An Interaction Model for Affect Monitoring

I. Song, G. Governatori, and R. Colomb

School of Information Technology & Electrical Engineering,
The University of Queensland, Brisbane, QLD, 4072, Australia
{insu, guido, colomb}@itee.uq.edu.au

Abstract. This paper investigates how we can precisely define what process designers are ought achieve for what they have promised and more importantly in a way that satisfies human users. Toward these goals, an interaction model for processes and an Affect Monitoring Framework (AMF) are proposed based on our analysis on speech act theory and cognitive-based emotion models. The Affect Monitoring Framework is to detect and predict negative affects on users and to resolve caused or predicted causes of negative affects automatically.

1 Introduction

For any businesses, it is critical to know and predict both negative and positive affects on the users interacting with the organizations managed by business process management systems such as workflow management systems.

One of the important affects on the users interacting with processes is *emotion* since emotions are motivational processes that influence cognition and actions [3]. Emotional states of users interacting with an information system can be caused by various failures and abnormal behaviors of the system, such as delayed responses, failed operations, missed notifications, and unpleasant actions. Many of these causes result in poor usability and frustration on the users [4].

This paper investigates how we can give objective semantics to what it means by failures and abnormal behaviors of processes in the view of human users. Further we want to precisely define what process designers are ought achieve for what they have promised and more importantly in a way that satisfies human users. Toward these goals, an interaction model for processes for an Affect Monitoring Framework (AMF) is proposed based on our analysis on speech act theory and cognitive-based emotion models. The framework is to detect and predict negative affects on users and to resolve caused or predicted causes of negative affects automatically.

In the next section we give an overview of AMF. Section 3 describes an interaction model for processes which lays out objectives of this paper and conditions for satisfiable processes. Section 3 and 4 develop methods to capture necessary information for estimating emotions. Section 5 describes an example emotion generation based on a cognitive emotion model.

G.I. Webb and Xinghuo Yu (Eds.): AI 2004, LNAI 3339, pp. 979–984, 2004.

2 Affect Monitoring Framework

Cognitive emotion theorists [6, 1, 7] claim that events relevant to users' concerns are the main sources of emotions. If this is true, we need to know what users' goals are. We can then enumerate the events that are relevant to the goals. We then need to formulate how these events give arise to emotions on the users.

We formulate this intuition into a framework called AMF which includes the following four components: (1) *Data-mining* monitors interactions between users and processes to collect necessary information for estimating possible causes of emotions, (2) *Emotion-generation* uses the collected information (users' goals, events, and actions) to estimate users' emotions based on an emotion model, (3) *Emotion-management* monitors emotional states of users and takes appropriate actions to prevent possible causes of negative emotions and to resolve negative emotions, and (4) *Emotion-adjustment* adjusts estimated emotional states from the direct feedback from users or other observation methods such as facial expression recognition, gesture recognition, emotion recognition from email messages.

The framework includes all three types of usability evaluation automation described in the taxonomy by Ivory et al. [2]: capture (data mining), analysis (emotion generation), and critique (emotion management). However, the framework focuses only on evaluating affects on the users of processes rather than evaluating the conventional usability measures such as performance, efficiency, easy of use, and easy to learn.

3 An Interaction Model

A user with a set Q of goals comes along to use the tools provided by a process to achieve a subset $q \subset Q$ of her goals. The system provides tools (some calls them mediators or services) through a set IF of user interfaces. The user uses her own planning ability to achieve q using a subset of IF. Given IF, which defines a set M of messages that can be exchanged between the user and the process, we can drive a set G of goals that can be created through IF. In this scenario, there are three primary questions we want to address:

1. How to make sure the process is designed the way that it can achieve all the goals in G.
2. How to make sure the process achieves the goals in G in a manner that satisfies human users. (We are not considering how well IF is designed to achieve q.)
3. How to *monitor* whether processes achieve all the goals in G in a way that satisfies human users.

The designer of a process must design the database DB and a set of plans of the process in the way that the process satisfies the first two questions. Although design time tests can address the first question, it cannot address the second question because of various runtime factors. Therefore, we need to find a way to address the third question.

3.1 Process Definition

A process is defined as a structure:

$$Pr =_{def} < IF, M, G, DB, L, R, f_L(), \hbar(), PL >$$

where IF is the set of interfaces, M is the set of all messages between the process and users, G is the set of users' goals that can be created through IF, DB is the database of the process, L is the set of predicate symbols and function symbols that define the language of the process, R is a set of rules, $f_L(DB)$ is a function that maps DB to a set S of grounded atoms, $\hbar(p)$ is a function that maps a conjunction of grounded atoms to a set of records, PL is a set of plans which achieves goals in G. Let $T = S \cup R$ be the theory of the process.

A message $m \in M$ can contain several illocutionary acts α each of which is represented as $F(p)$ where F is the illocutionary force and p is the proposition of α. Then, the semantics of illocutionary acts received from users can be given with respect to the process definition as follows: (1) If F is a directive, the user wants the system to bring about p. Therefore, the goal state p is a constraint on DB that $T \vdash p$. (2) If F is an assertive or a declarative, the user wants the system to believe p. Therefore, $(\hbar(p) \subseteq DB)$ is the goal state, i.e., the database contains information p and consequently $T \vdash p$. (3) If F is a commissive, $(\hbar(I(u,p)) \subseteq DB)$ is the goal state, i.e., u is intending p and u wants the system believe it.

The semantics of commissive messages sent to users are equivalent to the directive messages from users: the goal is $T \vdash p$. Assertive and declarative messages to users are usually informing events if they are related with the goals of the users, otherwise they are actions performed by the system. Directive and expressive messages are usually actions performed by the system.

The interface consists of eight sets of grounded well formed formulas: IF^i_F for incoming messages and IF^o_F for outgoing messages where the subscripts $F \in \{d, a, c, dc\}$ stand for directive, assertive, commissive, and declarative illocutionary force, respectively. We define three types of goals for a message $(sender, receiver, F(p))$: $(T \vdash p)$, $(\hbar(p) \subseteq DB)$, and $(\hbar(I(sender,p)) \subseteq DB)$ where $p \in IF$. The set G of users' goals that this process must fulfill is defined as follows:

$$G = \{(T \vdash p) | p \in IF^i_d\} \cup \{(\hbar(p) \subseteq DB) | p \in IF^i_a \cup IF^i_{dc}\} \cup \{(\hbar(I(u,p))| p \in IF^i_c\} \cup \{(T \vdash p) | p \in IF^o_c\}$$

Now, we impose constraints on DB and PL:

1. For all $g \in G$, there must exist a state of DB so that g is true.
2. For all $g \in G$, there must exist a plan $pl \in PL$ whose execution will lead to a state of DB that makes g true.

Those constraints are necessary conditions for a process to achieve all the goals that can be created through its interfaces. But, this does not tell us whether the process achieves the goals in a way that satisfies human users. The following sections, 4, and 5 develop a method to monitor whether the process achieves the goals in a way that satisfies human users.

Table 1. Prospective events for a goal g and their classifications

Event Names	**Symbols**	**OCC Types**	**Symbols**
goal failure time event	gte_g	Prospective	all
response failure time event	rte_g	Unexpected	none
confirming event	ice_g	Desirable	ice_g
disconfirming event	ide_g	Undesirable	gte_g, rte_g, ide_g
informing new time event	ite_g	Unconfirmed	rte_g, ite_g
response event	ire_g	Confirming	ice_g
		Disconfirming	ide_g

4 Events

For a process there are two types of goals concerning with its users: user requested goals and promised goals by the process. A requested goal is created when a user sends a message to the process. A promised goal is created when the process sends a message containing a commissive speech act. Given the two types of goals, we enumerate the events relevant to the goals.

When a user interacts with the system, the user is aware of all prospective events related with the messages. Therefore, the two types of goals trigger *user-side-time-events,* $TE_g = \{gte_g, rte_g\}$. The system responsible for the goals struggles to prevent the time events occurring by producing *informing-events,* $IE_g = \{ice_g, ide_g, ite_g, ire_g\}$. Table 1 lists these events and shows the classification based on Ortony, Collins and Clore (OCC) [6].

4.1 User Side Time Events

We make the following assumptions for the two types of goals. For a requested goal g, the user is expecting a response within a certain response time rt_g. The response must be either a goal achievement confirming event ice_g or a response event ire_g. If the response is not a confirming event, the user is expecting another response for a confirming event within a certain goal achievement time gt_g. The new response must be either a confirming event ice_g or an informing new time events ite_g which resets gt_g to $gt_g + \delta$ for some value $\delta > 0$. We assume that $gt_g > rt_g$ is usually the case. For a promised goal g, the user is expecting a response within a certain response time rt_g. The response must be either a confirming event ice_g or an informing new time events ite_g which resets rt_g to $rt_g + \delta$.

When the user is not informed of the achievement of the goal within gt_g, a goal failure time event gte_g fires. When neither the achievement nor an acknowledgement is informed to the user within rt_g, a response failure time event rte_g fires. When a process promises that a promised goal g will be satisfied within in a certain time rt_g and the process fails to inform the user within the time whether the goal is satisfied or a new response time rt_g is set, a response failure time event rte_g fires.

Therefore, for any goal g, $TE_g = \{gte_g, rte_g\}$ is the set of all possible time events that can cause negative effects on the user of the goal if the process does not inform appropriately the user. If g is a requested goal, the set of possible user side time events is $\{gte_g, rte_g\}$. If g is a promised goal, the set of possible events is $\{rte_g\}$.

4.2 Informing Events

The process must create appropriate informing events to prevent the user side time events occurring. We define four types of informing events: confirming events ice_g, disconfirming events, ide_g, informing new time events ite_g, and response events ire_g. These events are detected by examining the messages $F(p)$ sent to the users as follows:

$$ice_g \Leftarrow requested(g) \wedge (g = (T \vdash p)) \wedge (assertive(F) \vee declaritive(F))$$
$$ide_g \Leftarrow requested(g) \wedge (g = (T \vdash \neg p)) \wedge (assertive(F) \vee declaritive(F))$$
$$ire_g \Leftarrow requested(g) \wedge (g = (T \vdash p)) \wedge (commisive(F))$$
$$ite_g \Leftarrow (requested(g) \vee promised(g)) \wedge (g = (T \vdash p)) \wedge (commisive(F)) \wedge ire_g$$

If a confirming event ice_g or a disconfirming event ide_g occurs, no events in TE_g will fire anymore. The following two formulas summaries the event firing rules for the user side time events described in the previous subsection:

$$rte_g \Leftarrow (rt_g < t) \wedge \neg ire_g \wedge \neg ite_g \wedge \neg ice_g \wedge \neg ide_g$$
$$gte_g \Leftarrow (gt_g < t) \wedge \neg ite_g \wedge \neg ice_g \wedge \neg ide_g.$$

5 Emotion Generation

This section describes how emotional states can be deduced from the information captured in the previous sections based on the work of [5]. We only consider a subset of the OCC [6] cognitive appraisal theory of emotion: hope, satisfied, fear, fears-confirmed, disappointment, and reproach. These emotions are prospective-based emotions that are emotions in response to expected and suspected states and in response to the confirmation or disconfirmation of such states [6].

Given a set of events and a set G of goals captured for a user, we can derive the following emotions of the user for a goal $g \in G$:

$$hope(g) \Leftarrow requested(g) \wedge \neg(rte_g \vee gte_g \vee ice_g \vee ide_g) \quad (1)$$
$$fear(g) \Leftarrow requested(g) \wedge rte_g \wedge \neg(gte_g \vee ice_g \vee ide_g) \quad (2)$$
$$fear(g) \Leftarrow promised(g) \wedge rte_g \wedge \neg(ice_g \vee ide_g) \quad (3)$$
$$satisfied(g) \Leftarrow hope(g) \wedge \neg fear(g) \wedge ice_g \quad (4)$$
$$fearConf(g) \Leftarrow fear(g) \wedge request(g) \wedge (gfe_g \vee ide_g) \quad (5)$$
$$fearConf(g) \Leftarrow fear(g) \wedge promised(g) \wedge ide_g \quad (6)$$
$$disappoint(g) \Leftarrow hope(g) \wedge (gfe_g \vee ide_g) \quad (7)$$
$$relieved(g) \Leftarrow fear(g) \wedge ice_g \quad (8)$$

(Eq. 1) says that if there is a goal that is desirable to the user and no fear prospect is triggered, the user feels hope over the goal. (Eq. 2 & 3) says that if there is a goal that is desirable to the user and a fear prospect is triggered for the goal, the user might feel fear over the failure of the goal. (Eq. 4) says that if the user felt hope of an event and a confirming event occurs, the user might feel satisfied. (Eq. 5 & 6) says that if the user felt fear of an event and a disconfirming event occurs, the user might feel that the fear is confirmed. (Eq. 7) says that if the user felt hope of an event and a disconfirming event occurs, the user might be disappointed. (Eq. 8) says that if the user felt fear of an event and a disconfirming event of failure occurs for the event, the user might be relieved.

6 Conclusion

This paper proposed a human-process interaction model based on speech act theory and the cognitive-based emotion model of OCC. The model allows us to specify processes user-oriented way and observe interactions to monitor not only whether it achieves users' requests, but also in a way that satisfies human users. The model also clearly defines the requirements of the database and procedures of a process for a set of interfaces defined for the process.

We have also described how the goals of the users interacting with an information system can be captured and how such goals can be used to define events that can be used in detecting affects on the users. We believe that the framework and the model provided is independent of culture, education, and context of users. Although we have shown an emotion generation method based on the OCC model, we believe the information captured can be used with most of other cognitive-based emotion models.

References

1. Frijda, N.H.: The Emotions. Cambridge University Press (1986)
2. Ivory, M.Y., Hearst, M.A.: The state of the art in automating usability evaluation of user interfaces. ACM Computing Surveys **33** (2001) 470–516
3. Izard, C.E.: Four systems for emotion activation: Cognitive and noncognitive processes. Psychological Review **100** (1993) 68–90
4. Klein, J., Moon, Y., Picard, R.W.: This computer responds to user frustration. In: Proceedings of ACM CHI 99. Volume 2. (1999) 242–243
5. O'Rorke, P., Ortony, A.: Explaining emotions. Cognitive Science **18** (1994) 283–323
6. Ortony, A., Clore, G.L., Collins, A.: The Congnitive Structure of Emotions. Cambridge University, Cambridge (1988)
7. Roseman, I.J., Spinde, M.S., Jose, P.E.: Appraisals of emotion-eliciting events: Testing a theory of discrete emotions. Journal of Personality and Social Psychology **59** (1990) 899–915