An Adaptive, Emotional, and Expressive Reminding System

Nadine Richard and **Seiji Yamada**
National Institute of Informatics
2-1-2 Hitotsubashi, Chiyoda-ku
Tokyo-to 101-8430, Japan
nadine@nii.ac.jp and seiji@nii.ac.jp

Abstract

We are currently developing an adaptive, emotional, and expressive interface agent, which learns when and how to notify users about self-assigned tasks and events. In this paper, we describe the learning system and the user feedback mechanism we have designed. Then, we discuss issues concerning the expression of emotions, in the situation where the user should not be distracted by an adaptive tool and is not expected to create a strong relationship with it.

Introduction

An interface agent is a mediator between an automated service and a human being. The interaction with such an agent extends from a simple command line to a natural conversation with a human-like digital actor or a robot. As a particular case of interface agents, personal assistants help us reducing the ever-growing load of information, events and various commitments we need to handle, for instance by learning how to organize and keep track of relevant items.

When designing a context-aware, interactive learning system, two major issues are:

- Selecting an appropriate set of input values, including information about the current context of both the user and the application.
- Choosing a feedback mechanism, to let the user reward or punish adaptive components after they respectively lead to correct or incorrect actions.

In the case of an expressive interface agent, an additional question is whether the displayed emotions can influence the user positively when interacting with the agent.

In this paper, we present our on-going work on an adaptive, emotional, and expressive assistant for reminding of self-assigned tasks and events. TAMACOACH is an interface agent that adapts to the organizational skills and preferences of a user by learning *when and how* to present notifications, instead of requiring the user to set explicit alarms. Figure 1 presents a typical view of the TAMACOACH GUI.

On the one hand, the triggering of a reminder depends on:

- The relative temporal distance to the event starting date or to the task due date.

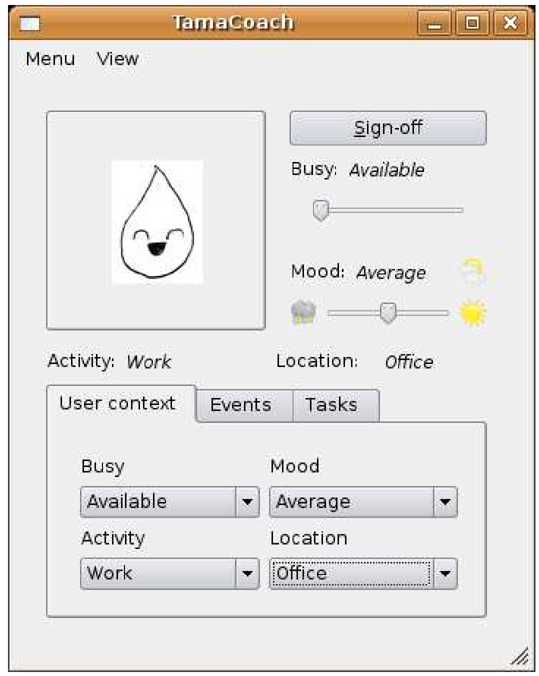

Figure 1: The main window of the TamaCoach GUI.

- Various attributes that help to classify user habits and preferences, *e.g.* contact information, priority or categories.
- The current status of the user: availability, on-going activity, physical location, and mood.

On the other hand, the form of a reminder mainly depends on the user status and on the capabilities of the device on which TAMACOACH is running. A reminder can be presented in different but combinable forms: a pop-up dialogue box, an e-mail message, a mobile e-mail message, or a sound alarm. The necessary data about tasks and events are extracted from ICALENDAR files, which are produced by an external calendaring or todo-management application. We gather information about the current context of the user through our GUI, and information about the agent execution context through the operating system of the host device.

As users may interact infrequently with their calendaring application, designing an appropriate feedback system for the learning module is a delicate issue. TAMACOACH learns from both:

- Explicit interaction, through the dedicated GUI. For now, user actions in TAMACOACH are limited to responding to a notification.
- Implicit interaction, through the user's favourite calendaring application. User actions on calendar data include adding a task, modifying a deadline, postponing a phone call, indicating the completion of a task, *etc.*

For the explicit interaction, we propose to simply present a notification together with a restricted set of user replies (*e.g.* accept or ignore), in order to get a direct but less-intrusive feedback from the user about the usefulness of the reminding system.

In the context of TAMACOACH, we are also investigating the effectiveness of expressed emotions regarding two separate goals:

- Representing the self-satisfaction level of the agent in a natural and non-intrusive way, *i.e.* giving feedback on the self-evaluation of the agent usefulness for the user.
- Influencing the user's behaviour by inducing emotional responses, at least to incite him/her to interact with our reminding system.

In order to situate the purpose of our reminding system, we first present some related work about personal assistants and expressive interface agents. We then describe the overall architecture and the learning system of TAMACOACH, especially the contextual input data and the simple mechanism that gets feedback from the user. Before concluding and presenting future work, we discuss the usual role of displaying emotions in virtual agents, and clarify the specificity of our approach.

Yet Another Personal Assistant

Robotic or software personal assistants can be adaptive recommenders for music or movies, helpers for home appliances, e-mail filters, meeting negociators, exercise coachs, TV presentors, museum guides, shopping assistants, *etc.* Categories of issues concerning personal assistants include machine-learning, interaction modalities, context-awareness, emotional display, interruption strategies, and the social impact of affective agents. Interface agents have been extensively studied by researchers from various disciplines: agent-based systems and interface design of course, but also psychology and sociology (Middleton 2001; Moldt & von Scheve 2001; Shiaffino & Amandi 2002). In particular, S. Shiaffino and A. Amandi state, in an excellent survey, the main issues for building a virtual secretary: the need for personalization, as each user would like to interact with a different kind of assistant, and the reaction of users towards interruptions, errors and explicit requests for feedback (Shiaffino & Amandi 2003).

Personal Time Managers

The management of personal calendars and todo lists is difficult, in both personal and professional contexts: it requires remembering on-going activities, and continuously rearranging priorities. People tend to remember quite well the main tasks and events they are involved in, but they have to face a growing volume and variety of information and commitments, as well as more and more sources of interruptions.

As pointed out in (Berry *et al.* 2006), human time management has an intensely personal nature. People are usually reluctant to delegate this specific task to others, and even more to software. They also have different preferences and practices regarding how they schedule their time. Moreover, people tend to use various media to keep track of things they intend to do (from paper post-its to mobile e-mails), and often do not record all the tasks and events they should remember (Bellotti *et al.* 2004). Interruption strategies are an important issue, as the task performance of a human may dramatically decrease when disrupted (Cutrell, Czerwinski, & Horvitz 2001; Shiaffino & Amandi 2003; Czerwinski, Horvitz, & Wilhite 2004; Weber & Pollack 2005; Horvitz 2007). Therefore, a good personal time manager should keep track of various sources of information and events through different time scales, and should be aware of the current context of its user in terms of activity, availability, location or even emotional state, in order to notify him/her when appropriate.

Calendars and todo managers are used to collect, maintain and organize lists of self-assigned tasks and events, with different natures, durations, regularities and frequencies. Electronic calendars and todo managers are different but closely related applications, often part of PIM (Personal Information Management) software suites. Within such tools, users can set alarms for the most important events, such as meetings or deadlines.

Many desktop or PDA (Personal Digital Assistant) applications, like SUNBIRD or EVOLUTION, propose traditional ways of organizing todo-lists and calendars, but do not take into account the multiple sources of task-oriented information. To overcome this lack, V. Belloti *et al.* have developed TASKVISTA (Bellotti *et al.* 2004) and TASKMASTER (Bellotti *et al.* 2003). TASKVISTA is a light-weight task list manager, which reduces the cognitive overload of information and events coming from different sources (*e.g.* e-mail, todo manager and Web browsing). TASKMASTER enables users to keep track of threads of activity and discussions, manage deadlines and reminders, and mark-up tasks directly within a single e-mail application. However, those applications do not automatically adjust to the specific organizational habits of their users.

On the other hand, adaptive systems like PTIME (as part of the PEXA assistant) learn about user scheduling habits and preferences, mostly in order to autonomously negotiate meetings (Berry *et al.* 2005; 2006; Myers *et al.* 2006). Like most of the investigated office- or healthcare-related personal assistants, PEXA is a cognitive agent that learns and reasons about tasks, user behaviour and its own behaviour, in order to justify its actions, answer questions and give advice. Interestingly, it also performs some scheduling tasks on behalf of the user.

Finally, AUTOMINDER is an adaptive reminding system, intended for cognitively impaired elders, who prefer to live at home (Pollack *et al.* 2003). It learns about routine activities, and monitors their daily performance in order to issue reminders whenever an essential task is not executed on time.

Expressive Assistants

It has been demonstrated that the representation of an embodied agent strongly influences the motivation and the performance of the user when interacting with the application (Dehn & van Mulken 2000; Bartneck 2002). As advocated in (Moldt & von Scheve 2001), interface agents should be equipped with emotional components, as emotions are an efficient mechanism for reducing the complexity of interpersonal interactions. Displaying emotions through facial expressions and gestures for virtual agents has now been widely explored (Bartneck 2002; Isbister & Doyle 2002). However, most applications concern ECAs (Embodied Conversational Agents), with complex dialogue abilities that simulate human behaviours. In particular, relational agents build and maintain long-term social-emotional relationships with their users. For instance in (Bickmore & Mauer 2006), the authors conducted experiments with various interfaces (text-only, static image and animated virtual agent) for relational agents on PDAs. Animations include facial displays of emotion, head nods, eye gaze movements, and posture shifts. The experiments show that users tend to create relationships more easily with animated agents. Such human-like agents are mainly developed for e-commerce or marketing (negociation, selling), healthcare (phobia or autism therapy, anxiety or exercise advisor), education (personal tutor) and entertainment (games, story-telling).

Nevertheless, the positive impact of embodied agents that express emotions still needs to be proven. As claimed in (Dehn & van Mulken 2000), it also strongly depends on the application and on the kind of agent and representation considered. For instance, in some stressful situations, users can feel more comfortable when supported by an empathetic assistant (McQuiggan & Lester 2006; Prendinger & Ishizuka 2005; Creed & Beale 2006). On the other hand, R. Rickenberg and B. Reeves have observed that the social presence of an anthropomorphic agent can reduce task performance of users, in particular when the assistant seems to monitor their work (Rickenberg & Reeves 2000). The agent representation can also mislead the users, and cause them to overestimate what the agent can do: an animated humanoid face is expected to conduct a dialogue in natural language, which would not be the case if the agent was represented as a dog.

For more information about emotion-based architectures for autonomous agents and human-machine interfaces, recent reviews can be found in (Sarmento 2004) and (Spinola de Freitas, Gudwin, & Queiroz 2005).

Virtual coachs

As a psychological process, *coaching* aims at pushing a person to achieve a particular task. A sport coach trains a team, but essentially supports its members to lead them to victory. In a company, the coach is a psychological expert, who temporarily helps employees to improve their work habits. The coach uses various psychological prises to help subjects in finding, by themselves, their ways for improvement.

Although widely investigated for tutoring or healthcare advising (Rickenberg & Reeves 2000; Nabeth *et al.* 2005; Creed & Beale 2006; Bickmore & Mauer 2006), the psychological aspects of motivating a user have yet barely pertained to the specific domain of time management assistance.

TamaCoach

Our adaptive reminding system, TAMACOACH, learns when to interrupt the user, depending on a categorization of tasks and events, and on the current context (time, user status, host device). Our goal is not to help users in maintaining a todo-list, but to learn when and how to remind them about what they have explicitly planned to do. However, such an application is a complement to more complex assistants like PEXA, TASKVISTA or TASKMASTER.

Furthermore, an animated, emotional agent would be more engaging: reducing the anxiety over a long list of things to do, or inducing amusement or even guilt, should lead the user to better interact with TAMACOACH and with his/her calendaring application. We believe that the basic expressivity of a cartoon-like, virtual creature should be sufficient, while providing a less intrusive and less distractive interface for gathering user feedback than conversational human-like assistants.

As suggested by its name, the ultimate purpose of the TAMACOACH project is to investigate virtual coaching in the case of personal time management. Our mid-term goal is however to experiment on its adaptiveness and expressiveness.

Learning About Our Organizational Habits

The XCS-based learning system of TAMACOACH decides when to present a reminder to its user, and in what form. In order to trigger an appropriate reminder, it is necessary to extract relative temporal distances and additional values from ICALENDAR data and past experiences, and to incorporate the context of both the user and the application. Figure 2 exposes an overview of our general architecture.

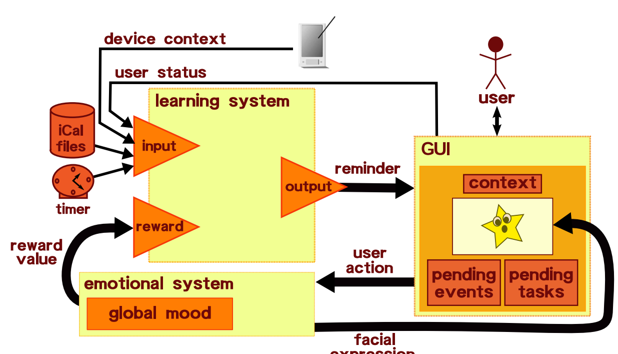

Figure 2: The overall architecture of TamaCoach.

Two Cascaded Learning Classifier Systems

A Learning Classifier System is a set of rules (*classifiers*) representing actions to be triggered depending on perceived situations. Each rule is a pair of condition-action elements, associated with a *fitness* value, which indicates the estimated efficiency of the rule. The condition part may contain *insignificant* items (environmental features that can be ignored), meaning that the rule can be general and match various situations. The fitness values are updated by incremental learning, depending on the success or the failure of the chosen rule for the current situation. Classifiers are added when new or more specific situations are detected. Then a Genetic Algorithm selects the best rules in the whole set, while maintaining the size of the rule population. The XCS model has proved to be an efficient, evolutionary, rule-based learning mechanism (Wilson 1995; Butz, Goldberg, & Lanzi 2003). Like other Learning Classifier Systems, XCS has been extensively used for data-mining and to control animats, but not to design interface agents.

TAMACOACH learns about two distinct functions: when to trigger a reminder, and how to present this notification to the user. As shown in figure 3, the learning module is composed of two XCS-based Classifier Systems, with different purposes. The first Classifier System (CS1) categorizes the data concerning a given calendar item. Its output is a priority value, which indicates how urgently TAMACOACH should produce a reminder for the item. Then, this priority value is used to create a situation vector for the second Classifier System (CS2).

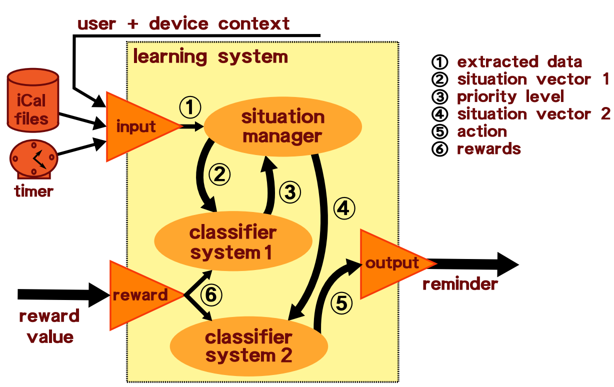

Figure 3: Architecture of the XCS-based learning system.

Depending on the current context of both the user and the host device, and on the priority given by the first set of classifiers, CS2 is in charge of deciding if triggering a reminder now is appropriate. If so, CS2 chooses the kind(s) of reminder to be produced. Because the possible reminder forms are compatible (*e.g.* TAMACOACH can launch a pop-up window and send an e-mail for the same item), the output of CS2 is a set of actions to be performed. As CS2 needs CS1 output to process the given situation, CS1 and CS2 are said to be cascaded.

Data From Calendar Files

Most of the conditional attributes in CS1 correspond to fields provided by the ICALENDAR format. ICALENDAR data are extracted and transformed before being stored into a local database. Some attribute values are also computed from past experiences stored in the database, *e.g.* the average delay in achieving a given category of task. Most of the values in the database take part in the decision process; in this case, before being sent to the learning system, the values are discretized. Categorical information like priority, transparency, user-defined categories, or involved contacts are directly extracted from the raw ICALENDAR files. The rest of the data, like the summary or the attached documents, is stored only to be presented to the user in reminders.

We call *relative temporal distance* the distance between the current date, and the starting date of an event or the due date of a task. Such a value is required to decide when to trigger a reminder. Discrete, multi-granular relative temporal distances are computed from ICALENDAR dates, durations and reccurrence rules. To be relevant for the learning mechanism of our agent, the raw distance is approximated and translated into a symbolic value expressing how close the current date is to the deadline or starting date. With such a decomposition, a calendar item is easily categorized as happening soon or in a long time.

Keeping track of the granularity is necessary, in order to consider relative symbolic values like *very close* or *far* in an appropriate time context: an event that starts in five minutes is obviously close to happen, but the due date of a task that is expected to take 8 weeks to be achieved and is only 5% completed can also be considered as *close* in a *weeks* time scale if the current date is less than 9 weeks before the deadline. For clarity and efficiency purposes, we chose a set of significant thresholds in order to categorize both the granularity (from *very short-term* to *long-term*) and the approximate value of a distance (from *very close* to *very far*, or *late*[1]), instead of proposing functions for computing this categorization. More details on the processing of temporal distances can be found in (Richard & Yamada 2007).

Historical Attributes

For learning a generalized classification of calendar items, we assumed that the most useful data are the user-defined categories assigned to an event or a task. Therefore, in order to learn about the user habits concerning a kind of activity, we propose to use additional attributes related to the categories of past events and tasks: the average duration and the average delay observed for a given category, plus a flag indicating whether the user is generally early when achieving this kind of task or attending this kind of event[2].

On the other hand, notifications can be repeated only if the user explicitly asks to be reminded later. As a frequent repetition of the same reminder will probably annoy the user,

[1] This additional distance value can be used for each granularity, in order to express the severity of the delay since the starting date or the due date occurred.

[2] If this is the case, the category delay represents how long before the deadline this kind of task is usually achieved.

information about previous reminders enables the system to learn when to present again the same notification content. We thus also store the following historical values for each item: the number of reminders already triggered, and the temporal distance since the last reminder. The distance to the previous reminder, as well as the average duration and delay for a category, have the same value domain as the granularity of temporal distances.

User Context

The user status is processed by the second Classifier System. It is the agregation of the following information:

- *busy state*: available, busy, very busy, away, signed-off
- *mood*: very good, good, average, bad, very bad
- *activity*: work, leisure, vacation, commuting, sick, conference, meeting, *etc.*
- *location*: office, transportation, home, business trip, *etc.*

The domains of the location and the activity attributes can be defined and extended by the user, while the busy state and the mood have a fixed set of values. Presently, the user status is gathered directly through the GUI. Nevertheless, it would be more convenient and less distractive to detect the user activity load automatically, through monitoring (keyboard/mouse activity, estimated posture, biosensors, *etc.*).

From the number and the priority of current tasks, extracted from iCalendar data, we also compute the current *task load* of the user. This value completes the information about the busy state, which might not be accurate or updated enough.

Device Context

The execution context of TamaCoach is also handled by CS2. It is composed of the following attributes:

- *device* : desktop, laptop, PDA, *etc.*
- *has network?* : does the device have an active network connection?
- *has sound?* : can the device play a sound?

Such information is acquired from the operating system of the hosting device. It is required to adapt the selected actions to the execution environment: for instance, if the device has no network connection, the reminding system cannot send an e-mail.

Getting Feedback from the User

As stated by S. Shiaffino (Shiaffino & Amandi 2003), the principal source of learning for an interface agent is user feedback. Such feedback can be either explicit (information is given on purpose by the user to improve the agent behaviour) or implicit (the agent observes user actions). In our case, because users may not interact very often with their calendar and todo managers, the design of the feedback mechanism is particularly crucial. Moreover, our agent has no means to detect the user attitude through facial expression recognition or physiological parameters, as seen in (Bartneck 2002; Prendinger & Ishizuka 2005).

As described in the general introduction of this paper, TamaCoach learns only from both direct and indirect interactions with its user, respectively through the dedicated GUI and by the successive modifications of the calendar data *via* an external application.

Reminders

A reminder can be presented under combinable forms, depending on the context. Presently, only a few forms are available (pop-up window, e-mail, mobile e-mail, or sound). However, other kinds of notifications could be added after the learning system has proven to be effective. Figure 4 shows an example of pop-up task reminder.

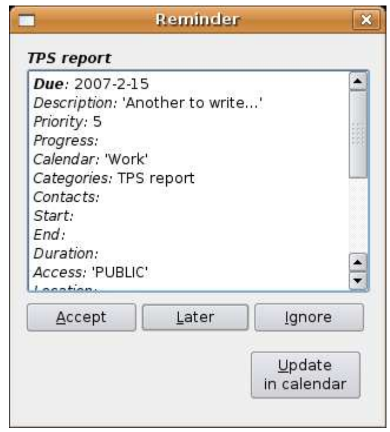

Figure 4: A task reminder with four predefined replies.

Whatever form is selected by TamaCoach, the new reminder is also appended to the appropriate list of pending events or pending tasks. If the lists are displayed in the TamaCoach GUI, as in figure 5, selecting an item in a list will make the corresponding non-modal pop-up reminder to appear.

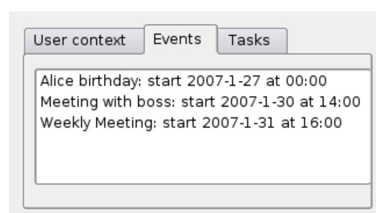

Figure 5: The list of events within the main panel.

Together with a reminder, a set of possible replies is presented to the user. A notification will stay in the list until the user directly answers back, by e-mail or through the GUI. Any user reply causes the reminder to be deleted from the according list of pending tasks or events.

Direct User Feedback

Together with a reminder, a set of six possible replies is presented to the user. This light-weight, explicit feedback system enables TAMACOACH to evaluate the quality of its actions, without disturbing busy users too much. As shown in figure 4, the main three possible replies are as follow:

- *Accept*: the user thinks that the reminder is useful and will act immediately, *e.g.* finish the task or attend the meeting.
- *Later*: the reminder has to be presented again later.
- *Ignore*: the user does not care about the reminder.

In the case of a *later* or *ignore* reply, the user can specify whether the reminder was respectively *too early* or *too late*. This distinction is important to avoid taking a neutral feedback as a punishment:

- *Later*: the user may have found the reminder useful but would like to be notified again anyway, *e.g.* if he/she cannot perform the task immediately. It does not necessarily mean that the reminder was issued too early.
- *Ignore*: the user informs the agent about his/her lack of interest in that particular task or event. It does not necessarily mean that the reminder arrived too late.

All the above replies concern the reminder itself: they give direct feedback on the usefulness of TAMACOACH actions. Nevertheless, we are dealing with ICALENDAR files that are modified by an external calendaring application. When notified, the user may notice that he/she has not updated the calendar for the reminded task or event, *e.g.* that a report was completed, a meeting was postponed, or a dinner was cancelled. We thus added an *update* reply, which indicates that the user will modify the calendar data in order to reflect the actual situation. Again, this situation should not be considered as a punishment by the learning system, and could even been used to get further information about the user habits. Note that TAMACOACH could ideally detect such a situation and cancel the reminder if possible (*e.g.* make the pop-up window disappear).

The items for which the user has replied *accept*, *ignore* or *too late* are considered as *acknowledged*, and will not be processed by the learning system anymore. A *later*, *too early* or *update* reply might trigger a new reminder later.

The main advantage of having a limited number of predefined replies is that the user can quickly understand the meaning and the consequences of each action. The acquired routine of answering to a reminder should reduce both the interruption length and the cognitive overload when selecting an appropriate reply. A conversational or an adaptive interface would certainly be more disruptive and require more attention from the user.

Remote and Delayed User Feedback

The user can interact with a running instance of TAMACOACH *via* e-mails sent from a computer, a PDA or a mobile phone. For now, this kind of remote interaction is possible only when replying to an e-mail reminder. However, remote control of the agent could be extended to other operations available through the GUI.

The user may not reply immediately to a reminder for various reasons: for instance, he/she forgot to update the user status before leaving the office, did not check his/her mobile e-mails, or is too busy to reply by e-mail or through the GUI. Delays in the user reply lead to delays in the feedback for the triggering rules. However, a delayed reply should not be considered as a negative feedback, as we cannot know why the user did not react immediately. We thus just consider that, in case of a delay, the next reminders will not be selected by an updated set of rules.

Simulating and Expressing Emotions

As stated in (Hassenzahl *et al.* 2000), the so-called *fun factor* is important to pull humans in using artefacts. Associating an expressive character to our reminding system might motivate the user in honouring various self-assigned commitments, by being both engaging and as least distractive as possible. We consider two compatible uses of basic emotional display:

- Informing the user: the expressed emotions should reflect the *satisfaction level* of the agent, in order to facilitate the interaction with the user. We believe that displaying emotions is a natural and efficient way to inform the user about the estimated satisfaction level of the agent, *i.e.* the global evaluation of its usefulness.
- Influencing the user: the expressed emotions might increase the involvement of the user and thus incite him/her in taking appropriate actions.

A basic reminding system does not require an elaborate relational agent to make the user feel comfortable. Creating a strong relationship with the agent could even be counterproductive, as it would distract the user. We thus believe that a non-cognitive agent, with limited learning and expressivity, should be sufficient for our purpose.

An Emotional System for TamaCoach

Our emotional system is inspired from L. Sarmento's architecture (Sarmento 2004). The agent emotional state is a set of *emotional accumulators*, associated with specific evaluation functions, appraisal thresholds and decay values. The *global mood*, which is a long-lasting state, is calculated from the values of emotional variables, which represent short-lived phenomenons. As shown in figure 1, the mood is used both as a reward value for the learning system, and to display the emotional state of the agent through its graphical representation.

As suggested in (Spinola de Freitas, Gudwin, & Queiroz 2005), the choice of an appropriate set of emotions depends on the application. Facial expressions should reflect the emotional state of the agent, in a simple but effective way, and must be easily understood, so no distraction or over-expectation can arise from the user. We chose to focus on a few basic emotions to be "felt" and expressed by the agent: joy, sadness, boredom, worry and surprise. Expressing *happiness* after several rewarding user actions should efficiently inform the user that the given responses improve the learning system. On the opposite, *sadness* indicates that the learning system has received many successive negative feedbacks, and thus evaluates itself as acting poorly. *Boredom* and *worry* both indicate that the user does not interact enough with the application: TAMACOACH gets bored when there are few events and tasks to process, and starts to worry when too many pending items are accumulating. Finally, *surprise* may arise when the user reply to the notification was unexpected.

The initialization of both the learning and the emotional systems strongly depends on the personality of the user. A user profile could be gathered through an initial questionnaire, in order to accelerate the learning process and reduce the risk of inadequation of the agent personality. Moreover, the user should be able to give more precise feedback whenever the mood expressed by the agent seems inconsistent with the actual satisfaction of the user. For instance, if the agent looks sad while the user actually finds the tool useful, the user could adjust the emotional system parameters through a short questionnaire.

Why Expressing Emotions?

As previously discussed, the influence of relational, human-like agents is rather controversial. In our case, the agent is just a notification system, which learns to interrupt the user only when relevant. We think that a conversational agent would be unnecessary, and even distractive. Moreover, letting the user feel like being monitored and judged by an anthropomorphic agent about his/her organizational habits might be counter-productive. Therefore, inspired by the famous TAMAGOCHI game, we chose to represent the agent as a virtual creature, which simulates and demonstrates basic emotions.

A player does not expect his/her TAMAGOCHI to be smart, but the expressed, basic needs and emotions are sufficient to encourage the human to feed, clean or play with the artificial creature. As with a TAMAGOCHI, we hope that inducing emotions like amusement or guilt will lead the user in interacting with the reminding system and the calendar application. A limited social bond may even arise, if the user comes to trust TAMACOACH in effectively reminding items, or enjoys interacting with it. We believe that this should naturally incite the user in taking appropriate actions to fulfil self-assigned commitments, without being too distractive.

Conclusion

In this paper, we have presented our adaptive, emotional and expressive reminding system. We have described our learning architecture, and defined the requirements in terms of contextual input data, to be extracted from raw data stored in ICALENDAR files, or obtained from the user and the host device. We also have described how to get useful but lightweight feedback from the user, and how to extend the modular agent architecture with an emotional system. Finally, we have discussed the role of expressed emotions within this specific case of human-agent interaction: as we ultimately aim at developing a virtual coaching system applied to personal time management, we expect that the display of emotions on a simple, artificial creature would at least push the user in interacting with its adaptive reminder and thus organize personal time more efficiently.

However, a coaching system should guide the user along the consecutive steps required to achieve long-term tasks. This is not yet possible with the scarce information provided by ICALENDAR files and the actual TAMACOACH system: we need to gather more information from the user about tasks, events and contacts, take more historical data into account, and move towards project planning assistance.

We are developing a first prototype, incrementally tested on three different platforms: a desktop PC (GNU/Linux), a laptop (MS-Windows), and a PDA (Zaurus C-3000 under GNU/Linux). In parallel, we are investigating how to simulate various user profiles for evaluating and tuning the learning system. Then, we will integrate the described emotional system to display facial expressions. Afterwards, we will conduct experiments on human users, in order to validate our hypothesis about the possible influence of an adaptive, emotional time manager on different kinds of users. If the adaptive and expressive parts of our reminding system appear to be useful and accepted by the users, we will be able to extend it with guidance abilities: TAMACOACH would then accompany its user through an introspection and self-improvement process, and fulfil its role as a virtual coach for personal time management.

Acknowledgements

This research project is founded by the Japan Society for the Promotion of Science (JSPS). We would like to thank the anonymous reviewers for their most valuable comments, as well as Sébastien Duval (NII) for his constructive remarks.

References

Bartneck, C. 2002. *eMuu An Embodied Emotional Character for the Ambient Intelligent Home*. Ph.D. Dissertation, Technische Universiteit Eindhoven.

Bellotti, V.; Ducheneaut, N.; Howard, M.; and Smith, I. 2003. Taking Email to Task: The Design and Evaluation of a Task Management Centered Email Tool. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. Fort Lauderdale, USA.

Bellotti, V.; Dalal, B.; Good, N.; Flynn, P.; Bobrow, D. G.; and Ducheneaut, N. 2004. What a To-Do: Studies of Task Management Towards the Design of a Personal Task List Manager. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. Vienna, Austria.

Berry, P. M.; Gervasio, M. T.; Uribe, T. E.; and Yorke-Smith, N. 2005. Mixed-Initiative Issues for a Personalized Time Management Assistant. In *Proceedings of the Workshop on Mixed-Initiative Planning and Scheduling*.

Berry, P.; Conley, K.; Gervasio, M.; Peintner, B.; Uribe, T.; and Yorke-Smith, N. 2006. Deploying a Personalized Time Management Assistant. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Hakodate, Japan.

Bickmore, T., and Mauer, D. 2006. Modalities for Building Relationships with Handheld Computer Agents. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. Montréal, Canada.

Butz, M.; Goldberg, D.; and Lanzi, P. L. 2003. Gradient Descent Methods in Learning Classifier Systems: Improving XCS Performance in Multistep Problems. Research report, IlliGAL (Illinois Genetic Algorithms Laboratory), University of Illinois at Urbana-Champaign.

Creed, C., and Beale, R. 2006. Evaluating Affective Embodied Agents Over Extended Interactions. In *Proceedings of the First Workshop on Emotion and Computing – Current Research and Future Impact*. Bremen, Germany.

Cutrell, E.; Czerwinski, M.; and Horvitz, E. 2001. Notification, disruption, and memory: Effects of messaging interruptions on memory and performance. In *Proceedings of the International Conference on Human-Computer Interaction (INTERACT)*. Tokyo, Japan.

Czerwinski, M.; Horvitz, E.; and Wilhite, S. 2004. A Diary Study of Task Switching and Interruptions. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. Vienna, Austria.

Dehn, D. M., and van Mulken, S. 2000. The Impact of Animated Interface Agents: a Review of Empirical Research. *International Journal of Human-Computing Studies* 52(1).

Hassenzahl, M.; Platz, A.; Burmester, M.; and Lehner, K. 2000. Hedonic and ergonomic quality aspects determine a software's appeal. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. The Hague, The Netherlands.

Horvitz, E. 2007. Machine Learning, Reasoning, and Intelligence in Daily Life: Directions and Challenges. In *Proceedings of the Workshop on AI Techniques for Ambient Intelligence (AITAmI)*. Hyderabad, India.

Isbister, K., and Doyle, P. 2002. Design and evaluation of embodied conversational agents: A proposed taxonomy. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Bologna, Italy.

McQuiggan, S. W., and Lester, J. C. 2006. Learning Empathy: A Data-Driven Framework for Modeling Empathetic Companion Agents. In *Proceedings of the International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*. Hakodate, Japan.

Middleton, S. 2001. Interface agents: A review of the field. Technical report, University of Southampton.

Moldt, D., and von Scheve, C. 2001. Emotions and Multimodal Interface-Agents: A Sociological View. Technical report, University of Hamburg.

Myers, K.; Berry, P.; Blythe, J.; Conley, K.; Gervasio, M.; McGuiness, D.; Morley, D.; Pfeffer, A.; Pollack, M.; and Tambe, M. 2006. An Intelligent Personal Assistant for Task and Time Management. *AI Magazine*.

Nabeth, T.; Razmerita, L.; Angehrn, A.; and Roda, C. 2005. InCA: a Cognitive Multi-Agents Architecture for Designing Intelligent & Adaptive Learning Systems. *Special issue of the ComSIS Journal* 3(2).

Pollack, M. E.; Brown, L.; Colbry, D.; McCarthy, C. E.; Orosz, C.; Peintner, B.; Ramakrishnan, S.; and Tsamardinos, I. 2003. Autominder: An Intelligent Cognitive Orthotic System for People with Memory Impairment. *Robotics and Autonomous Systems* 44(3-4).

Prendinger, H., and Ishizuka, M. 2005. The empathic companion: A character-based interface that addresses users' affective states. *International Journal of Applied Artificial Intelligence* 19(3-4).

Richard, N., and Yamada, S. 2007. Context-awareness and User Feedback for an Adaptive Reminding System. In *Proceedings of the Workshop on AI Techniques for Ambient Intelligence (AITAmI)*. Hyderabad, India.

Rickenberg, R., and Reeves, B. 2000. The effects of animated characters on anxiety, task performance, and evaluations of user interfaces. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI)*. The Hague, The Netherlands.

Sarmento, L. M. 2004. *An Emotion-Based Agent Architecture*. Ph.D. Dissertation, Faculdade de Ciências da Universidade do Porto.

Shiaffino, S., and Amandi, A. 2002. On the design on a software secretary. In *Proceedings of the Argentina Symposium on Artificial Intelligence (ASAI)*. Sante Fe, Argentina.

Shiaffino, S., and Amandi, A. 2003. User-interface agent interaction: personalization issues. *International Journal of Human-Computer Studies* 60(1):129–148.

Spinola de Freitas, J.; Gudwin, R. R.; and Queiroz, J. 2005. Emotion in Artificial Intelligence and Artificial Life Research: Facing Problems. Technical report, Electrical and Computer Engineering, State University of Campinas.

Weber, J., and Pollack, M. 2005. Simulating Users to Support the Design of Activity Management Systems. In *Proceedings of the Winter Simulation Conference*. Orlando, USA.

Wilson, S. W. 1995. Classifier Fitness Based on Accuracy. *Evolutionary Computation* 3(2):149–175.