A Methodology to Design Human-Like Embodied Conversational Agents

Zacharie Alès, Guillaume Dubuisson Duplessis, Ovidiu Şerban, Alexandre Pauchet

A Methodology to Design Human-Like Embodied Conversational Agents based on Dialogue Analysis

Z. Alès, G. Dubuisson Duplessis, O. Şerban, and A. Pauchet

LITIS - EA 4108, INSA Rouen, Avenue de l'Université - BP 8,
76801 Saint-Étienne-du-Rouvray Cedex, France
contact: pauchet@insa-rouen.fr

Abstract. This paper presents a bottom-up methodology to study human-human and human-agent dialogues in order to improve the design of embodied conversational agents (ECAs). The methodology uses a matrix representation of dialogues, constructed by means of a (semi-)automatic annotation process. A multidimensional dialogue pattern extraction and a clustering algorithm are applied to the coding matrices. The extracted patterns model the human behaviours.

Keywords: Embodied conversational agents, affective and multimodal dialogue, automatic annotation, pattern extraction.

1 Introduction and Objectives

Heterogeneous multiagent systems and mixed communities, composed of collaborative humans and software agents, become commonplace. Human-agent interactions have to be rich and efficient to ease the comprehension. As humans already use robust communication protocols and reasoning processes, the behaviour model of artificial agents needs to suit human standards. For example, a dialogical assistant system has to be efficient concerning the task to perform but should also integrate a natural communicative behaviour.

Thus, Intelligent Virtual Agents (IVAs) and Embodied Conversationnal Agents (ECAs) [14] are two particular types of agents which aim at offering human-like interaction. However, the design of an ECA is difficult due to the involved algorithms: multimodal input recognition (i.e. utterances, gestures, gazes, vocal inflections), natural language understanding and generation, dialogue management, planning and cognitive capacities, emotion modelling, prosodic speech generation, non-verbal behaviour, etc. In particular, multimodal and affective dialogue management remains inefficient in ECAs, even though this aspect is essential for effective interaction and collaboration [54].

The design of ECAs can be improved by analysing and modelling human-human and human-agent interactions. Among the existing interaction and dialogue models, the most common approach adopted uses regular structures (for

instance automata [58], timed automata [45], sequence diagrams [21], etc.), manually extracted or computationally learned from a corpus of dialogues, traces or logs. All these data structures can only represent linear interaction patterns, whereas dialogue management involves multi-dimensional levels [10]. The interaction model of an ECA needs to take into account all the aspects involved in human interaction (individual and collective task management, dialogue feedbacks, affective aspects, social commitments, etc.), expressed according to various modalities (semantics, prosody, gestures, facial expressions, etc.).

Firstly, a bottom-up methodology is proposed to build an interaction model for an ECA dedicated to a given task. This methodology needs a corpus of dialogues as input. These dialogues are using matrix representation for labels, ideally using automatic annotation techniques. Secondly, a multidimensional pattern extraction and a clustering algorithm are applied to the coding matrices, to collect the interaction regularities. Lastly, the interaction modelling phase consists in using the interaction patterns arising from the human behaviours.

This study is divided into five different sections. Section 2 draws some links with related work, with particular attention on ECAs, affective computing and dialogue systems. Section 3 describes more precisely the approach. Section 4 focuses on an example of automatic annotation, i.e. emotion detection. Section 5 concerns automatic extraction of multi-dimensional dialogue patterns. Section 6 links dialogue patterns and human-agent dialogue management. Lastly, section 7 concludes this article.

2 Embodied Conversational Agents, Affective Computing and Dialogue Systems

2.1 Embodied Conversational Agents

Recent research projects have a particular interest in increasing the interactiveness of ECAs, adding expressiveness and special graphic features to improve the general agent quality [46, 14]. Since recent studies have proven that ECA increase the interaction time [14], a special category of non-animated agents appear in many of the client-support applications (e.g.: Laura from EDF -Electricity of France Company-, Anna from IKEA). Although these examples are scientifically irrelevant for most of the ECA research community (their level of interactivity tends to zero), they demonstrate the increasing interest for ECAs. André and Pelachaud categorized ECAs according to their capabilities and expressiveness: TV-style presenter systems, team presentations, one-to-one interaction (one ECA vs one human) and multi-party conversations (multiple ECAs and users) [5]. This work is mainly linked with one-to-one interaction.

REA [14] is an agent designed for natural conversations, whose embodiment and human-like appearance help to maintain the communication channel between agents and humans. SmartKom [56] is a multimodal dialogue system integrated into an information kiosk scenario, that supports complex input-output modalities. SmartKom considers all the classic modalities (i.e. speech), nonverbal (i.e. gestures) and touch actions, through interactive boards. Max [33]

uses synthetic speech activities, gestures, facial expressions and gaze to communicate. The MARC (Marc Affective Reactive Characters) platform [18] considers the affective modalities (detection and expression) as the key aspect. Greta [46] would be currently the most advanced ECA, considering expressiveness, features and formats. Greta supports Behaviour Markup Language (BML) [34], a language that has been created to ease the agent behaviour programming and produce complex gestures, postures and gazes. Greta is also one of the few ECAs that support FML-APML, which is an implementation of the Functional Markup Language [28] through the APML markup language for behaviour specification [19]. FML-APML aims to regroup all the behaviours of an agent into functions, rather than describing individual localized behaviour. Finally, the European project SEMAINE [49] should be reminded as a promoter of the Sensitive Artificial Listeners field. SEMAINE offers a fully functional platform for further development, that includes basic emotion detection, affective speech synthesis, a basic turn-taking dialogue model and a functional ECA, based on Greta.

As pointed out by Swartout *et al.*, dialogue management on a multimodal and affective level remains inefficient in existing ECAs [54]. We aim at designing richer interaction models for the existing platforms, i.e. models dealing with multidimentional dialogue management and multimodal inputs and outputs, in order to increase the interaction time between humans and ECAs.

2.2 Affective Computing and Agents

Affective models are integrated into ECAs to adapt their behaviours to the emotional status of the user (emotion detection) and to improve their expressiveness (facial expressions, vocal inflections, etc.). Affective phenomena are part of Affective Computing (AC) research field and include emotions, feelings, moods, etc. Emotions are complex and fuzzy but universal and therefore essential to model. The affects can be considered from six perspectives. Four perspectives (expressions, embodiments, cognitive appraisal, and social constructs) are derived from traditional emotion theories. They could be extended with theoretical contributions from the affective neuroscience (fifth perspective). The sixth perspective concerns the measurements of physiological signals describing emotions [47]. The AC applications are widely spreading from user interfaces to the well-known lie detector [11]. In the user interface area, some ECAs are also able to detect and produce emotion [13, 41]. While the production part is more advanced [14], the detection is far from being accurate for some classes of emotions [47], leading to a complete new area of research like opinion mining and affective valence [57].

This article focus on emotion detection on a semantic level, as an example of the automatic annotation process. The research on detecting emotional content in texts refers to written language and transcriptions of oral communication. Since different languages use similar emotional concepts, a number of researchers have built corpus of emotional terms. For instance, Wordnet is a lexical database of English terms used in computational linguistic research [44]. Strapparava and Valitutti [53] extended Wordnet with information on affective terms. SentiWordNet [8] was also built to serve as sentiment analysis support and opinion mining.

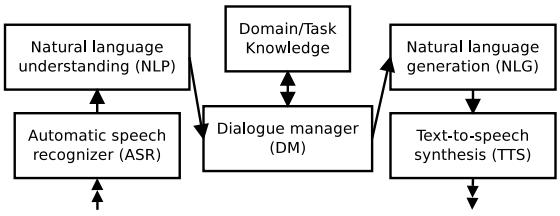

Fig. 1. Basic architecture of a spoken dialogue system.

The most complex approach to textual affect sensing involves systems that construct affective models from large corpora of world knowledge and apply these models to identify the affective tone in texts [9, 1].

2.3 Dialogue Systems

The complexity of a dialogue system mainly depends on its application. However, number of components are recurrent. Figure 1 presents the basic architecture of a spoken dialogue system (similar diagrams can be found in the literature, e.g., [30]). The Automatic Speech Recognition (ASR) component acquires the user's input as speech and converts it into a sequence of words. Next, the Natural Language Understanding (NLU) component generates an adequate representation of the user's input for the Dialogue Manager (DM). This latter integrates the user's contribution into a dialogue context and determines the next system action, possibly by using domain or task knowledge. Then, the Natural Language Generation (NLG) component generates words to express. Finally, the spoken form of the response is produced by the text-to-speech (TTS) component.

Several approaches exist but no approach clearly dominates. The simplest one is the *finite-state* approach (for instance see [43]) that represents the structure of the dialogue as a finite-state automaton where each utterance leads to a new state. This approach describes the structure of the dialogue but do not explain it. In practice, this approach is limited to system-directed dialogues.

The *frame-based* approach represents the dialogue as a process of filling in a frame (also called *form*) which contains a series of slots (for instance, see [6]). Slots usually correspond to information that the system needs to acquire from the user. It is less rigid than the finite-state approach. Indeed, the dialogue manager includes a control algorithm which determines the response of the system. For instance, the user can fill several slots in one utterance unlike the finite-state approach. However, the possible contributions of the system are fixed in advance.

The *plan-based* approach [4] comes from classic AI. It combines planning techniques such as plan recognition with ideas from the speech act theory [7, 50]. An example of implementation is TRAINS [3]. This approach is rather complex from a computational perspective. In particular, it requires advanced NLU components in order to infer the speaker's intentions.

The Information State Update (ISU) framework [37], proposed by the TRINDI project, implements different kinds of dialogue management models. The

central component of this approach is called the Information State (IS). It is a formal representation of the common ground between the dialogue participants as well as a structure to deal with agent reasoning. Dialogue act triggers update the IS. GoDIS is an example of system based on this approach [36].

The *logic-based* approach represents the dialogue and its context in some logical formalism and takes advantage of mechanisms such as inference (for instance, see [29, 42] which also present *dialogue games*). Most of the logic based approach works are only on a theoretical level.

More recent approaches aim to learn dialogue policies with machine learning techniques such as reinforcement learning [23]. In this approach, the dialogue management is seen as a decision problem and the dialogue system is modelled as a Markov Decision Process. These approaches require an extensive effort of annotation since a large amount of annotated data is necessary.

As already pointed out, dialogue management remains a major deadlock in ECAs [54]. Most of the existing ECAs only integrates basic dialogue management processes, such as a keyword spotter within a finite-state approach or a frame-based approach (for instance, see the SEMAINE project [49]). It is mainly due to the complexity of all the components that compose a dialogue system, the addition of fuzziness along the processing flow and the multidimentionality and multimodality of dialogues. As Hulstijn [29], we are convinced that the dialogue can be managed on a deliberative way considering the task resolution and on a reactive way when dealing with dialogical conventions. Particularly, interaction patterns represented with dialogue games should model dialogical conventions.

3 A New Methodology to Design ECAs

The interaction capabilities of ECAs can be improved studying and analysing of human-human and human-agent interaction. In this study, the proposed methodology is a generalization and an automation of the manual classic approach used to study dialogue corpora in various research fields, such as linguistic psychology. Analysing a corpus of complex dialogues, e.g. multimodal and/or affective dialogues, consists in the various steps shown on figure 2.

1. Firstly, *collecting and digitizing* a dialogue corpus, using an audio format, or a video format enable to encode multimodal information. Usually, this step is carried out during a user experiment designed on purpose (human-human dialogues, wizard of Oz, etc.). Since our methodology is bottom-up, the experiments performed to collect a corpus are task dependant.
2. Secondly, the *transcription and coding* step produce raw dialogues with various levels of details (speaking slots, utterances, onomatopoeia, pauses, etc.) depending on the phenomena and characteristics that the resulting agent must exhibit (i.e. depending on the final application).
3. Next, a *knowledge extraction* step is applied in accordance with criteria defined using a predefined coding scheme (e.g.: illocutionary force applied to a proposition following the speech act theory, facial expression, gestures, emotions, etc.).

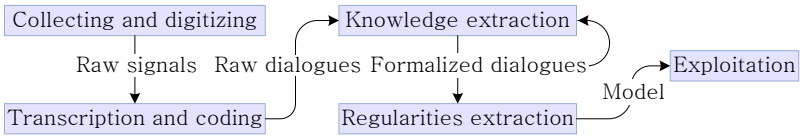

Fig. 2. Workflow of this methodology

Speaker	**Utterance**	**Annotations**
T	Good morning, what can I do for you ?	E G V
C	One ticket to Valence	D G M
C	Quickly, please !	D P
T	I check...	E A
T	The next one is in 10 minutes	A A
T	Would that suits you ?	E V

Table 1. 3D annotations of a dialogue between a ticket clerk and a customer.

4. Then, a *regularity extraction* is applied to the formalized traces. Regularities represented in a computable format constitute the model as a pattern database.
5. Finally, the model is *exploited*, depending on the application field (i.e. the design of an ECA dedicated to a particular task in our case). Exploitation requires a pattern recognition algorithm with a pattern-based model.

Step 2 can be solved using existing tools such as, for the transcription step, Sphinx software or a non-free solution for instance. This article focus on proposing automatic tools to cover steps 3 and 4.

As outlined by Bunt, dialogue management involves multilevel aspects [10]. In order to design an ECA that supports multidimensionality, like humans, a matrix representation is chosen for annotations. Each utterance is characterised by an annotation vector, which components match the different coding dimensions. Consequently, a dialogue is represented by a matrix: one row by utterance, one column by coding dimension.

To illustrate this two-dimensional representation, table 1 presents a hypothetical dialogue between a ticket clerk and a customer in a train station. Each utterance is characterized by a speaker (T: ticket clerk, C: customer), a transcription and its encoding annotations along three columns:

1. The first column characterizes the speech act type of the utterance (E: engaging, D: directive, A: assertive) [50].
2. The second column encodes social aspects (G: greetings, P: politeness).
3. The third column describes gazes (M: mutual eye contact, V: unilateral gaze of the speaker, A: gaze to an other identified element).

With this matrix representation, the knowledge extraction step consists in manual, semi-automatic and/or automatic annotation processes - one for each

dimension/column. Once the dialogues are coded, the regularity extraction step is applied onto the coding matrix.

4 Automatic Annotation of Emotion

Annotation is the process of associating a set of finite labels to a piece of data, which would represent a uniform structure. In the whole process of the presented methodology, the automatic annotation step enables to formalize a raw dialogue into a matrix representation. Examples of fully annotation schemes, manually performed, can be found in [15] regarding multimodal interactions, [39] concerning dialogue acts or [51] for affective annotation.

Many annotation techniques exist, based on data type (i.e. audio-video data, text or interaction logs), experimental objective (i.e. emotion detection, dialogue acts, knowledge representation) and degree of automation (i.e. fully automatic, semi-automatic or manual). In this section, as example, we focus on a particular type of automatic annotation: emotion detection in textual utterances. To obtain a completely formalized dialogue, other annotation algorithms should be applied, one for each dimension in the coding matrix.

4.1 Emotional learning corpus

An algorithm dedicated to emotion detection requires a learning database. The chosen corpus is the one from SemEval 2007 conference, task 14 [52]. The data set contains headlines (newspaper titles) from major websites, such as New York Times, CNN, BBC or the search engine Google News. The corpus characteristics suits our problem requirements: textual sentences the length of which is similar to that of usual cuttings of utterances. Moreover, the results could easily be compared to other systems that participated to the SemEval task, which also consists in emotion detection.

The corpus was manually annotated (as it was provided by the authors of the SemEval challenge) by 6 different persons using the Ekman's basic set of emotions: Anger, Disgust, Fear, Joy (Happiness), Sadness, Surprise [20]. The annotators were instructed to tag the headlines according to the presence of affective words or group of words with emotional content. In situations where the emotion was uncertain, they were asked to follow their first feeling. Headlines are annotated with a 0 to 100 scale for each emotion. A valence annotation was also carried out. Valence represents the intrinsic attractiveness (positive valence) or aversiveness (negative valence) of an event, object, or situation. In SemEval task, the valence is used to describe the intensity of a positive or negative emotion. The valence label ranged from -100 to 100.

Table 2 presents examples of headlines from the corpus, among with their significant emotions. In this example only the significant emotions were chosen, by picking up the labels with a value in the neighbourhood of the dominant emotion (all the values between 20% range).

A	D	F	J	Sad.	Sur.	Headline
-	-	-	0.15	0.25	-	Bad reasons to be good
-	-	-	-	-	0.36	Martian Life Could Have Evaded Detection by Viking Landers
-	-	0.68	-	-	-	Hurricane Paul nears Category 3 status
-	-	-	0.75	-	0.57	Three found alive from missing Russian ship - report
0.52	0.64	0.50	-	0.43	-	Police warn of child exploitation online

Anger=**A**, Disgust=**D**, Fear=**F**, Joy=**J**, Sadness=**Sad.**, Surprise=**Sur.**

Table 2. Headlines from the training corpus, presented with dominant emotions

4.2 Classification model

The classifier we have chosen is a classic unsupervised method, Self-Organizing Maps (SOM) [32]. This method is a particular type of neural network used for mapping large dimensional spaces into small dimensional ones. The SOM has been chosen because: 1) it usually offers good results with fuzzy data, 2) the training process is easier than other neural networks and 3) the classification speed is sufficiently high to fit the requirements. This technique requires a 3-step process, each step assuring the output for the next step.

1. **Preprocessing:** filtering and cleaning the text information.
2. **Feature extraction:** a projection is made using a Latent Semantic Analysis (LSA). Hence, all the occurrences of key terms are counted and introduced into a matrix (a row for each keyword, a column by headline). Two different strategies are tested concerning the projection set of words. We use 1) *LSA Training*: the words of the training set and 2) *LSA Gutenberg*: the top 10 000 most frequent English words, extracted from approximately 1 000 000 documents existing in the Project Gutenberg[1].
3. **Classification:** the SOM algorithm is applied and the trained model is used in the classification step.

While the first two steps are used both for training and testing the corpus, the SOM algorithm is applied only during the training phase. More details about the algorithm can be found in [51].

4.3 Results and Discussion

Table 3 presents our results as well as the most significant scores obtained by the systems participating in the SemEval 2007, task 14 [52]. The LSA All emotion system [52], is a meta-LSA method applied on the corpus, using as support sets of words the existing in the WordNet Affect database and all direct synonyms,

[1] Project Gutenberg is a large collection of e-books, processed and reviewed by the project's community. All the documents are freely available at the website: `http://www.gutenberg.org/wiki/Main_Page`

Method	Precision	Recall	F1
LSA Training	*20.50*	*19.57*	*20.02*
LSA Gutenberg	*24.22*	*23.31*	***23.76***
LSA All emotion [52]	9.77	**90.22**	17.63
UA [35]	17.94	11.26	13.84
UPAR7 [16]	**27.60**	5.68	9.42

Table 3. Overall results. Our two methods are presented in italics; the other methods are the best methods from SemEval competition. Bold caracters highlight the best result for each evaluation measure.

linked by the synset relation. UA [35] uses statistics gathered from three search engines (MyWay, AlltheWeb and Yahoo) to determine the amount of emotion in each headline. Emotional score is obtained with the Pointwise Mutual Information (PMI) algorithm [35]. UPAR7 [16] is a rule-based system with a linguistic approach. The system uses the Stanford syntactic parser on the titles and identifies information about the main subject by exploiting the dependency graph obtained by the parser.

The LSA All emotions offers a good coverage over the emotional words, but its synonym expansion algorithm introduces a lot of noise in the method, and therefore offers a very poor precision. UPAR7 leads to a good precision, thanks to its analytical nature, but it lacks in recall. Our LSA Gutenberg system is a good compromise between precision and recall, as F1 measure shows.

The choice of an annotation method is highly influenced by the degree of automation and the algorithmic speed required. A system dedicated to affect discovery in a semi-automatic way, would probably prefer a method with a high recall (precision is not required, but desired since a lot of false positives could influence the filtering process). On the contrary, an automatic system able to recognize emotions in real-time would require a good balance between precision, recall and time, as our LSA Gutenberg proposes.

This emotion detection technique enables to annotate the utterances of a dialogue along a single dimension. As we work with labels in a matrix form, other annotation algorithms, very similar to the one presented here, could be applied: speech act tagging for utterances, gesture detection on a multimodal level, etc. This would add multiple dimensions to our annotation system.

5 Extraction of Dialogue Patterns

The extraction of regularities is a crucial step in our approach since the model is directly defined through them. In this context, we present a dynamic programming based method designed to extract two-dimensional patterns from the annotions of the dialogues. We thereafter describe how clustering heuristics can be used to highlight the most recurrent patterns and therefore define the model.

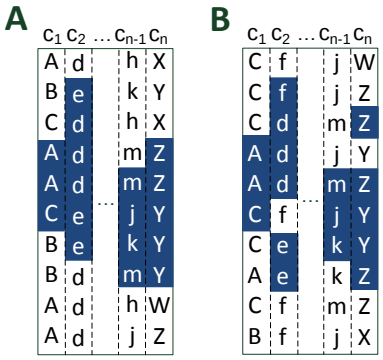

Fig. 3. Example of similar (but not identical) patterns in the annotations of two different dialogues, **(A)** and **(B)**. The encoding matrices are composed of n columns. Each column has its own alphabet. The colored parts represent the two versions of the pattern which differ by an insertion/deletion, 2 gaps and 2 subtitutions.

5.1 Dialogue Patterns in Annotations

With our matrix representation, a *dialogue pattern* is defined as a set of annotations whose arrangement occurs - exactly or approximately - in several dialogues. No structural constraint is imposed: a pattern can contain non-adjacent annotations in rows or columns (i.e. it may contain gaps), and two versions of a pattern can contain insertions, deletions and subtitutions (see figure 3).

5.2 Two-dimensional Pattern Extraction

The method we have chosen to extract two-dimensional patterns is a generalization of the local string edition distance which can be assimilated to sequence alignment. The edit distance *ed* (or Levenshtein distance) between two string s_1 and s_2 corresponds to the minimal cost of the three elementary edit operations for converting s_1 to s_2. These three operations are insertion and deletion of a character, and substitution of a character by another.

The two-dimensional pattern extraction problem corresponds to matrix alignment. A local alignment of two texts S_1 and S_2, of size $m_1 \times n_1$ and $m_2 \times n_2$ respectively, consists in finding the portion of S_1 and the portion of S_2 which are the most similar (among all the pairs of portions of S_1 and S_2). To this end, a four-dimensional matrix T of size $m_1 \times n_1 \times m_2 \times n_2$ is computed, such that $T[i][j][k][l]$ is equal to the local edition distance between $S_1[0..i-1][0..j-1]$ and $S_2[0..k-1][0..l-1]$ for all $i \in [\![1, m_1 - 1]\!]$, $j \in [\![1, n_1 - 1]\!]$, $k \in [\![1, m_2 - 1]\!]$ and $l \in [\![1, n_2 - 1]\!]$. In our heuristic, the calculation of T is obtained by the minimisation of a recurrence formula. Once T is computed, the best local alignment is found from the position of the maximal value in T, using a traceback algorithm to infer the characters which are part of the alignment. Further details can be found in [15, 2].

Let $c^* \in \mathbb{Z}^+$ be the cost of the best alignment for a given matrix T and let $p^* \in \mathbb{Z}^{4+}$ be its position in T. Since a list of the most similar alignments is needed (and not only the best alignment), several tracebacks are performed: one from p^* and one from each position in T whose cost c is such that $c \geq c^* - \varepsilon$ with $\varepsilon \in \mathbb{Z}^+$. The higher the value of ε, the more alignments are returned. Each alignment contains two patterns, one in S_1 and one in S_2.

Method		Type	K fixed	Rep.
Single-Link	[22]	Hierarchical	×	×
ROCK	[27]	Hierarchical	×	×
CHAMELEON	[31]	Hierarchical	×	×
Unormalized spectral clustering	[55]	Partitional	✓	×
Shi and Malik spectral clustering	[55]	Partitional	✓	×
Jordan and Weiss spectral clustering	[55]	Partitional	✓	×
Affinity propagation	[24]	Partition	×	✓

Table 4. Features of the implemented clustering heuristics. **Type:** Partitional algorithms return a partition of the data whereas hierarchical methods return a sequence of nested partitions. **K fixed:** ✓if the number of clusters is an input of the method, × otherwise. **Rep.:** ✓if the method gives a representative for each cluster.

5.3 Pattern Clustering

The patterns, extracted using the matrix alignment algorithm, only appear - exactly or approximately - in two dialogues. In order to determine the importance of each pattern, we propose to group them by means of various standard clustering heuristics (see table 4). The underlying idea is that large clusters of patterns represent behaviors commonly adopted by humans whereas patterns from small clusters tend to be marginal. A matrix of similarities between patterns is computed using a global edition distance applied on all pairs of selected patterns. This similarity matrix is used as input for the clustering heuristics.

5.4 Results and Discussion

The method has been tested on a corpus of 70 manually annotated dialogues. The coding space is composed of five columns which alphabets include between three and six symbols. The average size of dialogues is fifty utterances.

During the extraction phase, 1740 dialogue patterns have been collected to produce the similarity matrix used by the various clustering heuristics. Ideally, the evaluation of the results should be carried out manually by an expert. However, as the number of solutions is to large to compare them one by one, the Dunn's index is used to assess the various methods.

Let s_{ij} be the similarity between patterns i and j, and $c(i)$ the number of the cluster which contains i for a given solution. Dunn's index is equal to

$$\frac{\min\limits_{c(i)\neq c(j)} s_{ij}}{\max\limits_{c(k)=c(l)} s_{kl}}$$

Thus, solutions with a large value of Dunn's index tend to be relevant because they are composed of compact and well separated clusters.

As the number of clusters to be found is part of the problem, the various methods are tested for all the possible values. Table 5 presents some scores

	Number of clusters					
	5	20	50	80	116	150
Single-Link	41	97	183	270	320	360
CHAMELEON	458	605	**628**	-	-	-
ROCK	520	600	621	626	629	630
Unormalized spectral clustering	277	**658**	563	155	194	226
Shi and Malik spectral clustering	524	615	**628**	**631**	**631**	**632**
Jordan and Weiss spectral clustering	**555**	616	**628**	**630**	**631**	**632**
Affinity propagation	-	-	-	-	**632**	-

Table 5. Dunn's index results according to the number of clusters for implemented heuristics. '-' is used whenever no solution is produced for a given number of clusters. Bold caracters highlight the best result(s) for each number of clusters.

of Dunn's index which are quite representative of the overall results. The best methods for the presented problem seem indeed to be the affinity propagation and the spectral clustering methods.

6 Dialogue Patterns and Dialogue Management

One striking observation in a human-human corpus is the presence of *interaction patterns* [12, 45, 38]. We called *interaction pattern* or *dialogue pattern* an ordered set of utterances that is naturally and frequently reoccurring during dialogues. An exchange of greetings and question-answer pairs are two examples of interaction patterns. These patterns can be analyzed along a single dimension (e.g. the performative axis), whereas they participate to other dimensions such as social commitments, affective states, dialogue control functions, etc. [10]. Moreover, each dialogue act is usually expressed through various modalities. Our purpose is to extract semi-automatically these dialogue patterns from corpora (cf. section 5). The dimensions concerned by the pattern extraction process depends on the annotion step. Section 4 presented an example of automatic annotation algorithm (emotion tagging) that has to be completed (with, e.g., performative tagging, facial expression detection, etc.) in order to obtain a multidimentional representation of the dialogues.

From the dialogue system community perspective, two approaches are opposed [29, 42]: the *plan-based* one and the *conventional* one. The first approach aims at modelling the *task structure* (also called *intentional structure* [26]). Basically, this approach considers that an utterance conveys an intention that plays a role to accomplish the goal that is motivating the dialogue. An example of this approach can be found in [4]. The second approach can be called the *conventional approach*. It aims at studying the *interaction patterns* to produce rules that describe admissible sequences of utterance types, with no particular focus on the underlying intentions. Many types of utterances are not consciously emitted but are conventionally triggered by the context. These reoccurring patterns

can be studied either in terms of *dialogue grammars* [48] or *dialogue games* [38, 40].

Some researchers strongly argue that these two approaches are actually *complementary* instead of being opposed [29, 42]. Their point is based on the theory of planning and action [25] that explicitly considers *joint activities* that progresses through *joint actions* carried out by the participants. The key characteristic of a joint action is the *coordination* of *participatory actions* by two or more people. From this point of view, carrying a piano, playing a duet and paddling a canoe together are examples of joint actions. Coordination can be achieved by *conventions*, by a precedent, by explicit agreement or by a communication of some sort. It has been strongly argued that communication processes can be considered as joint actions between a speaker and hearers [17]. From this point of view, dialogue, as a shared and dynamic activity, requires both high-level deliberative reasoning and low-level reactive responses. As a result, Hulstijn [29] proposes to go towards a hybrid reactive/deliberative architecture where "a theory of joint actions may serve as a motivation or 'semantics' to the interaction patterns described as dialogue games". In the same way, Maudet [42] proposes to use dialogue games to model certain aspects of dialogical conventions.

In the presented methodology, *interaction patterns* are extracted and clustered from a given corpus. The set of obtained clusters are subsequently used to create a pattern database. We can benefit from the pattern database in two different ways: 1) as a support to create *dialogues games* and 2) new dialogues can also be characterized by the patterns from the database it contains. To that end, an algorithm to perform the recognition of a pattern in a dialogue is necessary. Such a method applied on two-dimensional annotated dialogues have been developed during a previous work [2].

7 Conclusion and Future Work

In this article, we have presented a methodology which aims at improving and easing the design of human-machine models of interaction dedicated to ECAs. The proposed methodology is based on a matrix representation of dialogues, obtained throughout an automatic annotation step. This representation enables to encode the multidimensational aspects of dialogues. A regularity extraction algorithm is applied onto the annotation matrices of the dialogues, in order to obtain dialogue patterns. The collected patterns are then clustered to select the more representative ones for the interaction model. We are convinced that an interaction model based on dialogue patterns, such the one we propose, is perfect to manage most of the aspects of dialogical conventions. The annotation, pattern extraction and clustering steps are already validated.

One of the interesting perspectives of this work is the effective integration of extracted patterns into the dialogue model of an ECA and the validation of the whole methodology. As Maudet [42], we propose to use dialogue games to manage the dialogical conventions represented by the dialogue patterns.

As application field, we propose the CNRS INS2I-INSHS PEPS project ACAMODIA. This project aims at designing an affective ECA dedicated to a story-telling task for children, situated into an ambiant intelligence environnement. A corpus of story-telling dialogues between a mother and her child has already been collected (section 5 contains the description of the corpus). A second corpus has also been collected, using a wizard of Oz experiment. The study of these two dialogue corpora will lead to the design of an affective ECA dedicated to the narration of stories to children. The story telling part, dealing with the task level, will be managed with a plan-based approach whereas the interaction part (digression, dialogue management, etc.) corresponds to dialogue games and dialogue pattern approach of the methodology presented in this study.

References

1. Akkaya, C., Wiebe, J., Mihalcea, R.: Subjectivity word sense disambiguation. In: Proc. of EMNLP09. pp. 190–199. Association for Computational Linguistics (2009)
2. Alès, Z., Pauchet, A.: Reconnaissance de motifs dialogiques approchés. In: proceedings of MFI. pp. 9–19. Rouen, France (2011)
3. Allen, J., Ferguson, G., Miller, B., Ringger, E., Sikorski-Zollo, T.: Dialogue systems: From theory to practice in TRAINS-96. In: Handbook of Natural Language Processing. pp. 347–376 (2000)
4. Allen, J., Perrault, C.: Analyzing intention in utterances. Artificial Intelligence 15(3), 143–178 (1980)
5. André, E., Pelachaud, C.: Interacting with embodied conversational agents. Speech technology pp. 123–149 (2010)
6. Aust, H., Oerder, M., Seide, F., Steinbiss, V.: The philips automatic train timetable information system. Speech Communication 17(3-4), 249–262 (1995)
7. Austin, J.: How to Do Things with Words. Oxford University Press, Oxford (1962)
8. Baccianella, S., Esuli, A., Sebastiani, F.: Sentiwordnet 3.0: An enhanced lexical resource for sentiment analysis and opinion mining. In: Seventh conf. on Int. Lang. Res. and Eval., Malta. Retrieved May. vol. 25, p. 2010 (2010)
9. Breck, E., Choi, Y., Cardie, C.: Identifying expressions of opinion in context. In: Proc. of IJCAI07 (2007)
10. Bunt, H.: Multifunctionality in dialogue. Computer Speech and Language 25(2), 222–245 (2011)
11. Calvo, R., D'Mello, S.: Affect detection: An interdisciplinary review of models, methods, and their applications. IEEE Transactions on Affective Computing pp. 18–37 (2010)
12. Carletta, J., Isard, S., Doherty-Sneddon, G., Isard, A., Kowtko, J., Anderson, A.: The reliability of a dialogue structure coding scheme. Computational linguistics 23(1), 13–31 (1997)
13. Cassell, J., Pelachaud, C., Badler, N., Steedman, M., Achorn, B., Becket, T., Douville, B., Prevost, S., Stone, M.: Animated conversation: rule-based generation of facial expression, gesture & spoken intonation for multiple conversational agents. In: Proc. of the 21st annual conference on Computer graphics and interactive techniques. pp. 413–420. ACM (1994)
14. Cassell, J., Thorisson, K.: The power of a nod and a glance: Envelope vs. emotional feedback in animated conversational agents. Applied Artificial Intelligence 13(4), 519–538 (1999)

15. Chanoni, E., Lecroq, T., Pauchet, A.: Extraction de motifs dans des dialogues annotés par programmation dynamique. In: N. Maudet, PY. Schobbens, e.M.G. (ed.) MFI. vol. 1, pp. 101–112 (2009)
16. Chaumartin, F.: Upar7: A knowledge-based system for headline sentiment tagging. In: Proc. of SemEval-2007. pp. 422–425 (2007)
17. Clark, H.: Using language. Cambridge University Press Cambridge (1996)
18. Courgeon, M., Martin, J., Jacquemin, C.: Marc: a multimodal affective and reactive character. In: Proceeding of Workshop on Affective Interaction on Natural Environment (2008)
19. DeCarolis, B., Pelachaud, C., Poggi, I., Steedman, M.: Apml, a mark-up language for believable behavior generation. Life-like Characters. Tools, Affective Functions and Applications pp. 65–85 (2004)
20. Ekman, P., Friesen, W.: The repertoire of nonverbal behavior: Categories, origins, usage, and coding. Semiotica 1(1), 49–98 (1969)
21. FIPA: FIPA communicative act library specification. Tech. rep., Foundation for Intelligent Physical Agents (2002)
22. Florek, K., Lukaszewicz, J., Perkal, J., Steinhaus, H., Zubrzycki, S.: Sur la liaison et la division des points d'un ensemble fini. In: Colloquium Mathematicum. vol. 2, pp. 282–285 (1951)
23. Frampton, M., Lemon, O.: Recent research advances in reinforcement learning in spoken dialogue systems. The Knowledge Engineering Review 24(04), 375–408 (2009)
24. Frey, B., Dueck, D.: Clustering by passing messages between data points. science 315(5814), 972 (2007)
25. Grosz, B., Kraus, S.: Collaborative plans for complex group action. Artificial Intelligence 86(2), 269–357 (1996)
26. Grosz, B., Sidner, C.: Attention, intentions, and the structure of discourse. Computational Linguistics 12(3), 175–204 (1986)
27. Guha, S., Rastogi, R., Shim, K.: Rock: A robust clustering algorithm for categorical attributes* 1. Information Systems 25(5), 345–366 (2000)
28. Heylen, D., Kopp, S., Marsella, S., Pelachaud, C., Vilhjalmsson, H.: Why conversational agents do what they do? functional representations for generating conversational agent behavior. In: The First Functional Markup Language Workshop. Estoril, Portugal (2008)
29. Hulstijn, J.: Dialogue games are recipes for joint action. In: Proc. of Gotalog'00 (2000)
30. Jokinen, K., McTear, M.: Spoken Dialogue Systems, vol. 5. Morgan & Claypool (2010)
31. Karypis, G., Han, E., Kumar, V.: Chameleon: Hierarchical clustering using dynamic modeling. Computer 32(8), 68–75 (1999)
32. Kohonen, T.: The self-organizing map. Proc. of the IEEE 78(9), 1464–1480 (1990)
33. Kopp, S., Jung, B., Lessmann, N., Wachsmuth, I.: Max - a multimodal assistant in virtual reality construction. KI 17(4), 11 (2003)
34. Kopp, S., Krenn, B., Marsella, S., Marshall, A., Pelachaud, C., Pirker, H., Thórisson, K., Vilhjálmsson, H.: Towards a common framework for multimodal generation: The behavior markup language. In: Intelligent Virtual Agents. pp. 205–217. Springer (2006)
35. Kozareva, Z., Navarro, B., Vázquez, S., Montoyo, A.: Ua-zbsa: A headline emotion classification through web information. In: Proc. of SemEval07. pp. 334–337. Association for Computational Linguistics (2007)

36. Larsson, S.: Issue-based dialogue management. Ph.D. thesis, Department of Linguistics, Göteborg University (2002)
37. Larsson, S., Traum, D.: Information state and dialogue management in the TRINDI dialogue move engine toolkit. Natural language engineering 6(3&4), 323–340 (2000)
38. Levin, J.A., Moore, J.A.: Dialogue games: Metacommunication structures for natural language interaction. Cognitive Science 1(4), 395–420 (1977)
39. Loisel, A., Duplessis, G.D., Chaignaud, N., Kotowicz, J.P., Pauchet, A.: A conversational agent for information retrieval based on a study of human dialogues. In: Filipe, J., Fred, A. (eds.) Proc. of ICAART12. vol. 1, pp. 312–317. SciTePress (2012)
40. Mann, W.: Dialogue games: Conventions of human interaction. Argumentation 2(4), 511–532 (1988)
41. Massaro, D., Cohen, M., Beskow, J., Cole, R.: Developing and evaluating conversational agents. Embodied conversational agents pp. 287–318 (2000)
42. Maudet, N.: Modéliser les conventions des interactions langagières: la contribution des jeux de dialogue. Ph.D. thesis, Toulouse 3 (2001)
43. McTear, M.: Spoken dialogue technology: toward the conversational user interface. Springer-Verlag New York Inc (2004)
44. Miller, G., Beckwith, R., Fellbaum, C., Gross, D., Miller, K.: Introduction to wordnet: An on-line lexical database*. Int. Journal of lexicography 3(4), 235 (1990)
45. Pauchet, A., El Fallah-Seghrouchni, A., Chaignaud, N.: Simulating a human cooperative problem solving. In: Proc. of CEEMAS. pp. 225–235. Leipzig, Allemagne (2007)
46. Pelachaud, C.: Modelling multimodal expression of emotion in a virtual agent. Philosophical Trans. of the Royal Society B: Biological Sciences 364(1535), 3539 (2009)
47. Picard, R.: Affective computing: From laughter to ieee. IEEE Transactions on Affective Computing 1, 11–17 (2010)
48. Polanyi, L., Scha, R.: A syntactic approach to discourse semantics. In: Proceedings ICCL84. pp. 413–419 (1984)
49. Schroder, M.: The semaine api: towards a standards-based framework for building emotion-oriented systems. Advances in HCI 2010, 2–2 (2010)
50. Searle, J.: Speech Acts: An Essay in the Philosophy of Language. Cambridge University (1969)
51. Serban, O., Pauchet, A., Pop, H.: Recognizing emotions in short text. In: Filipe, J., Fred, A. (eds.) Proc. of ICAART12. vol. 1, pp. 477–480. SciTePress (2012)
52. Strapparava, C., Mihalcea, R.: Learning to identify emotions in text. In: Proc. of ACM SAC08. pp. 1556–1560. ACM (2008)
53. Strapparava, C., Valitutti, A.: WordNet-Affect: an affective extension of WordNet. In: Proc. of LREC. vol. 4, pp. 1083–1086. Citeseer (2004)
54. Swartout, W.R., Gratch, J., Jr., R.W.H., Hovy, E.H., Marsella, S., Rickel, J., Traum, D.R.: Toward virtual humans. AI Magazine 27(2), 96–108 (2006)
55. Von Luxburg, U.: A tutorial on spectral clustering. Statistics and Computing 17(4), 395–416 (2007)
56. Wahlster, W.: Smartkom: Symmetric multimodality in an adaptive and reusable dialogue shell. In: Proc. of the Human Computer Interaction Status Conference. vol. 2003, pp. 47–62. Berlin, Germany: DLR (2003)
57. Wiebe, J., Mihalcea, R.: Word sense and subjectivity. In: Proc. of ICCL06. pp. 1065–1072. Association for Computational Linguistics (2006)
58. Winograd, T., Flores, F.: Understanding Computers and Cognition: A New Foundation for Design. Ablex (1986)