Personality and Emotions – Finite State Modelling by Dialogue Automata

Ivan Kopeček
Faculty of Informatics,
Masaryk University Brno
Botanická 68a, 602 00 Brno,
Czech Republic
E-mail: kopecek@fi.muni.cz

ABSTRACT

The paper deals with an algebraic approach to the problem of modelling basic psychological terms, like personalities and emotions. We describe basic structure of dialogue automata that are used as user models, their basic relevant properties and we demonstrate how they can be applied to the modelling psychological terms.

1. INTRODUCTION

The research concerning human-computer interaction, user modelling and dialogue systems cover a wide range of approaches. Moreover, practical implementations and applications in real dialogue systems demand solutions of many tasks of different types, often having very interdisciplinary features.

In this situation, there is a natural need to use general models that can be applied to a wide variety of related problems, both theoretical and practical. Dialogue automata can be used as a general framework for this task [7].

The concept of dialogue automata use simple algebraic structures relating internal states of a participant of a dialogue (agent) with the attributes of the dialogue steps by means of transition and output functions.

The possibility to determine unknown attribute values in the context of the dialogue communication can be used for finding unknown or unsure parameters and values in the dialogue context. Suitable choice of the attributes of dialogue automata enables to model different instances of the dialogue communication.

Dialogue automata can relate emotions (that are described by the attributes of the internal states) with prosody (which is described by the attributes of the dialogue utterances) and other parameters that are entering the model. We can also analyse semantic ambiguity in a dialogue, etc.

In order to determine the structure of the dialogue automaton, we need to have sufficient data describing the internal states and dialogue utterances. To this aim we can inquire user, deriving the emotional state from facial expressions, using measurements, etc [see e.g. 12, 13].

In what follows we use standard terms and notation of algebraic theory of formal languages and automata. If M is a set, then M^* will denote the free monoid over the set M, i.e. the set of all strings consisting of the elements of the set M (including the empty string). An alphabet is a finite nonempty set. If U and V are sets, $U \times V$ denotes the Cartesian product of U and V.

2. INFORMATION SYSTEM, DIALOGUE AND RELATED NOTIONS

2.1. Formalized Information Systems

We apply the notion of information system to describe the properties of basic elements that we use to model users and dialogue communication. Formally, *information system* (in the sense of Pawlak, see e.g. [11]) is defined as follows.

Let U, T, V be nonempty sets and let f be a mapping of the set $U \times T$ into V. Then the ordered quadruple

$$\boldsymbol{S} = (U, T, V, f)$$

is said to be an information system. The elements of U are called objects, the elements of T attributes, and the elements of V values of attributes. Two objects u, v are called interchangeable if $f(u, t) = f(v, t)$ for all t from the set T. In what follows, we assume that no objects are interchangeable. Usually, it is assumed that the sets

U, T, V are finite. We do not assume it and use the notion of information system in a slightly generalized sense.

For our purposes (see Section 4.4) we introduce the operator ° defined as follows.

If $\boldsymbol{S}_1 = (U_1, T_1, V_1, f_1)$ and $\boldsymbol{S}_2 = (U_2, T_2, V_2, f_2)$ are information systems, then

$$\boldsymbol{S}_1 \circ \boldsymbol{S}_2 = (U, T, V, f)$$

is the information system that satisfies:

$$U = U_1 \times U_2,$$

$$T = T_1 \cup T_2,$$

$$V = V_1 \cup V_2 \cup (V_1 \times V_2),$$

and the function f is defined in the following way:

1. If $a \in T_1$ and $a \notin T_2$ then

$f((u_1, u_2), a) = f_1(u_1, a);$

2. If $a \in T_2$ and $a \notin T_1$ then

$((u_1, u_2), a) = f_2(u_2, a);$

3. If $a \in T_1 \cap T_2$ then

$f((u_1, u_2), a) = (f_1(u_1, a), f_2(u_2, a)).$

This operation can be used to derive new information systems from old ones and in connection with dialogue automata it appears to be a very useful tool to master complexity of user models.

Examples of information systems related to emotions and personality are presented in the following text.

2.2. Dialogue, Dialogue Strategy, Goal of the Dialogue

In this section we present formalization of the basic notions related to dialogue communication [5, 7].

If $\boldsymbol{U} = (U, T, V, f)$ is an information system, and $u \in U^*$, then u will be called *dialogue*.

This definition can be interpreted as follows: Let $u = (u_1, u_2, \ldots, u_n)$ be a dialogue. Then the odd indexed elements of u correspond to the utterances of the first participant of the dialogue and the even indexed elements correspond to the utterances of the second participant of the dialogue. Empty string corresponds to the empty dialogue.

Let $\boldsymbol{U} = (U, T, V, f)$ be an information system and # a special symbol not belonging to U. Then a function

$$s : \{U^* \cup \{\#\}\} \rightarrow \{U \cup \{\#\}\}$$

is called *dialogue strategy*. The set of all dialogue strategies related to $\boldsymbol{U}$ will be denoted by $Stg(\boldsymbol{U})$.

Suppose s is a dialogue strategy, $u \neq \#$ and $s(u) = v$. Then the interpretation is as follows: based on the present dialogue u, the dialogue strategy s determines the continuation of the dialogue, which is uv, (concatenation of u and v) provided $v \neq \#$, or terminates the dialogue, if $v = \#$.

Suppose now that $u = \#$ and $s(u) = v$. Then the interpretation is as follows: the dialogue strategy s determines the beginning element of the dialogue, which is v, provided $v \neq \#$, or terminates the dialogue, if $v = \#$ (resulting in the empty dialogue).

Let $s_1, s_2 \in Stg(\boldsymbol{U})$. We shall say that the ordered pair of strategies (s_1, s_2) *generates* the dialogue $u = (u_1, u_2, \ldots, u_n) \in U^*$ (or that the dialogue u is generated by (s_1, s_2)), if $s_1(\#) = u_1$, $s_2(u_1) = u_2$, ..., $s_1(u_n) = \#$ provided n is even or $s_2(u_n) = \#$ provided n is odd. We will use the notation $u = [s_1, s_2]$.

A function E from U^* into the set of real numbers will be called *evaluation function*. The set of all evaluation functions related to $\boldsymbol{U}$ will be denoted by $E(\boldsymbol{U})$. The evaluation function is a subjective evaluation of the dialogue by the user. Let $S_1, S_2 \subset Stg(\boldsymbol{U})$ and $E_1, E_2 \subset E(\boldsymbol{U})$. Then the ordered quadruple $M = (S_1, S_2, E_1, E_2)$ can be considered as a dialogue communication.

Goal of the dialogue for the first (second) participant of the dialogue is to minimize the function E_1 (E_2), i.e. to choose such a strategy $s_1 \in S_1$ ($s_2 \in S_2$) that $E_1([s_1, s_2]) = min$ ($E_2([s_1, s_2]) = min$). For instance, the participant could minimize the number of turns that is needed to obtain desired information, or minimize his or her lost in the dialogue.

Based on the previous notions we can characterize dialogues as: co*operative*, if $E_1 = E_2$; *non-cooperative*, if $E_1 \neq E_2$; *zero-sum*, if $E_1 = -E_2$. (This formalisms enables a direct application of the game theoretical approach, which seems to be promising especially if combined with formal models in social and behavioural sciences [4, 10]).

3. DIALOGUE AUTOMATA

3.1. Definition of Dialogue Automata

Dialogue automata generalize the notion of Mealy-type automata (see e.g. [3]) and can be seen as a very simple user model of the participant of dialogue. States of the automaton correspond to the internal states of the user, and input and output symbols correspond to the dialogue utterances. Transition functions represent the scheme of the user behaviour.

Let $\boldsymbol{S} = (S, A_S, V_S, f_S)$ and $\boldsymbol{X} = (X, A_X, V_X, f_X)$ be information systems. We will assume that the sets of attributes f_S and f_X are finite. *Dialogue automaton* is an ordered quadruple

$$\boldsymbol{A} = (S, X, \lambda, \delta),$$

where

$$\delta : S \times X \to S \text{ and } \lambda : S \times X \to X$$

are *transition function* and *output function*, respectively. Sets S and X are the *sets of states* and *dialogue utterances*, respectively. Being in the state $a \in A$, the automaton detects the actual input symbol (dialogue utterance of the other participant of the dialogue), changes its state according to the function δ and outputs a dialogue utterance according to the function λ. A dialogue automaton for which the set of the states and the set of the dialogue utterances are finite is said to be *finite*.

Provided that we have selected a state to be the initial state and that the special symbol # belongs to X, a dialogue automaton represents a dialogue strategy.

3.2. Example

In [7] the model is applied to dialogue communication between two gamblers playing poker, involving emotional and prosodic attributes. As an example, we present here the structure of the information systems used in this model. (A fragment of the transition function and further details can be found in [7])

Attributes of the states (AS):

AS1 - my_cards;

AS2 - bet_in_pot;

AS3 - self confidence;

AS4 - counter_player_cards;

AS5 – plan;

having the following attribute values (related to the corresponding attributes):

AS1 - my_cards: bad, medium, good;

AS2- bet_in_pot: high, medium, low;

AS3 - self_confidence: high, medium, low;

AS4 - counter_player_cards: bad, medium, good;

AS5 - plan: careful, risky, bluffing.

(The semantics of the states and attributes is described by the corresponding word meaning. Let us mention that AS4 should be understand as the estimation of the counter player cards in the present step of the dialogue.)

Attributes of dialogue utterances (AX):

AX1 – choice;

AX2 - prosody_self-confidence;

AX3 - prosody_credibility;

AX4 - prosody_excitement;

having the following attribute values (related to the corresponding attributes)

AX1 - choice: call, raise, fold;

AX2 - prosody_self-confidence: high, medium, low;

AX3 - prosody_credibility: high, medium, low;

AX4 - prosody_excitement: high, medium, low;

4. BASIC PROPERTIES OF DIALOGUE AUTOMATA

4.1. Deriving Internal Structure from Dialogue Corpus

Let A_1 and A_2 be dialogue automata. For the sake of simplicity, by A_1 (A_2) we will understand also the set of all corresponding strategies (i.e. strategies defined by the corresponding automata and a chosen initial state). In accordance with this convention, we denote by $[A_1, A_2]$ the set of all dialogues generated by A_1 and A_2 (for all possible initial states).

In reality, one of the most crucial problem in constructing such models lies in the fact that we do not know the internal structure of the used models – we have to consider them as black boxes and deduce or investigate their structure from their behaviour.

It can be shown [9] that (roughly speaking): given a (finite) set of dialogues D, we can construct (finite) dialogue automata A_1 and A_2 such that $D \subseteq [A_1, A_2]$. Moreover, minimizing their structures we obtain the corresponding canonical forms [3].

4.2 Determining Unknown Parameters and Solving Ambiguities

A powerful feature of the dialogue automata approach lies in the possibility to determine unknown attribute values in the context of the dialogue communication. It can be used for determining the unknown or unsure parameters and values in the dialogue context [8, 13].

4.3 Continuous Dialogue Automata

In many situations, we can represent the values of attributes of internal states and dialogue utterances in a numerical form. Real numbers, which express a measure of the intensity of an attribute, are a typical instance.

In such cases, we may obtain models, for which the number of internal states as well as the number of dialogue utterances is infinite. Then, we can assume

that the sets of attributes are indexed and that the values of attributes are real numbers. Hence, both internal states and dialogue utterance can be considered as real vectors. This leads us to the following problem:

Given a finite set of dialogues D and two finite sets of attributes (one set of attributes related to the states and other set of attributes related to dialogue utterances), find (infinite) dialogue automata $\boldsymbol{A}_1$ and $\boldsymbol{A}_2$ such that $D \subseteq [\boldsymbol{A}_1, \boldsymbol{A}_2]$, and the internal states of the automata coincide with the observed states for all the dialogues from the set D.

To obtain an approximate solution of this problem, we suppose that the functions $\delta : S \times X \to S$ and $\lambda : S \times X \to X$ are sufficiently smooth. Let us remark that this assumption is quite acceptable – it expresses that a small change of the attribute will also imply a small change of the function and that this relation is "smooth". Accepting this assumption, we can – at least locally – approximate the functions linearly.

Hence, we will suppose that the functions

$\delta_i : S \times X \to s_i$

and

$\lambda_j : S \times X \to x_j$

are bilinear forms for all

$I \in (1,\ldots,m), j \in (1,\ldots,n)$

$(s = (s_1, \ldots, s_m), x = (x_1, \ldots, x_n), s \in S, x \in X)$.

Under this assumptions, the functions δ_i and λ_j can be represented by matrices $\boldsymbol{D}_i$ and $\boldsymbol{L}_j$ in the form: $\delta_i (s, x) = s \boldsymbol{D}_i x$, $\lambda_j (s, x) = s \boldsymbol{L}_j x$.

Thus, if we have sufficient data (i.e. a set of dialogues for which the corresponding utterances are classified by means of the attributes and the set of corresponding states, also classified by means of attributes), we can compute all the matrices $\boldsymbol{D}_i$ and $\boldsymbol{L}_j$, generally from an overdetermined system of linear equations by means of Least Square Method.

4.4. Products and Decomposition of Dialogue Automata

Let $\boldsymbol{A}_1 = (S_1, X_1, \lambda_1, \delta_1)$, and $\boldsymbol{A}_2 = (S_2, X_2, \lambda_2, \delta_2)$, be dialogue automata. Product of these automata is the automaton

$$\boldsymbol{A}_1 \otimes \boldsymbol{A}_2 = \boldsymbol{A} = (S, X, \lambda, \delta),$$

where:

$$S = S_1 \circ S_2,$$

$$X = X_1 \circ X_2,$$

(The operation $\circ$ is described in Section 2.1)

The transition and output functions are defined as follows:

$$\delta ((s_1, s_2), (x_1, x_2)) = (\delta_1 (s_1, x_1), \delta_2 (s_2, x_2)),$$

$$\lambda ((s_1, s_2), (x_1, x_2)) = (\lambda_1 (s_1, x_1), \lambda_2 (s_2, x_2)).$$

Intuitively, making product of two dialogue automata means that we create a new dialogue automaton that combines the properties of the original automata.

Hence, these techniques can use for deriving new models from basic ones. Moreover, we can use it to define decomposition [9] of automata, which can be used for simplifying the automata structure. Roughly speaking, the automaton $\boldsymbol{A}$ can be decomposed into the automata $\boldsymbol{A}_1$, $\boldsymbol{A}_2$ if $\boldsymbol{A}$ is isomorphic with the product $\boldsymbol{A}_1 \otimes \boldsymbol{A}_2$.

5. MODELLING PERSONALITY AND EMOTIONS BY DIALOGUE AUTOMATA

4.1 Modelling Emotions

To illustrate how formal information systems can be used to describe emotions, we adopt a simple model presented in [1].

In this model the current emotional state is characterized by valence (representing overall happiness), that can be positive (happy), neutral or negative (sad), and arousal (representing the intensity level of emotions, that can be excited, neutral or calm. The corresponding information system (U, T, V, f) can be expressed as follows:

U = {Angry, Joyful, Sad, Relaxed, E, C, P, N, NN}

T = {Valency, Arousal}

V = {Negative, Positive, Excited, Calm, Neutral}

The function f is defined as follows:

f(Angry, Valency) = Negative;
f(Angry, Arousal) = Excited;
f(Joyful, Valency) = Positive;
f(Joyful, Arousal) = Excited;
f(Sad, Valency) = Negative;
f(Sad, Arousal) = Calm;
f(Relaxed, Valency) = Positive;
f(Relaxed, Arousal) = Calm;
f(E, Valency) = Neutral;
f(E, Arousal) = Excited;
f(C, Valency) = Neutral;
f(Joyful, Arousal) = Calm;
f(P, Valency) = Positive;
f(P, Arousal) = Neutral;
f(N, Valency) = Negative;
f(Relaxed, Arousal) = Neutral;
f(NN, Valency) = Neutral;
f(NN, Arousal) = Neutral;

Encapsulated in a dialogue automaton, this can be used to describe user models, as we can see in what follows.

4.2 Modelling Personality

Based on the information system defined in the previous section we illustrate how basic types of personalities can be modelled by dialogue automata.

Let $S = (U, T, V, f)$ where U, T, V and f are defined as in the previous Section 4.2. Further, suppose that an information system $X = (U_X, T_X, V_X, f_X)$ is defined as follows:

$U_X = \{N0, N1, N2, P1, P2\}$
$T_X = \{\text{meaning}\}$

$V_X = \{\text{neutral_news, negative_news, very_negative_news, positive_news, very_positive_news}\} \cup U$

$f_X(N0) = \text{neutral_news};$
$f_X(N1) = \text{negative_news};$
$f_X(N2) = \text{very_negative_news};$
$f_X(P1) = \text{positive_news};$
$f_X(P2) = \text{very_positive_news};$

Now, we can create a model A_C of a choleric and a model A_P of a phlegmatic by assuming

$\lambda_C(s, x) = s,$
$\lambda_P(s, x) = s$

(for the sake of simplicity we suppose that the output function just outputs output symbols corresponding to the actual state of the dialogue automaton)

and by defining the corresponding transition functions δ_C and δ_P as follows:

definition of δ_C	**N0**	**N1**	**N2**	**P1**	**P2**
Angry	E	Angry	Angry	Joyful	Joyful
Joyful	E	Angry	Angry	Joyful	Joyful
Sad	N	Angry	Angry	Joyful	Joyful
Relaxed	P	Angry	Angry	Joyful	Joyful
E	E	Angry	Angry	Joyful	Joyful
C	NN	Angry	Angry	Joyful	Joyful
P	P	Angry	Angry	Joyful	Joyful
N	N	Angry	Angry	Joyful	Joyful
NN	NN	Angry	Angry	Joyful	Joyful

Table 1. Definition *of* λ_C

definition of δ_P	**N0**	**N1**	**N2**	**P1**	**P2**
Angry	N	Angry	Angry	NN	NN
Joyful	P	NN	NN	Joyful	Joyful
Sad	N	Sad	Sad	NN	P
Relaxed	P	NN	Sad	P	P
E	NN	N	Angry	P	Joyful
C	C	NN	N	P	P
P	NN	N	Sad	P	Joyful
N	N	Sad	Angry	NN	P
NN	NN	NN	N	P	Joyful

Table 2. Definition *of* λ_P

Clearly, this is not a "canonical" model but merely an illustration. These tables aroused just by the author's guess of how the choleric or phlegmatic could react in a given situation. We can see that the definition of the transition function for both cases can be obtained in many ways. Similarly, we could create models for sanguine and melancholic as well.

Of course, to obtain more realistic models, we have to get the information about the structure of the models from measured data.

On the other hand, we can also compare our guesses with real data and we can try to get canonical forms of the personality models, i.e. the models from the other ones can be derived (by the technique of products and decomposition or some other means).

6. CONCLUSIONS

In the paper we have outlined a possible approach to emotions and personality, which is based on finite state analysis. This approach offers a general view in which formal (especially algebraic) methods and results can be applied to a wide variety of different problems.

To be able to use effectively this approach in the theoretical analysis and practical implementation of dialogue systems and user models, we have to enhance the technique of obtaining the models from data and combining and deriving modified models from basic prototypes. This is our main task in the future research.

ACKNOWLEDGEMENT

The research has been partially supported by Grant Agency of the Czech Republic under the Grant 201/99/1248.

REFERENCES

1. Ball, G., Breese, J. "Modeling the Emotional State of Computer Users", Proceedings of 1st Workshop on Attitude, Personality and Emotions in User-Adapted Interaction, Banff, Canada, 2000.
2. Batusek, R., Kopeček, I. "User Interfaces for Visually Impaired People". Proceedings of the 5th ERCIM Workshop on User Interfaces for All, Dagstuhl, pp. 167 – 173, 1999.
3. Gecseg, F., Peak, I. *Algebraic Theory of Automata.* Akademiai Kiado, Budapest , 1972.
4. Greenberg, J. "Situation Approach to Cooperation". in Hart. S, Mass-Colell, A. (Eds.): *Cooperation: Game-Theoretic Approaches*. NATO ASI Series, Springer Verlag, 1997.
5. Kopeček, I. "Modelling of the Information Retrieval Dialogue Systems". Proceedings of the Workshop on Text, Speech and Dialogue - TSD'99, LNAI 1692, Springer-Verlag, pp. 302-307, 1999.
6. Kopeček, I. "Active and Passive Strategies in Dialogue Program Generation". Proceedings of the Workshop on Text, Speech and Dialogue – TSD 2000, LNAI 1902, Springer-Verlag, pp. 427-432, 2000.
7. Kopeček, I. "Emotions and Prosody in Dialogues: An Algebraic Approach Based on User Modelling". Proceedings of the ISCA Workshop on Speech and Emotions. Belfast, pp. 184-189, 2000.
8. Kopeček, I., Pala, K., Stranakova-Lopatkova, M.: "Ambiguity Problems in Human-Computer Interaction", to appear in Proceedings of UAHCI 2001 Conference, New Orleans, 2001.
9. Kopeček, I., "Dialogue Automata", to appear in FI MU Technical Report Series, 2001.
10. Osborne, M.J., Rubinstein, A. *A Course in Game Theory*. MIT Press, London, 1999.
11. Pawlak, Z. "Information Systems, Theoretical Foundations". *Information Systems 6* (1981), pp. 205-218, 1981.
12. Pittam, J., Gallois, C., Callan, V. "The long term spectrum and perceived emotion", Speech Communication, 9, pp. 177-187, 1990.
13. Scherer, K., Koivumaki, J., Rosenthal, R. "Minimal cues in the vocal communication of affect: Judging emotions from content-masked speech." Journal of Psycholinguistic Research, 1, 269-285, 1971.
14. Stranakova, M. (1999). "Selected Types of Pg-ambiguity". *Prague Bull. of Math. Linguistics 72*, pp. 29-58.