with the system by assembling a puzzle from physical car pieces; the visitors' actions are observed

---

[1]Gathering of Animated Lifelike Agents, http://hmi.ewi.utwente.nl/gala.
[2]Embodied Real-time Intelligent Commentary, http://www.martinstrauss.id.au/eric/

by two embodied agents who are able to comment on the actions, provide advice and additional information, or make small-talk. The COHIBIT system uses a type of augmented transition network to model the dialogue interaction. The networks are easy to author and read, to maintain and debug: in contrast, low maintainability can be a drawback of large rule-based systems such as ERIC. However, using a rule-based system for all levels of processing has the advantage that the knowledge representation and behaviour generation are all in the same homogeneous framework, which also makes the system easier to maintain.

Two early systems that automatically generate commentary based on computer vision input are VITRA and Rocco [7, 1]. Rocco generates multimedia commentary output from three layers of processing: low-level visual analysis, high-level scene analysis producing a scene description (e.g. player positions and actions), and multimedia presentation generation, which plans the multimedia output. While both systems feature more complex reasoning over the input knowledge (plan recognition) they do not deal with issues of believability of an embodied character and the underlying emotions. These are the focus in ERIC.

The Virtual Human Presenter [20] is a system that animates an embodied agent from annotated input speech text for presentation scenarios such as a weather report or the news. However, it neither deals with how the information is generated nor does it integrate emotions.

Our ERIC framework specifically contributes the following novelties to the field:

- Embedding affective appraisal in Jess with a de-coupling of causal relations from goals and desires
- Embedding template based natural language generation in Jess, incorporating centering for coherent discourse, which makes it easy to author new (language) content
- Qualitative and quantitative analysis of the domain independence of our system

We first give an overview of ERIC's architecture before focussing on the three core modules and discussing the issue of domain-independence.
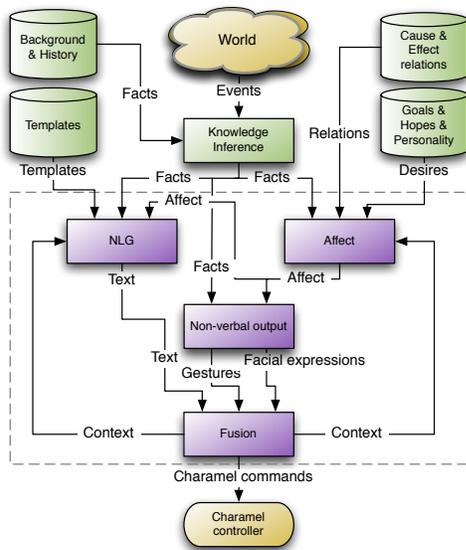

Figure 1: The system architecture of ERIC

2. SYSTEM ARCHITECTURE

The modular architecture of the system is shown in Figure 1. Components shown in yellow are interfaces to the outside world and the embodied agent; components in green are designed to be domain specific, and components in blue are domain independent. Each rectangle represents a standard Jess/Java module: these modules differ only in their expert system rules. This diagram is somewhat simplified: non-verbal output is in fact generated by multiple modules, one for each modality. The fusion module is responsible for matching the generated non-verbal modalities to each other and to the speech output, as well as resolving semantically conflicting outputs. In this paper, we focus on three modules: knowledge inference, affect and NLG.

3. KNOWLEDGE MODULE

The knowledge module of ERIC is responsible for elaborating the limited incoming information from the domain interface into a rich world model which then forms the basis for the agent's natural language and affect generation. The knowledge module is implemented as an expert system in Jess [8]: information from the world interface is represented as Jess facts, and then by logical inference over these facts (via Jess rules) many more facts are inferred from the input information.

The dynamic input information sent to ERIC once per second is shown in Table 1. The knowledge module's elaboration may also use static background knowledge that has been specified when configuring the agent (Table 2). From this information the knowledge module can deduce facts and events such as the order of the horses in the race, or a tank aiming its guns at another tank (Table 3).

```
(defrule knowledge::fallen
  "a horse has fallen when its speed is 0"
  (speed (timestamp ?t1) (horse ?h1) (speed 0))
  (speed {timestamp == (- ?t1 1)}
    (horse ?h1) {speed > 0})
  (not (raceover))
  =>
  (assert (fallen (timestamp ?t1) (horse ?h1)))
  )
```

Figure 2: An example of a Jess rule

RaceSim	dTank
Timestamp	Location of each stone on the field
Location of each horse	The name of each tank
Speed of each horse	The location of each tank
	The orientation of each tank
	The orientation of each tank's gun turret
	Each tank's health
	Each tank's shields status
	Each tank's ammunition levels
	A tank is commanded to fire
	A tank is commanded to raise its shields
	A tank is commanded to move forward
	A tank is commanded to rotate
	A tank is commanded to turn its turret

Table 1: Sparse input data sent by the two domains every second

The facts generated by the knowledge module make up the world model, and are sent to the other modules of ERIC for generation of affect, language and gesture. All incoming facts are timestamped. This relieves ERIC from having to retract facts that are no longer true, and also allows ERIC to detect patterns over time, or correct

past predictions that turned out to be incorrect: for example, when commenting the RaceSim ERIC predicts that a horse will soon overtake another horse, and can also observe that the pursuing horse failed to overtake as predicted.

RaceSim	dTank
Odds of each horse	The current weather
Each horse's name	Each tank's commander
Each horse's age	A summary of a tank's past success
Each horse's handicap	
The length of the racetrack	
The condition of the track	
Each horse's preference in track condition	
A past win of a horse	

Table 2: Sample background knowledge facts

RaceSim	dTank
Whether the race is over	Each tank's kills score
The distance between adjacent horses	A summary of the terrain style
A horse is about to overtake another horse	A row of stones on the map
A horse has overtaken another horse	A column of stones on the map
A horse failed to overtake after being about to	The map quadrant in which each tank is located
A horse has taken the lead	Relative location of pairs of tanks
Each horse's cardinal position	A tank is aiming at another tank
A horse has fallen over	A tank is under fire
A horse has gotten up again	A tank has been destroyed
An increasing gap between adjacent horses	A tank's location is unchanged
A decrease in the speed of a horse	A tank's health is unchanged

Table 3: Sample generated knowledge facts

4. AFFECTIVE APPRAISAL

The emotional dimension enhances the character's believability and the amount we engage with the character [4]. Emotions can also be of informational value (e.g., excitement signalling that a decisive action is happening) or convey the preferences of the commentator (e.g. preference for a horse or player) [22, 30].

ERIC generates affective responses by assigning appraisals of emotion-eliciting conditions (EECs) to facts in the world model based on goals and desires. These appraisals are used by ALMA [3] to generate an affective state according to the OCC cognitive model of emotions [21]. The generated moods and emotions are expressed by ERIC in his verbal and non-verbal output.

4.1 Affective Computation in ALMA

ERIC's emotional state is modelled by ALMA [3]. ALMA models three distinct types of affect: emotions (short-term), moods (medium-term) and personality (long-term). Emotions are usually bound to a specific event, action or object, and decay through time; moods are generally not related to a concrete cause, and are more stable than emotions; and personality is a long-term description of the agent's affective behaviour.

Personality is modelled using the Big Five model of [5], which defines a personality along five factors: openness, conscientiousness, extraversion, agreeableness and neuroticism. These factors are used to calculate an initial mood; also, the values of these traits affect both the intensity of the character's emotional reactions, and the decay of his emotion intensities.

Mood is modelled as an average of emotional states across time. Each mood is described by a value on the three scales pleasure, arousal and dominance, according to the PAD model of [16]; these three values range from -1.0 to 1.0, and form a three-dimensional mood space (PAD space). Thus each mood represents a point in the PAD space. The intensity of the mood is the magnitude of the vector describing its PAD point.

Since the character's mood is changed, not set, by his emotions, an initial mood must be calculated from the personality traits. Using a mapping of emotions to PAD space, the character's emotion changes his mood according to the *pull and push mood change function*. The specifics of mood calculation are described in [3].

Emotions are deduced from stimuli according to the OCC cognitive model of emotions [21]. The inputs to this model are appraisals of the world, called emotion-eliciting conditions. Events, actions and objects are appraised: desirability, likelihood and realisation of events, liking of others affected by events, praiseworthiness and agency of actions, and appealingness of objects. The value of each appraisal, along with the personality model, is used to compute the intensities of the emotions.

4.2 Rule-based Appraisal

ALMA expects as input appraisal values formulated as emotion-eliciting conditions. ERIC makes these appraisals by comparing events, actions and objects observed in the world against the agent's goals and desires. Cause-effect relations allow us to appraise events, actions and objects that are not specified in the goals and desires.

4.2.1 Goals and Desires

We specify events or actions we desire (*goals*), events or actions we desire to avoid (*antigoals*), and objects we like and dislike. Additionally, the agent maintains a set of beliefs about other actors' goals and desires, to enable him to judge the desirability of events for the players in the scenario.

If a goal occurs, this is appraised positively (*desirable* for events, or *praiseworthy* for actions); if an antigoal occurs, this is appraised negatively (*undesirable* for events, or *unpraiseworthy* for actions). When an object we like appears in the discourse state (i.e. it is mentioned in the commentary), it is appraised positively (*liking*); conversely, when an object we dislike appears, it is appraised negatively (*disliking*).

Appraisals of agency and realisation are made by observing the world model. Agency is appraised by identifying the agent of each action. An event is appraised as realised if it has occurred in the world.

The goals and desires are specified in a separate configuration file to the main affect module, since they are dependent on the world model and thus domain-specific. They can easily be altered to match both a change in domain and a change in intended audience.

4.2.2 Cause-Effect Relations

In order to reason about events and actions that are not directly part of our goals, but still related to (or influencing) our goals, we codify the relations between non-goal events, actions and objects and goals, using the four relations: *leadsto*, *hinders*, *supports* and *contradicts*. The first two relations model causality:

- a leadsto b if a increases the likelihood of b, and
- a hinders b if a decreases the likelihood of b.

The second two relations model logical deduction or belief:

- a supports b if knowledge of a supports belief in b, and
- a contradicts b if knowledge of a contradicts belief in b.

These relations are used to propagate appraisals of desirability, praiseworthiness, and likelihood in the following way:

- a leadsto b
  - a inherits some of b's desirability/praiseworthiness[3], and
  - the likelihood of b is increased when a occurs.
- a hinders b
  - a inherits the opposite of b's desirability/praiseworthiness, and
  - the likelihood of b is decreased when a occurs.
- a supports b
  - if a is observed then b also has realization $=$ true
- a hinders b
  - if a is observed then b has realization $=$ false

Applied recursively, these rules allow us to make appraisals of all events and actions related to our goals.

Like the reasoning in the knowledge module, this reasoning is quite naturally expressed in the form of rules; thus it is implemented as a rule-based expert system in Jess. The expert system generates appraisals of events, actions and objects in the world, which are then passed to ALMA.

4.3 Expression

The agent's affective state can be expressed in one of three ways:

- his selection of words and phrases from the utterance database to fill a NLG template,
- his hand and body gestures, and
- his facial expressions.

A prosody module has been outlined and is foreseen as future work.

4.3.1 Utterance Selection

Each of ERIC's utterances can be tagged with emotional states, and are then only used when the character is in one of the specified states. The agent is able to generate emotionally loaded referring expressions and pronouns – for example "The wonderful Carmine" in Figure 4. Also, the agent can comment directly on his current affect – for example "I hope this goes well!" in Figure 4.

4.3.2 Gesture

ERIC features two types of gesture. First, we specify a set of "idle" gestures which the character plays back when no other gesture has been specifically requested. These gestures are varied according to ERIC's emotional state, so that ERIC moves his hands more vigorously when he is excited, for example. Then, we can command the agent to perform a specific gesture. The gesture module contains a number of rules triggering gestures based on specific emotions: for example, disappointment will be expressed in a shake of the head.

The Paul model (Figure 3) supports over three hundred different gestures, ranging from pointing and waving gestures to more complex gesture sets such as "more or less", "speaking" or "waiting".

---

[3]In these calculations, desirability and praiseworthiness are interchangeable: if a is an action and b is an event, a's praiseworthiness will increase proportional to b's desirablity.

Figure 3: The Charamel character "Paul"

4.3.3 Facial Expression

Paul provides a variety of facial expressions which are used to depict ERIC's emotions. The facial expression module is a map of emotions onto facial expressions: when ALMA generates any of the mapped emotions, the corresponding facial expression command is sent to Charamel. The Paul model supports 13 different facial expressions, grouped into categories such as neutral, happy, disgust and fear.

5. TEMPLATE-BASED NLG & CENTERING

The natural language generation module of ERIC uses a template-based algorithm to generate candidate utterances from a world model of Jess facts from the knowledge inference module and an emotional state from the affect module. A single best utterance is chosen from these candidates according to their salience and discourse coherence and sent to the text-to-speech system.

5.1 Template-based NLG

Template-based natural language generation systems map their non-linguistic input directly to the output text, rather than via some intermediate representation [28]. They have a number of advantages over more complex systems: in particular, their simplicity makes them easier to implement and maintain. Also, the only domain dependent parts of a template-based natural language generation system are the templates themselves, and since these are expressed in terms of the output text, it is easy to author a new set of templates for any domain, even for non-experts.

5.1.1 Template Structure

The NLG module in ERIC uses templates similar to the D2S system [27]. Templates in ERIC consist of a set of conditions, a discourse state, a salience and an index into an utterance database. The conditions are both on the knowledge state and the world state. They are expressed as four sets of Jess facts: facts that must be known, facts that must be unknown (for the knowledge state), facts that must be true, and facts that must be false (for the world state). The discourse state consists of a single backwards-looking center and multiple forward-looking centers for each template; these are used by the discourse coherence mechanism. The salience value represents the salience of the facts expressed by the template; this is used to ensure that the most salient facts are reported.

5.1.2 Utterance Database

The utterance database consists of a list of phrases or utterances, one or more for each template, that are output by the NLG system

when the template is activated. The utterances in the database contain slots for referring expressions and other variables, for example horse distances or tank orientations. These slots are filled by the templates before the utterances are output.

Except for these slots, the utterances are stored as flat text. This means that ERIC's NLG is not capable of dynamically generating surface realisations; however the task of authoring templates is much simplified.

5.1.3 Referring Expressions

Most of the utterance slots are bound by the facts on which the template is conditional: the distance between horses, or the health points of a tank, for instance. For referring expressions however we desire more variety; thus we use a more complex process to generate referring expressions.

The entities for which we need referring expressions are identified in the template. The generated referring expressions are then stored in variables corresponding to the slots in the utterance. The referring expressions are generated from the world state and knowledge state by Jess rules written by the domain author: for example a horse can be referred to as "the horse in blue" or "the favourite", depending on the current world state and what the listener already knows about that horse.

5.1.4 Implementation in Jess

The process by which template-based NLG generates text – find all matching templates, and generate their candidate utterances – is the same as the execution cycle of an expert system – select all matching rules, and execute their right-hand sides. Thus it is natural to implement each template as a rule in an expert system, with the conditions forming the left-hand sides of the rules, and the utterance output (as well as necessary updates to the knowledge and discourse states) forming the right-hand side of the rules.

To keep the utterance database separate from the templates, we store it in a Jess facts declaration. Referring expressions are also represented as Jess facts; however the referring expression facts are generated by rules rather than declared in advance.

The firing of the rules corresponding to the NLG templates produces a set of candidate utterances, again in the form of Jess facts. From these, a single most coherent utterance is selected for output.

5.2 Discourse coherence using centering

The usual approach to the question of ordering utterances into a coherent discourse is to use a discourse planner. However offline planning requires all the utterances to be available at the time of planning: this is not feasible for ERIC, since he must spontaneously react to ongoing events. Discourse coherence reasoning could be performed online using dynamic replanning; however this is complicated and not particularly fast, and thus unsuitable for ERIC. To be satisfactorily reactive, ERIC must be able to choose a suitably coherent next utterance from the list of available utterances using only local information.

Centering Theory[6] claims that we can describe the global coherence of a discourse entirely in terms of local coherence relationships. This implies that offline text planning is unnecessary for generating a coherent discourse: we merely need to ensure that the discourse is locally coherent from utterance to utterance. This is precisely what we are looking for in ERIC.

We consider a discourse as a set of utterances U. Centering Theory assigns each utterance U_n a number of semantic focal points called *centers*: a single *backward-looking* center $C_b(U_n)$ and a partially ordered set of *forward-looking* centers $C_f(U_n)$. The forward-looking centers are partially ordered by *prominence*, that is, the

likelihood that a forward-looking center of U_n will be the backward-looking center of U_{n+1}. In effect, the backward-looking center of a sentence describes its topic, and the forward-looking centers describe possible topics for a coherent following sentence. Given this semantic information, Centering Theory defines three relations between subsequent utterances, ordered by decreasing coherence:

Center continuation $C_b(U_{n+1}) = C_b(U_n)$ and $C_b(U_{n+1})$ is the most highly ranked element of $C_f(U_{n+1})$

Center retaining $C_b(U_{n+1}) = C_b(U_n)$ but $C_b(U_{n+1})$ is not the most highly ranked element of $C_f(U_{n+1})$

Center shifting $C_b(U_{n+1}) \neq C_b(U_n)$.

Coherence processing in ERIC is inspired by Centering Theory. Whereas in Centering Theory the forward-looking centers are partially ordered, in ERIC they are unordered; this makes authoring templates simpler, since an author merely needs to identify the forward-looking centers of an utterance, not order them as well. The three relations used to compare utterances are also simpler; in decreasing order of coherence, the relations are:

Center retaining $C_b(U_{n+1}) = C_b(U_n)$

Smooth shifting $C_b(U_{n+1}) \in C_f(U_n)$ but $C_b(U_{n+1}) \neq C_b(U_n)$

Abrupt shifting $C_b(U_{n+1}) \neq C_b(U_n)$ and $C_b(U_{n+1}) \notin C_f(U_n)$

For example, consider the sentences:

(1) a. Carmine has overtaken the favourite.

    b. She is in first place.

    c. Eben has overtaken the favourite.

    d. The favourite is in third place.

    e. Topaz is wearing green today.

The sentences 1a and 1b share a backward-looking center ("Carmine"), so they are related by center retaining. The two sentences 1c and 1d are less coherent: they do not share a backward-looking center ("Eben" for 1c, "the favourite" for 1d), but the backward-looking center of 1d is in the forward-looking centers of 1c ({"Eben", "the favourite"}), so their relation is smooth center shifting. Least coherent are the two sentences 1d and 1e. The backward-looking center of 1e ("the favourite") is neither the same as that of 1d ("Topaz") nor is it in its forward-looking centers ({"the favourite"}), so here we have abrupt center shifting, the least coherent relation.

Once all the candidate utterances are generated by the template-based NLG module, they are sent to the fusion module. There, these relations are assessed by comparing the forward-looking and backward-looking centers of the previously spoken utterance with the backward-looking centers of each candidate utterance; the candidate utterance with the strongest coherence is then spoken.

6. DOMAIN INDEPENDENT ASPECTS

The ERIC agent was designed to be a highly reusable framework; thus a clear separation of domain dependent and domain independent components was an important goal. We aimed to keep as much of the agent as possible domain independent, to minimise the amount of work required to reuse the ERIC framework.

To evaluate the degree of domain independence of the agent, we quantify the amount of effort required to implement the framework for a new domain. As a baseline, we compare this effort against the effort required to implement ERIC from scratch (for the RaceSim domain).

We have used two measurements to quantify the effort: final lines of code, and final number of Jess definitions. Software size is directly related to development effort, and easier to quantify [29, Chapter 7]. We compared the size of the code that was common to both domains with the size of the code that was unique to each domain, to avoid counting code modified during improvements to domain independence as domain dependent code.

The results of this tally are shown in Table 4 (in lines of code) and Table 5 (in Jess declarations).

6.1 Aspects

ERIC was designed for a clear separation between domain dependent and domain independent aspects. The knowledge reasoning is necessarily domain dependent, since it describes the domain. However once a world model has been created, processing of this model should be domain independent: the natural language generation module, affect module, and all non-verbal output modules are domain independent. Domain dependent information needed for such processing is stored in separate components: for example, the templates for natural language generation.

6.2 Evaluation

Three modules required substantial changes: the knowledge module, the language module and the affect module. The remainder of the modules could be reused almost without any modification: although the gesture module had some domain dependent rules, these were few and small since the bulk of the gestures are generated by the domain independent rules. Within the domain dependent modules, the intended separation between domain specific and domain independent information is evident.

The evaluation shows some reuse of code in the knowledge module: this is unexpected, as the knowledge module was intended to be entirely domain specific. The code reuse here is a result of reasoning common to both domains, rather than domain independent reasoning.

In the language module, the utterance database and templates were domain specific, whereas the code generating Jess rules from template specifications was domain independent. Although the referring expression generation was also intended to be domain independent, it was found to rely on some domain dependent assumptions: this is one potential avenue for improvement.

Within the affect module, the specification of goals and desires, and the rules observing events, actions and objects in the world model were domain specific, whereas the ALMA interface was domain independent. The causality relations were not entirely domain independent: our evaluation did not distinguish between the specification of the relations, which is necessarily domain dependent since it describes the domain, and the rules that propagate appraisals across relations, which are domain independent and form the bulk of the rules.

6.3 Feedback from GALA

The ERIC agent was submitted to GALA '07[4]. As part of this submission, a video was produced of the agent commentating a three minute horse race. This video was evaluated by two independent judges. As well as their feedback, ERIC won a judge's prize in the race reporter category[5].

Overall, the judges found ERIC to be engaging and believable. Three points were identified as needing improvement: the selection of 'idle' gestures, the lack of prosody, and occasional delays between an event and its commentary.

The random process used to select gestures for 'idle' behaviour can sometimes appear unrealistic. The behaviour selection could be improved by generating behaviours synchronised to other modalities such as ERIC's speech [10, 19].

Prosody is a important channel for communicating affect. A module to generate prosody was foreseen as future work, but not implemented in the GALA submission. Adding such a module would improve the perceived affectivity of the commentator.

If an events in the world occurs while ERIC is speaking, he will first finish his current utterance before commenting the event. This can lead to delays between an event and its commentary. A solution to this would be to enable ERIC to interrupt his commentary if a new event is sufficiently significant. Another solution would be to predict such significant events, and then avoid beginning an utterance just before the significant event.

7. CONCLUSION

We presented the ERIC framework for providing running commentary on a continuous event in real-time. ERIC has been implemented to commentate a simulated horse race and a tank combat game, and won the GALA 2007 Award. We showed how reasoning, template-based natural language generation, and affective appraisal can be implemented within the same rule-based paradigm, based on Java and Jess. The template-based natural language generation system is capable of generating anaphora and coherent discourse. A layered model of emotions, mood and personality guides output generation; affect is generated from dynamic appraisal of events, actions and objects against goals and desires.

Particular focus has been given to make the framework generic, i.e. easy to change domains. For example, the same system could commentate Robocup matches, computer games, or play the role of tourist guide during a self-guided tour of a city. To keep ERIC generic, domain-specific knowledge is kept separate from domain-independent reasoning (for example, goals/desires and cause/effect relations are separate from the affective appraisal rules). We identified and isolated the two major domain-specific areas: the knowledge module and the NLG templates. The affect module also requires some minimal modification. We showed that the effort required to change to a new domain is not only quite reasonable but also concentrated in the isolated modules.

As future work, a prosody module and more elaborate gesture generation [10, 19] is planned to improve the agent's non-verbal output. To improve the agent's reactivity, we plan to add a mechanism for allowing the agent to either interrupt current utterances with more important ones, or avoid uttering the less important ones entirely in this situation.

Also, further evaluation is planned: on the one hand, user studies to investigate how users perceive ERIC's output; on the other, an extended domain independence evaluation involving a range of developers comparing the effort required to author an ERIC commentator with that required to author a commentator using a different framework or from scratch.

---

[4]Gathering of Animated Lifelike Agents, at the 7th International Conference on Intelligent Virtual Agents – IVA 2007. The GALA entries can be viewed at http://hmi.ewi.utwente.nl/gala/finalists.
[5]Joint prize with the DEIRA commentator, by a team from the University of Twente [12].

---

[6]http://www.charamel.com
[7]http://www.nuance.com

Module		LOC unique to domain				LOC unchanged			
		Horse race		dTank		Horse race		dTank	
Knowledge	templates	466	62.9%	532	65.9%	275	37.1%	275	34.1%
	knowledge rules	549	97.3$	932	98.4%	15	2.7%	15	1.6%
	total	**1015**	**77.8%**	**1464**	**83.5%**	**290**	**22.2%**	**290**	**16.5%**
Language	referring expressions	97	49.5%	324	76.6%	99	50.5%	99	23.4%
	utterance database	216	100%	134	100%	0	0%	0	0%
	templates	1012	100%	536	100%	0	0%	0	0%
	template Java code	0	0%	0	0%	755	100%	755	100%
	total	**1325**	**60.8%**	**994**	**53.8%**	**854**	**39.2%**	**854**	**46.2%**
Affect	causality relations	98	8.2%	32	2.8%	1091	91.8%	1091	97.2%
	goals and desires	128	100%	20	100%	0	0%	0	0%
	observation rules	24	100%	49	100%	0	0%	0	0%
	Java functions	0	0%	0	0%	755	100%	755	100%
	total	**248**	**11.8%**	**101**	**5.2%**	**1846**	**88.2%**	**1846**	**94.8%**

Table 4: Lines of code changed vs unchanged by module (absolute lines and relative percentages)

Module		unique to domain				unchanged			
		Horse race		dTank		Horse race		dTank	
Knowledge	templates	164	67.5%	97	55.1%	79	32.5%	79	44.9%
	knowledge rules	54	96.4%	108	98.2%	2	3.6%	2	1.8%
	total	**218**	**72.9%**	**205**	**71.7%**	**81**	**27.1%**	**81**	**28.3%**
Language	referring expressions	7	33.3%	30	68.2%	14	66.7%	14	31.8%
	utterance database	6	66.7%	7	70.0%	3	33.3%	3	30.0%
	templates	81	100%	37	100%	0	0%	0	0%
	total	**94**	**84.7%**	**74**	**81.3%**	**17**	**15.3%**	**17**	**18.7%**
Affect	causality relations	1	0.9%	3	2.6%	114	99.1%	114	97.4%
	goals and desires	2	100%	2	100%	0	0%	0	0%
	observation rules	19	100%	8	100%	0	0%	0	0%
	total	**22**	**16.2%**	**13**	**10.2%**	**114**	**83.8%**	**114**	**89.8%**

Table 5: Jess declarations changed vs unchanged by module (absolute declaration count and relative percentages)

Good afternoon, and welcome to our coverage! I'm broadcasting live from the IVA Gala Stakes. We have four entries today: Carmine, Topaz, Eben and Azure. Today's prize money is sponsored by the DFKI; and they will race one lap of a 2000-metre track. We have beautiful spring weather for the race today, and the track is exceptionally hard this week.

The race has begun! Azure is out in front. Eben breaks free from the pack! Azure in the lead. Eben really increasing the gap to Carmine.

There's just over 1700m to go. Azure in front. They're approaching the first post. Eben is about to overtake Azure. They're entering the first turn. Eben in the lead: I hope this goes well!

There's just over 1400m to go. Carmine getting through on the outside of Azure as the field leaves the first turn. Eben in the lead. Carmine is increasing her lead over Azure. Eben in the lead, and Carmine is breathing down the neck of Eben.

Azure finds another ten percent. Carmine passes Eben into the lead. You go, girl! Azure is bearing down on Eben.

They're approaching the second turn. The wonderful Carmine in front. They're entering the second turn. Topaz is about to overtake Eben... and Topaz beats Eben. And Eben won't let Azure through! Carmine in front.

Topaz is breathing down the neck of Carmine as the field leaves the second turn. In the home straight now, Topaz in the lead. Topaz is increasing her lead over Carmine. They're approaching the finish line–

That's the finish! Topaz takes out the IVA Gala, with Carmine in second, and Eben in third. A blanket finish! And lots of excitement in today's race, an excellent showing from Carmine. That's all we have time for ... until next time, farewell!

Figure 4: Sample transcript of ERIC's generated commentary in the horse race domain

software (Tom voice). Special thanks to Patrick Gebhard for help and modifications on his ALMA[8] software.

8. REFERENCES

[1] E. André, G. Herzog, and T. Rist. Generating multimedia presentations for robocup soccer games. In *RoboCup-97: Robot Soccer World Cup I*, pages 200–215, London, UK, 1998. Springer-Verlag.

[2] J. Cassell, T. Bickmore, L. Campbell, and H. Vilhjálmsson. Human Conversation as a System Framework: Designing Embodied Conversational Agents. In J. Cassell, J. Sullivan, S. Prevost, and E. Churchill, editors, *Embodied Conversational Agents*, pages 29–63. MIT Press, Cambridge, MA, 2000.

[3] P. Gebhard. ALMA – A Layered Model of Affect. In *Proceedings of the Fourth International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS'05)*, pages 29–36, Utrecht, 2005.

[4] P. Gebhard, M. Klesen, and T. Rist. Coloring multi-character conversations through the expression of emotions. In *roceedings of the Tutorial and Research Workshop on Affective Dialogue Systems (ADS'04)*, pages 128–141, 2004.

[5] L. R. Goldberg. An alternative "description of personality": The Big-Five factor structure. *Journal of Personality and Social Psychology*, 59(6):1216–1229, 1990.

[6] B. J. Grosz, A. K. Joshi, and S. Weinstein. Centering: A framework for modeling the local coherence of discourse. *Computational Linguistics*, 21(2):203 –225, 1995.

[7] G. Herzog and P. Wazinski. VIsual TRAnslator: Linking perceptions and natural language descriptions. *Artificial Intelligence Review*, 8(2–3):175–187, 1994.

[8] E. F. Hill. *Jess in Action: Java Rule-Based Systems*. Manning Publications Co., Greenwich, CT, USA, 2003.

[9] M. Kipp, K. H. Kipp, A. Ndiaye, and P. Gebhard. Evaluating the Tangible Interface and Virtual Characters in the Interactive COHIBIT Exhibit. In *Proceedings of the 6th International Conference on Intelligent Virtual Agents (IVA 2006)*. Springer, 2006.

[10] M. Kipp, M. Neff, K. H. Kipp, and I. Albrecht. Toward Natural Gesture Synthesis: Evaluating gesture units in a data-driven approach. In *Proceedings of the 7th International Conference on Intelligent Virtual Agents (IVA-07)*, pages 15–28. Springer, 2007.

[11] M. Klesen, M. Kipp, P. Gebhard, and T. Rist. Staging exhibitions: Methods and tools for modeling narrative structure to produce interactive performances with virtual actors. *Virtual Reality. Special Issue on Storytelling in Virtual Environments*, 7(1):17–29, 2003.

[12] F. L. Knoppel, A. S. Tigelaar, D. O. Bos, T. Alofs, and Z. Ruttkay. Trackside DEIRA: A Dynamic Engaging Intelligent Reporter Agent. In *AAMAS07: The Seventh International Conference on Autonomous Agents and Multiagent Systems*, 2008 (submitted).

[13] J.-C. Martin. On the use of the multimodal clues in human behaviour for the modelling of agent co-operative behaviour. *Connection Science, Special Issue on "Agent Autonomy in Groups"*, 14(4):297–309, 2002.

[14] M. Maybury, O. Stock, and W. Wahlster. Intelligent Interactive Entertainment Grand Challenges. *IEEE Intelligent Systems*, 21(5):14–18, 2006.

[15] H. McBreen and M. Jack. Empirical evaluation of animated agents in a multi-modal retail application. In *Proceedings AAAI Fall Symposium: Socially Intelligent Agents*, pages 122–126, 2000.

[16] A. Mehrabian. Pleasure-arousal-dominance: A general framework for describing and measuring individual differences in temperament. *Current Psychology*, 14(4):261–292, December 1996.

[17] G. P. Morgan, F. E. Ritter, W. E. Stevenson, and I. N. Schenck. dTank: An environment for architectural comparisons of competitive agents. In L. Allender and T. Kelley, editors, *Proceedings of the 14th Conference on Behavior Representation in Modeling and Simulation*, 05-BRIMS-043, pages 133–140, Orlando, FL, 2005. University of Central Florida, University of Central Florida.

[18] A. Ndiaye, P. Gebhard, M. Kipp, M. Klesen, M. Schneider, and W. Wahlster. Ambient intelligence in edutainment: Tangible interaction with life-like exhibit guides. In *Proceedings of the first Confernce on INtelligent TEchnologies for interactive enterTAINment (INTETAIN)*, pages 104–113, Berlin, Heidelberg, 2005. Springer.

[19] M. Neff, M. Kipp, I. Albrecht, and H.-P. Seidel. Gesture Modeling and Animation Based on a Probabilistic Recreation of Speaker Style. *Transactions on Graphics*, 2008. to appear.

[20] T. Noma, L. Zhao, and N. Badler. Design of a virtual human presenter. *IEEE Journal of Computer Graphics and Applications*, 20(4):79–85, July/August 2000.

[21] A. Ortony, A. Collins, and G. L. Clore. *The Cognitive Structure of Emotions*. Cambridge University Press, Cambridge, UK, 1988.

[22] R. W. Picard. *Affective Computing*. MIT Press, Cambridge, MA, USA, 1997.

[23] J. Rickel and W. L. Johnson. Animated agents for procedural training in virtual reality: Perception, cognition, and motor control. *Applied Artificial Intelligence*, 13:343–382, 1999.

[24] T. Rist, E. André, S. Baldes, P. Gebhard, M. Klesen, M. Kipp, P. Rist, and M. Schmitt. A review of the development of embodied presentation agents and their application fields. In H. Prendinger and M. Ishizuka, editors, *Life-Like Characters – Tools, Affective Functions, and Applications*, pages 377–404. Springer, Heidelberg, 2003.

[25] C. Rocchi, O. Stock, M. Zancanaro, M. Kruppa, and A. Krüger. The Museum Visit: Generating Seamless Personalized Presentations on Multiple Devices. In *Proceedings of International Conference on Intelligent User Interfaces IUI2004*, 2004.

[26] M. Strauss. Realtime generation of multimodal affective sports commentary for embodied agents. Master's thesis, Saarland University, 2007.

[27] M. Theune, E. Klabbers, J. Odijk, J.-R. de Pijper, and E. Krahmer. From data to speech: A general approach. *Natural Language Engineering*, 7(1):47–86, 2001.

[28] K. van Deemter, E. Krahmer, and M. Theune. Real vs. template-based natural language generation: a false opposition? *Computational Linguistics*, 31(1):15–23, 2005.

[29] H. van Vliet. *Software engineering (2nd ed.): principles and practice*. John Wiley & Sons, Inc., New York, NY, USA, 2000.

[30] V. Vinayagamoorthy, M. Gillies, A. Steed, E. Tanguy, X. Pan, C. Loscos, and M. Slater. Building Expression into Virtual Characters. In *Eurographics Conference State of the Art Report*, Vienna, 2006.

---

[8] http://www.dfki.de/~gebhard/alma.html