An Assistive Handwashing System with Emotional Intelligence

Using Emotional Intelligence in Cognitive Intelligent Assistant Systems

by

Luyuan Lin

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Master of Science
in
Artificial Intelligence

Waterloo, Ontario, Canada, 2014

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

Whether emotional intelligence (or, affective reasoning) is included can influence the effectiveness of a cognitive assistive system. This thesis presents a novel emotionally intelligent hand-washing assistant that aims to help older adults with Alzheimer's disease complete hand-washing tasks more independently. The thesis reviews previous works in the development of cognitive assistants and in the study of emotional intelligence, and then designs a hand-washing system prototype that combines the two research streams. The difficulties in designing such a system, including probabilistic and decision-theoretic reasoning of the user's functional and emotional states, computer-vision based activity monitoring and affective recognition, and embodied prompting are discussed. Designing the hand-washing system as an integration of independent components, the thesis also discusses coordination between the components. The thesis implements the system in the end, and shows by preliminary tests in laboratory settings that the system implemented (1) runs in real-time from the perspective of the user group, (2) is able to provide a level of functional assistance, (3) produces system prompts that have encoded to some extent the emotional state of its user. The preliminary tests also indicated that a user with emotions with high potency levels (and high activity levels) is more likely to receive system prompts with low potency levels (and high activity levels).

This thesis is one of the exploratory works in the area of integrating emotional intelligence with cognitive intelligent assistive systems. It provides a solution to fitting emotional intelligence in a functional system, as well as points out directions for future improvements. The framework designed in this thesis is portable and extensible, and can be generalized to be used in other applications.

Keywords. Affective reasoning, emotional intelligence, Affect Control Theory, BayesACT, handwashing assistive system, affect recognition, affect signal generation, cognitive intelligent assistant.

Acknowledgements

I would like to take this opportunity to thank all the people who made this thesis possible.

Foremost, I would like to express my sincere gratitude to my supervisor Prof. Jesse Hoey for the continuous support to my master's study and research. His guidance helped me in all the time of research and writing of this thesis.

I would like to thank the rest of my thesis committee: Prof. James Tung and Prof. Peter van Beek, for their insightful comments and questions.

I would also like to thank my friends and fellow labmates in the Health Informatics Group: Chengbo Li and Xiao Yang, and my boyfriend Enxun Wei, for the stimulating discussions, valuable advice in writing this thesis, and for all the fun we have had together. Additional thanks goes to my boyfriend who accompanied me for the sleepless nights writing this thesis.

Last but not the least, I would like to thank my parents, my little sister, and all my other friends, for their constant encouragement.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

According to the United Nation's population report 2013 [1], we are experiencing an aging world. The global share of older people (aged 60 years or over) increased from 9.2% in 1990 to 11.7% in 2013 and will continue to grow as a proportion of the world population, reaching 21.1% by 2050. Currently around the world, there are about 40% of older persons aged 60 years or over who live independently, that is to say, alone or with their spouse only. As countries develop and their populations continue to age, living alone or with a spouse only will likely become much more common among older people in the future. While many elders remain healthy and productive, overall this segment of the population is subject to physical and cognitive impairment at higher rates than younger people. Take a broad category of brain diseases, dementia, for example. While only 3% of people between the ages of 65 to 74 have dementia, 47% of people over the age of 85 have some form of dementia [9]. As more people are living longer, dementia is becoming more common. Dementia can cause long term loss of ability to think and reason clearly. Persons with dementia (e.g. Alzheimer's disease) are reported to have difficulty in daily functioning, such as hand-washing, preparing food and dressing. For example, while performing a task in daily life, persons with Alzheimer's disease (AD) may forget how much of the task he or she has completed, what an object looks like, or what the necessary steps are.

Luckily, more and more new technologies that incorporate Artificial Intelligence (AI) methods have been explored to build assistive systems that can help with an elder's everyday lives. Smart home systems are being developed to help older adults with AD in a variety of ways, for example, in automated reminders for tasks like handwashing [23]

[1] http://www.un.org/en/development/desa/population/publications/pdf/ageing/WorldPopulationAgeing2013.pdf

and meal preparing [52], and providing social and cognitive stimulation [30]. The assistive systems designed for the many and varied difficulties faced by older adults with cognitive disabilities typically take the form of automated methods of monitoring the users' behaviours [26], assessing the users' cognitive levels, and assisting the users to complete daily tasks by providing prompts when necessary [7, 51, 23]. However, even when the solutions satisfy functional requirements, the systems' effectiveness may still be limited by a lack of an affective (emotional) connection between the systems and their users.

A COACH system has been developed to assist older adults with dementia to carry out basic daily activities (e.g. hand-washing) [7, 40]. The system is effective at monitoring a user washing his/her hands, detecting when the user has lost track of what he/she is doing, and when needed, displaying a prerecorded assistive prompt [40]. However, while the system works well for some persons, it does not perform as well for others. One primary reason for this may be the limitation of the pre-recorded prompts in capturing the heterogeneity in socio-cultural and personal affective identities of the users. Each user of the system comes from a different background and has a different sense of "self". Thus, the users would have different emotional responses to the prompts given. For example, one person may find the prompt helpful and motivational, while another may find it imperious and impatient, and prefer a more servile instructional message. The disapprovement feelings caused by the prompts may affect a user's responsiveness. Apparently, the "one size fits all" style of prompting is not enough for an assistive system aiming at high cross-user performance. More sophisticated methods of generating prompts that align with the user's affective states (such as identities) are needed.

Affect Control Theory (ACT) [57] is a well established sociological theory that models affective reasoning during human interactions. It represents all affective meanings, such as those of identities and behaviours, by three-dimensional EPA vectors: evaluation (E, e.g. how positive), potency (P, e.g. how powerful), and activity (A, e.g. how active). The theory hypothesizes that people have fundamental sentiments about their identities and will act to minimize the deflection of transient impressions caused by social events from the fundamentals. ACT hypothesizes that the fundamental sentiments of identities and behaviours are shared within a same culture. These hypotheses have been supported by a variety of studies. Tests of ACT's validity on both verbal (e.g. INTERACT[2]) and non-verbal behaviours [61] have been reported. Based on ACT, an interactant's behaviours can be predicted given his/her affective state (i.e. his/her affective identity[3] and how he/she perceives the situation emotionally).

[2] Program accessible via http://www.indiana.edu/~socpsy/ACT/interact.htm. Readers are refered to http://www.indiana.edu/~socpsy/ACT/references.html for more readings on ACT.

[3] In this thesis, the term *identity* is used to denote a kind of person in a social situation.

Identities of persons with AD are not easy to obtain. Studies have shown that the identities of persons with dementia are changed by the disease [46], and that persons with AD have more vague or abstract notions of their identity [58]. To tackle this problem, a BayesACT model to learn interactants' identities has been formulated on the basis of the original ACT [24]. Based on the BayesACT model, the authors of [24] have built a program that, in simulation, can assist persons with AD to complete the handwashing task. Though the program is able to provide functional prompts while reasoning about the users' identities and emotions, it consumes pre-processed observation information (including user behaviour labels and the affective meaning as EPA vectors of the behaviours) as input, as opposed to perceiving the information from the environment itself. The program is not able to construct and display real prompts to its users; instead, it represents the prompts to be displayed by functional and emotional labels, with the former one describing the instructional content that should be contained in the prompts and the latter one indicating the affective meanings (again, EPA vectors) desired in these prompts.

In this thesis, based on the COACH and the BayesACT approaches, we developed a prototype of an assistive system that monitors a person with AD during a handwashing process, learns about the affective identity of the person, and provides prompts that both instruct what the person should perform next to complete the handwashing task, and simultaneously correspond with the person's emotional states. The system uses an RGB-D camera to track the user's hands while hand-washing, and recognizes functional meanings (e.g. has the person turned the water on?) and affective meanings (e.g. is the person active and feeling powerful?) of the user's behaviours. The detected functional and affective meanings of the user behaviours are then fed into a reasoning engine where the system's belief states, including how much the user has completed in the handwashing task and what the the user's affective identity is, are updated. The reasoning engine uses a partially observable Markov decision process (POMDP) to update the system's belief states, and the affective component is based on ACT. The POMDP policy of the reasoning engine then produces an approximately optimal action for the system to take, with the actions described with both functional (e.g. the instructional content of the prompt) and affective (e.g. how the content should be expressed) meanings. A most appropriate audio-visual prompt is finally selected from a set of pre-generated prompts with both functional labels and affective ratings. The final prompt is chosen in a way that both of its functional label and emotional rating are consistent with the functional and emotional meanings of the desired prompt recommended by the reasoning engine.

The goal of this thesis is to show that it is functionally possible to integrate the emotional reasoning of ACT with an existing cognitive assistive technology, the COACH. It focuses on the integration work of fitting emotional intelligence in a functional system, while pointing out directions for future improvements. The contribution of this thesis is therefore to demonstrate, in a controlled laboratory setting with human actors only, how emotional reasoning, a key missing component of most assistive systems, can be integrated into a cognitive intelligent assistive system. The objectives of this thesis are as follows:

Objectives. To augment the COACH system with an emotional reasoning engine based on BayesACT so that the augmented system: (1) is designed in a portable and extensible way; (2) runs in real-time from the perspective of the user group; (3) provides at least a level of functional assistance of as high quality as the COACH; (4) is able to tune the prompts in some way according to the emotional state of a user. The last objective (4) is ill-defined, as the question of how exactly tuning prompts to users will be most effective is not clear at this point.

The thesis is structured as follows. Basic concepts used throughout the thesis are defined in Chapter 2. Related previous studies, including approaches in building high performance assistive systems, are reviewed in that chapter as well. With the importance of including emotional intelligence in the design of human-computer interaction (HCI) discussed, Chapter 2 then briefly examines previous works in the topic of emotional intelligence, i.e. previous research in the areas of *recognition of affective states, generation of affectively modulated signals, psychological study of human emotions, and computationally modeling affective HCIs.* Chapter 3 of the thesis discusses the challenges in designing an assistive handwashing system that has combined all the aforementioned aspects of emotional intelligence altogether. The whole system is divided into independent components, and both general analysis and detailed examinations of input and output requirements and design difficulties of each of the components are provided. Communication between the components is discussed in the chapter as well. Finally, Chapter 3 explains the design of the system as an integration of independent components, among which some components are designed as extensions to existing programs. Chapter 4 describes how the system is implemented in detail. Experimental results are presented in Chapter 5. Chapter 6 wraps up the thesis by discussing the contributions of this thesis and possible future works.

Chapter 2

Background

2.1 Basic Concepts

2.1.1 Affective Computing and Affect Control Theory

Affective Computing refers to the study of developing machines (or computer programs) capable of recognizing, interpreting, processing and generating human affect. It is an interdisciplinary field spanning computer science, psychology, and cognitive science, and is believed to be an important topic for "harmonious human-computer interaction" [64]. One aspect that Artificial Intelligence (AI) researchers have recently been concerned about is to have machines interact with humans emotionally. That is to say, machines should interpret the emotional state of humans and adapt its behaviours to them, i.e. to give appropriate responses for those emotions. With the ability to process affective information, machines are believed to exhibit higher flexibility, and to work in uncertain or complex environments [53].

Affective computing research is based on theories of emotion [36]. Among all the popular emotions theories proposed, two main representations of emotional or affective states are used: categorical labels, and dimensional models. Based on their linguistic use in daily life, categorical labels can be used to describe affective states. Different set of labels can be chosen depending on the study. Most frequently, the following labels are used to describe affective states: anger, happiness, sadness, surprise, disgust and fear [18]. Dimensional models represent affective states as vectors containing a set of independent dimensions. The value for each of the dimensions can be real numbers. A common set of dimensions used to capture emotional experiences consists of three factors [60]. The first one is variously called friendliness-hostility, pleasure, or valence, describing a general positive versus negative evaluation of the emotional experience. The second is alternatively called dominance-submissiveness, control, power, or potency. It captures the amount of control over others and the surroundings versus feeling controlled by external circumstances. The third one is interchangeably called activation, arousal, or activity. Activity level corresponds to the level of activation, mental alertness, and physical activity. A fourth dimension relating to the "uncertainty" of situations is also proposed by some research [20]. Compared with categorical labels, dimensional representations may relate more to the underlying physiological changes [39].

One well known sociological theory that describes emotions as three-dimensional vectors that represent evaluation (E), potency (P), and activity (A) respectively is Affect Control Theory (ACT) [57]. Different from categorical labels, the EPA-vector representation describes affective meanings of concepts in precise, measurable ways[1]. By using the same three affective scales, programs implemented based on ACT are able to track the affective meanings of actors and behaviours in an interaction.

ACT describes social events by an Actor-Behaviour-Object (ABO) grammar: *Actor Behaves* towards *Object*, where Object is usually another actor (e.g. a human). Each of the ABO elements is associated with an EPA vector. The EPA values of the interactants' identities, behaviours, and environmental settings are referred to as fundamental sentiments in ACT. ACT hypothesises that the fundamentals of identities, behaviours, etc, are shared between people within a same culture. On the other hand, the emotional feelings of people evoked by a specific event are referred as transient impressions, and can be measured by in-context ratings of the ABO elements. ACT proposes that people behave in interactions to minimize the deflection of transient impressions to fundamental sentiments. This proposition is referred to as the Affect Control Principle defined below.

Definition 1 *The Affect Control Principle [57]: Actors work to experience transient impressions that are consistent with their fundamental sentiments.*

The ACT hypothesis that fundamentals are shared within the same culture is supported with a large variety of studies. Tests of ACT's validity on both verbal (e.g. INTERACT[2]) and non-verbal behaviours [61] have been reported. EPA profiles of concepts, including identities and behaviours, can be measured with the semantic differential, a survey technique where respondents rate affective meanings of concepts on numerical scales

[1] Each of E, P and A is a real number within range $[-4.3, 4.3]$

[2] Program accessible via http://www.indiana.edu/~socpsy/ACT/interact.htm

[48]. In general, within-cultural agreement about EPA meanings of social concepts is high even across subgroups of society, and cultural-average EPA ratings from as little as a few dozen survey participants are extremely stable over extended periods of time. For example, the EPA for the identity of "nurse" is $[1.65, 0.93, 0.34]$, meaning that nurses are seen as quite good, a bit powerful, and a bit active. Comparatively a "patient" is seen as $[0.9, -0.69, -1.05]$, less powerful and active than a "nurse".

With the hypothesis that people within same culture share the same expectations, or fundamental sentiments, for each identity and action, ACT proposes that the two communicating parties sharing the same cultural background would act to minimize deflections from fundamental sentiments during interactions. If a large deflection is caused for some reason, they choose actions that can restore the impression. ACT implies that the two communicating parties, knowing or having beliefs about their identities, would not only have expectations on what they should perform in the interaction, but would also have expectations on what the other party should perform. For example, in a situation where a student is asking a tutor questions and both of them are aware of their identities, the student is supposed to be polite, less powerful and more active, while the tutor is supposed to be patient, more powerful and less active. The interaction would go smoothly if the tutor performs these expected actions, e.g. answers the student's questions patiently. However, if the tutor suddenly yells at the student criticizing him/her for being stupid, a large deflection would be caused. The student's focus would shift from solving his/her previous problems; he/she would start to figure out why he/she was yelled and what he/she should perform to have the tutor become nicer.

ACT can serve as a general psychological principle of micro-regulation of interpersonal interactions. By presenting all affective meanings as three-dimensional vectors, i.e. the EPA vectors, it enables mathematical computations on past sentiment interactions and thus presents a maximum likelihood solution predicting optimal behaviours or identities.

2.1.2 POMDP

A partially observable Markov decision process (POMDP) is a general stochastic model that has been extensively studied in operations research and in artificial intelligence [42, 55]. Figure 2.1 shows a time slice of a general POMDP (solid lines). Capital symbols (e.g. X) are used to denote variables or features, small symbols (e.g. x) are to denote values of these variables, and boldface symbols (e.g. $\mathbf{X}$) are to denote sets of variable values. Primes are used to denote post-action variables, so x' means the value of the variable X after a single time step. As shown in the figure, a POMDP consists of a finite set $\mathbf{X}$ of states

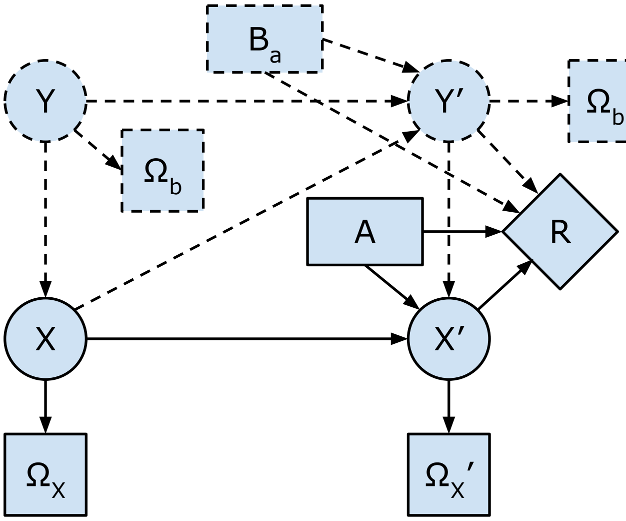


Figure 2.1: A time slice of a general POMDP (solid lines) and a POMDP augmented with affective states (dotted lines)

X; a finite set $\mathbf{A}$ of actions A; a stochastic transition model $Pr : X \times A \rightarrow \Delta(\mathbf{X})$, where $Pr(x'|x, a)$ denotes the probability of moving from state x to x' after an action a is taken, and $\Delta(\mathbf{X})$ is a distribution over $\mathbf{X}$, a finite observation set $\mathbf{\Omega_X}$, and a reward assigning function $R(A, X')$. The reward function $R(a, x')$ denotes the reward received after taking action a and transitioning to state x'. A stochastic observation model $Pr : X \rightarrow \Delta(\mathbf{\Omega_X})$ is used to denote the probability of making observation ω while the system is in state x. Basing on the aforementioned elements, a policy can be developed to map belief states (i.e. distributions over X) into choices of actions, such that the expected discounted sum of rewards is (approximately) maximized. POMDPs have been used as models for many human-interactive domains, including human assistance systems [23].

Figure 2.1 (dotted lines) shows a time slice of a general POMDP augmented with affective states. In addition to the basic POMDP elements, affective states $\mathbf{Y}$ are included in the POMDP process. $\mathbf{Y}$ describes the system's beliefs of the user's emotional states. Similarly to Ω_X and A, Ω_b denotes observations of user behaviours that gives the system evidence about state Y, and B_a is the affective meaning of system action that can cause state Y to change. Finally, the reward function $R(A, X', Y')$ is defined over state-action pairs and rewards those states and actions that are beneficial overall to the goals of the system-human interaction.

2.1.3 BayesACT

ACT models interactions between two persons with a prerequisite that the identities of the two communicators are known to each other. This prerequisite has limited its usefulness in our application, where the user's identity is unpredictable. On the other hand, a Bayesian version of the ACT theory, called BayesACT, was formulated in Hoey et al.'s work [24]. This new model can maintain multiple hypotheses about identities and behaviours simultaneously as a probability distribution, and can make value-directed action choices. By employing BayesACT, machines are able to generate affectively believable interactions with people by learning about their identity, predicting their behaviours, and taking actions that are simultaneously goal-directed and affect-sensitive.

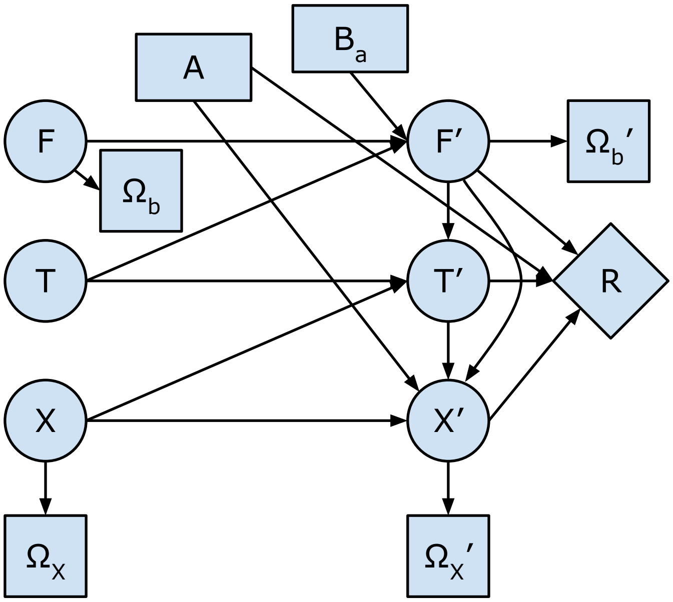


Figure 2.2: A factored POMDP for Bayesian Affect Control Theory

Figure 2.2 shows a factored POMDP for the BayesACT. It describes, from the perspective of the agent (although this is symmetric), how the variable state changes based on an interaction between an agent (i.e. a machine) and a client (i.e. a human). In our description of the figure, capital symbols (e.g. F, T) denote variables or features. Small symbols (e.g. f, t) denote values of these variables, and boldface symbols (e.g. $\mathbf{B_a}$) denote sets of variable values. Primes are used to denote post-action variables, so x' means the values of the variable X after a single time step.

In the figure, X is used to represent everything the system needs to know about the system state, for example, the functional state of the agent and the client. To indicate which party acts at the given time step, X encodes turn-taking messages as well. $F = \{F_{ij}\}$ denote the set of fundamental sentiments the agent holds, where each feature $F_{ij}, i \in \{a, b, c\}, j \in \{e, p, a\}$ denotes the j-th value of the i-th interaction object: a (actor), b (behaviour), or c (client). Similarly, $T = \{T_{ij}\}$ is defined and denotes the set of transient sentiments the agent holds. Variables F_{ij} and T_{ij} are continuous valued and F, T are each vectors in a continuous nine-dimensional space. Note that F and T are encoded as being for agent and client, regardless of who is currently acting in the BayesACT model. The observations Ω_X and Ω_b are anything the system observes in the environment that gives it evidence about the variables X and F, such as the actions the agent and the client have taken. The system action A denotes the propositional content of a system action (e.g. to instruct the user to perform a behaviour), and B_a denotes how the message should be expressed (e.g. with a friendly tone). Variable values $\mathbf{B_a}$ are three-dimension vectors, i.e. EPA vectors.

To sum up, the BayesACT model includes states $S = \{F, T, X\}$, observations $\Omega = \{\Omega_X, \Omega_b\}$, and actions $\{A, B_a\}$. Among the state symbols, $Y = \{F, T\}$ represents the emotional state of the client, and X represents the functional state of the client. A state S is a probability distribution over the emotional state Y and functional state X of the client. S is updated given the history of actions A and observations . By updating F, the probability distribution of the client's identity F_c is learned. BayesACT can also predict the client's next behaviour and calculate $\{A, B_a\}$ basing on this prediction given the current belief state of $\{F, T, X\}$. The following essential concepts and formulas are defined in BayesACT:

— The deflection between fundamental sentiments F and transient impressions T, denoted as $\phi(F, T)$, is defined as a nine-dimensional weighted Euclidean distance between F and T. The distance measure is proposed by the authors of [24] as the logarithm of a probabilistic potential:

$$\phi(f, t) \propto e^{-(f'-t')\Sigma^{-1}(f-t)} \tag{2.1}$$

where Σ is a general representation of weights.

— The probability of a post-action fundamental sentiments f' is computed by combining the deflection $\phi(f', t')$ with an "inertial" term that stabilizes the fundamentals over time. It gives the probabilistic generalization of the Affect Control Principle (Definition 1). This can be illustrated by the following formula:

$$Pr(f'|f, t, x, b_a, \phi) \propto e^{-\phi(f',t')-\xi(f',f,b_a,x)} \tag{2.2}$$

where t' can be computed from $\{f', t, x\}$ by empirically derived prediction equations of ACT. Equation 2.2 thus represents the temporal dynamics of f encoding both the stability of affective identities and the predictive dynamics of affective behaviours. ξ is such that: (1) $f'_b = b_a$ if the agent is acting, and unconstrained if otherwise; (2) f'_a and f'_c are likely to be close to f_a, f_c, respectively.

— $Pr(x'|x, f', t', a)$ is defined to denote how the application progresses given the previous state, the fundamental and transient sentiments, and the (propositional) action of the agent.

— $Pr(\omega_b|f)$ and $Pr(\omega_x|x)$ are observation functions for the client behaviour sentiment and system state, respectively. These functions are stochastic in general.

2.2 Related Work

2.2.1 Use Assistive Technologies to Help Elders

According to the United Nation's population report, we are experiencing an aging world. While many elders remain healthy and productive, overall this segment of the population is subject to physical and cognitive impairment at higher rates than younger people. More and more new technologies that incorporate AI methods have been explored to build assistive systems that can help with elder's everyday lives. Pollack [54] surveyed such systems, focusing on the ones that support older adults who are grappling with cognitive decline.

Assistive technology can assist older people with cognitive impairment in one or more of the following ways [54]: (1) Assurance systems (e.g. [26]) where the primary goal is to ensure safety and well-being of elders. A caregiver is alerted if the elder is detected not performing well; (2) Compensation systems (e.g. [7, 51, 23]) where the primary goal is to help the elder perform daily activities, i.e. monitoring and giving out prompts when necessary; (3) Assessment systems that aim to assess the elder's cognitive and physical status. While it is obvious that the ability to observe, recognize and reason about the elder's performance of daily activities is essential for assurance systems, this ability is equally important for both assistive systems and assessment systems: the more accurately an assistive system can recognize activities and estimate a user's current state and needs, the more useful assistance it can provide; the better an assessment system can recognize daily activities and reason about how and when a user performs these daily activities, the better assessment of the user's cognitive state it can provide.

Activity recognition is currently a very active research topic. Work has been done to use sensors to monitor the execution status of particular types of activities, such as handwashing [23], meal preparation [52], and movements around town [26]. In general, Bayesian networks are the principal technology used for performing activity recognition. A typical approach is taken in the PROACT system [52], which employs a dynamic Bayesian network that represents daily activities such as making tea, washing, brushing teeth, and so on. In their approach, radio frequency identification (RFID) tags are attached to household objects and the user of PROACT wears a specially designed glove that includes an RFID reader. Since special gloves are required in the PROACT system, this approach is somewhat inconvenient for users. Another assistive system example, the COACH system [7, 23], used a Bayesian network. In the COACH system, images are grabbed by a camera mounted above the sink, and fed into a hand and towel tracker. The tracker then processes these images and reports the positions of the hands and towel to a belief monitor that uses a POMDP framework and a Bayesian network to estimate where in the task the user is currently: what they have managed to do so far, what their internal mental state is, etc. The belief of the user's state is passed to a policy processor, where belief states are mapped into actions: based on the belief states, the system may play out audio-visual prompts, call for human assistance, or do nothing. The COACH system is more user-friendly than the PROACT system in the sense that it doesn't require the user to wear any specific devices.

As well as accurate activity recognition, high quality human-computer interaction (HCI) is also a desirable feature for assistive systems. In fact, as more and more intelligent objects (physical robots, programs, etc) are being developed, more and more research has been conducted in the field of HCI. Viewing HCIs as a social activity, Suchman reviewed how agents are currently configured and stated her view of how they might be reconfigured in her 2007 book [63]. As Suchman pointed out, the planning model was at that time (and probably still is) the dominant model for intelligent machines and rational action; however, the situated part has been neglected in such models.

Being a large subset of intelligent objects, systems that can provide guidance to humans are built to reason about observations towards certain objectives and act based on their reasoning. The effectiveness of these assistive systems not only relies on the accuracy of user-behaviour recognition and rational planning, but also depends on how well the user understands the instructions these systems give out. To achieve the goal of communicating purpose to users more effectively, actions of the system are required to encode more local situational factors, such as awarenesses and emotional states of the user at that time. The COACH system [23] took a step towards this direction by including variables that describe the user's state, such as awareness, responsiveness and overall dementia level. However, the user's emotion was not considered by the system when producing behaviour suggestions.

2.2.2 Including Emotional Intelligence in HCIs

It has been widely agreed that the essence of emotional intelligence should be included in the next generation of HCIs. Studies have shown that being capable of detecting and responding appropriately to its users' affective feedback can make a HCI system act more naturally and effectively. The topic of using emotional intelligence in HCI is mainly concerned with four main aspects: (1) *recognition of affective states*, example approaches include vision-based, acoustic-based, and modality approaches [49, 66]; (2) *generation of affectively modulated signals*, such as speech, facial and bodily expressions [11, 44]; (3) *psychological study of human emotions*, including affective interactions and adaptation [60]; (4) *computationally modelling affective human-computer interactions* [24, 56, 19, 14]. While research covering one or more of these topics have been conducted, few real-world applications that combine all of the four pieces together have been implemented. This thesis takes a look at each of these aspects and designs and builds an assistive system that integrates all four pieces together and harnesses the benefits of including emotional intelligence in HCI.

Affective States Recognition

A large body of psychological studies have been conducted on examining how factors influence human emotions and how these emotions can be measured. While there is no single gold-standard method for measurement of one's emotions, it is widely agreed that emotions consist of variably interrelated changes in activity across a set of five components [59]: (1) appraisals of event, (2) psychophysiological changes (bodily sensation), (3) motor expressions (face, voice, gestures), (4) action tendencies, and (5) subjective experiences (feelings). Table 2.1, borrowed from [59], gives some examples of values for these factors.

With all these indicators of emotional changes, in the context of automatically recognizing affective states using computers, approaches analysing both verbal and nonverbal behaviours have been conducted.

Studies concerned with "verbal behaviours", such as words selected in an interaction, is probably one of the most mature ones in the domain of sentiment analysis[3]. Several dictionaries mapping words into affective meanings have been constructed by human raters in survey-based studies and have been released for public accessibility online[4]. Several

[3] Sentiment analysis (also known as opinion mining) refers to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information in source materials. See http://en.wikipedia.org/wiki/Sentiment_analysis.

[4] See http://www.indiana.edu/~socpsy/ACT/data.html for a list of the dictionaries.

Table 2.1: Five factors that can cause emotional changes (from [59])

Appraisals of eliciting event (E)	How suddenly and abruptly did E occur? How familiar was the person with E? How pleasant/unpleasant is E in general, independently of the current situation? How important/relevant is E to the person's current goals or needs? ...
Physiological Symptoms	Feeling cold shivers (neck, chest), Weak limbs, Getting pale Lump in throat, Stomach troubles Heart beat slowing down/getting faster Muscles relaxing/tensing, restful/trembling (whole body) ...
Motor Expressions	Smiling, Frown, Tears Mouth opening, closing, tensing Eyes closing, opening Voice volume increasing ...
Action Tendency	Moving attention towards/away from E Information search Attention self-centered/directed towards others Physically moving towards/away from E ...
Subjective Experiences	Intensity, Duration, Valence, Arousal, Tension ...

computer programs have been implemented based on these dictionaries to describe and predict people's emotional states and behaviours in given situations. Among all these programs, INTERACT[5], which implemented ACT, is an interesting one to note.

On the other hand, studies from the point of view of "non-verbal behaviours" attempt to extract human emotions from their facial, bodily, vocal expressions, and other non-verbal behaviours during an interaction. Studies in psychology have shown that non-verbal behaviours is an important channel for expressing emotions as well [61], and that in real-life scenarios, selection of words does not necessarily reflect the actor's affective states. Furthermore, recognizing words used in real-life conversations is likely to be very difficult, especially when the communicating parties are feeling intensive emotions. Realizing the importance and advantages of non-verbal behaviour analysis, more and more work in this domain has emerged in recent years. While humans can detect and interpret interactive signals of their communicators with little or no effort, it is much more difficult to design and develop an automated system that accomplishes the same tasks. In the following paragraphs, we survey several examples of tackling the problems of machine detection and interpretation of human affective states from non-verbal behaviours, focusing on facial expression analysis and body movement analysis. Readers are recommended to refer to Pantic's work [49] and Zeng's work [66] for a more comprehensive review of previous work.

Vision-based and acoustic-based are two most common approaches in the domain of automatic detection of emotions from non-verbal behaviours. Recognizing the important role facial expression plays in delivering affective messages, a fairly large amount of vision-based methods have been applied to detect emotion from facial expressions. A typical first step of this detection is to get an objective description of facial expressions, leaving the judgement of emotional message underlying these signals to a higher-level of decision making. The Facial Action Coding System (FACS) [21], for example, is one of the most comprehensive and widely used sign judgement systems. In this anatomically-based system, visible effects of facial muscle activations are described by "action units" (AUs), after which high-level decision-making processes aiming to learn the underlying affective meanings of facial expressions are applied on the AU representations. One most desirable feature of using AUs and AU descriptors is its ability to represent the thousands of anatomically possible facial expressions that humans can perform, independently of what high-level interpretations these facial expressions may imply. Bartlett's work [3] is an example work that has taken this approach. However, building automatic AU detectors is not as easy as one might think. One difficulty in designing such auto-detectors comes from the differences, such as face shape, texture and behaviours, between individuals. This inter-personal difference in

[5] Accessible via http://www.indiana.edu/~socpsy/ACT/interact.htm.

facial structures would affect the performance of automatic AU detectors, and thus indirectly affect the performance of a generic classifier on top of the AU detectors applying to unseen persons. A recent work [13], Selective Transfer Machine[6] (STM), attempted to tackle this problem by personalizing a generic classifier in an unsupervised manner. Another problem researchers face when designing automatic AU detectors is that changes in pose, scale, illumination, input clarity, etc, can all cause different levels of visible changes of the same facial movements. To tackle this problem, as complement to the old databases which contain only front-view facial-movement recordings, databases consisting of profile-view [50] and even 3-D recordings [65] of facial expressions have been built up. In addition, realizing that "deliberate behavior differs in visual appearance, audio profile and timing from spontaneously occurring behaviours" [66], databases consisting of spontaneous facial expressions (including interview recordings to collect spontaneously expressions) have been built as well.

Studies have shown that bodily expressions encode affective messages as well [61, 15]. In fact, in situations where accuracy of analysis on facial expression might be affected, for example, situations where perception is from a distance, or situations where affective states can be conveyed through movements more easily, better results are likely to be achieved by including bodily expressions in affect analysis. However, while facial expression analysis has been receiving much attention in the context of emotion recognition, much less research on automatic recognition of bodily expressions has been done. Two recent surveys [32, 29] reviewed work on recognizing affect from bodily expressions using computational models, compared this kind of approach with that from the point of view of facial expression analysis, and discussed challenges researchers face in this field, such as the challenges in data collecting, labeling, modeling, the challenges in setting up benchmarks, and the ones in dealing with inter-individual and inter-cultural differences.

A typical process of automatic affect recognition of bodily expressions includes the following steps [29]:

1. collecting motion trajectories from sensor data,
2. segmenting data collected based on time windows or movement primitives,
3. describing segmented data using the selected feature set,
4. mapping the representation from last step to affective states.

[6] Program based on this method is accessible via http://www.humansensing.cs.cmu.edu/intraface/.

Step 1 and 2 address human movement analysis in general (i.e. not limited to affect recognition). With appropriate temporal segmentation being a common challenge faced by researchers from many fields (including those from facial expression analysis as well), most current studies use pre-segmented data. As for step 3, the following three approaches, or combination of them, are generally used for constructing feature spaces [29]: (1) Features describing human movement are hand-chosen and reduced by dimensionality reduction when necessary (e.g. [43]). This approach is most suitable in situations where sensor data cannot easily be related to a kinematic or shape-based model of human motions, for example, when sensor data is collected from a pressure sensor integrated in a seat (e.g. [17]). The disadvantage of this approach is that it is not grounded in psychological theories. (2) Features are selected based on findings from perceptual studies in psychology (e.g. [28]). (3) Features are defined as high-level descriptors in a movement notation system (e.g. [12]). Similar to facial expression analysis, a good movement notation system is beneficial to bodily expression recognition as well. However, despite the fact that several movement notation systems have been proposed (e.g. [6]), we still lack a widely-recognized notation system that can help map between movements and affective states quantitatively. Again, these three approaches are not mutually exclusive; on the contrary, they are usually combined together. It is interesting to note that across all the approaches, movement speed is selected as a feature in most studies. A rather comprehensive review on features that have been selected in previous works has been given in Kleinsmith's survey [32]. Step 4 aims at mapping representations of features to affective states. Results in previous studies [5] have shown that velocity and expansiveness correlate with arousal, and that the basic posture relates to the expressed evaluation of valence, with a contracted posture for low valence and an open posture for high valence. In this step, classifiers are usually trained and/or regression techniques are usually applied. One can find an overview of machine learning methods that have been applied in previous studies in Klensmith's survey [32].

Aside from vision-based approaches, acoustic-based approaches have also been taken for affect recognition. Popular acoustic features used in existing approaches include prosodic features (e.g. pitch-related features, energy-related features and speech rate) and spectral features (e.g. MFCC and cepstral features). Among all these features, pitch and energy have been reported to contribute the most to the speaker's affective states in studies on "artificial" datasets (e.g. [33]). However, as indicated by Batliner et al. [4]: "The closer we get to a realistic scenario, the less reliable is prosody as an indicator of the speaker's emotional state". Some work [35] included words spoken in emotion detection as well and improvement was indicated. However, including words spoken as a feature in practical automatic affect detection systems might be infeasible or even unnecessary. First, whether current automatic speech detection can reliably recognize words spoken in emotional speeches is unknown [2]. Second, relationships between words chosen and the speaker's affective states have been reported to be rather unreliable [1]. Third, the association between linguistic content and emotion is language dependent, which certainly affects the performance of an affect detector applied to a language different from the one it was trained upon.

As discussed in Zeng's survey [66], open questions in affect recognition also include: utilizing contextual information, such as environment, observed subject, or the current task, in the process of affect recognition; appropriately segmenting data collected for analysis; constructing a dataset and setting up a benchmark that is shared by researchers within the field.

Affective Signals Generation

Generation of affectively modulated signals, which falls into the second aspect of including emotional intelligence in HCI, is to some extent the reverse of affect recognition. A typical process of such generation includes the following steps [29]:

1. deciding affective state to be encoded,
2. selecting movement type (e.g. facial expressions or bodily movements), based on the affective state decided in last step, or the functional task to accomplish,
3. modulating movement affectively, which means to add affective expressiveness to functional or abstract movements, and
4. generating trajectories (and/or carrying out motor commands for robots).

Depending on application scenarios, different movement types can be chosen and different movement modulation techniques can be used.

Adding affective signals to movements is believed to be beneficial to enhancing the believability of a virtual agent or robot [34]. Animators at Walt Disney Studios have proposed a set of 12 design principles to create believable characters, among which four are associated with the expression of affective states [34, 31]. Aside from the studies and experiences in the animation industry, research in developing well-performed embodied conversational agents (ECAs) has examined the importance of techniques in affective signal generation as well. ECAs are virtual entities with human-like communicative capabilities [11, 44]. ECAs communicate through verbal and nonverbal communication channels such as facial expressions, hand and arm movements, body posture, and prosody. Models to create behaviours based on emotions described by both categorical labels and dimensional representations have been proposed. To enrich the emotional behaviours of a virtual agent, some of the models that rely on discrete facial expressions used fuzzy methods (e.g. [10]). More commonly, models based on a dimensional approach are used because they allow the creation of a variety of expressions with subtle differences for related emotional states. Boukricha et al. [8] proposed a FACS [21] based model to generate facial expressions from emotions described by three-dimensional Potency-Activity-Dominance (PAD) vectors. In their approach, randomly generated facial expressions composed of several action units as defined with FACS were evaluated in terms of PAD values in an empirical study. The rated expressions were placed in the dimensional space, where Dominance takes one of two discrete values (high/low dominance), while pleasure and arousal values are mapped into a continuous space. With the help of multivariate regressions, the authors are able to map from PAD values to facial expressions. While most research on models of emotional displays focus on facial expressions, interest in multimodal expression of emotions in ECA have recently emerged. Findings in psychological studies have shown that emotions are expressed through a set or a sequence of different nonverbal behaviours, rather than a static facial expression. Niewiadomski et al. [44] surveyed several models that have introduced multimodality and sequentiality into generation of affective signals.

Dynamic generation of affective movements requires the processing of multivariate contextual information and much computational resources. Thus, in our prototypical handwashing system, based on the affective signals needed, the system selects a most appropriate affectively modulated prompt from a set of pre-generated and rated prompts. The set of pre-generated prompts were created and evaluated in Malhotra's work [38]. Malhotra designed 30 emotionally aligned prompts that could be used by a cognitive assistive system in a handwashing scenario. Three dimensional vectors in EPA space were used to represent the emotional interpretations of the prompts. The prompts covered all of the five situations where the system needs to suggest the user to "turn on the water", "use some soap", "rinse your hands", "turn off the water" and "use the towel". For each of these propositional actions, Malhotra designed the prompts to cover five cases where the same message was expressed with emotional impressions of "kind, powerful, active", "kind, powerless, inactive", "mean, powerful, active", "mean, powerless, inactive", and "kind, powerless, active". The two screenshots captured from Malhotra's prompts in figure 4.2 shows how the prompts look like. An online survey was conducted in which participants were asked to watch the 30 video prompts and rate them based on Evaluation, Potency, and Activity dimensions (on a discrete scale of -4 to $+4$ with increments of 1 for a total of 9 options). The questions were presented in random orders and participants were able to exit the survey at anytime.

There were a total of 27 respondents (16 male/9 female with 18 nationals and 9 internationals) who answered more than 90% of the questions. Analysis showed that participants' answers were consistent with each other. The mean of all valid ratings of a prompt was then computed as the final EPA vector for that prompt. With the correspondence between EPA values, instructional content, and the prompts generated by Malhotra, we are able to select at a given timestep a most appropriate prompt based on the required affective signals and instructional contents.

Psychological Study and Computational Models of Emotional Interactions

There exists a large body of work in psychology and sociology that studies human emotions and their roles in interactions. In these studies, emotions are usually represented as categorical labels, or dimensional values. Models, such as ACT [57], describing how people perceive emotions and how their interactions are regulated by these emotions have been proposed. Section 2.1 Basic Concepts should give readers a conceptual overview of the studies of human emotions; a more comprehensive review of this area is beyond the scope of this thesis.

Basing on findings in studies of human emotions, significant work in affective computing that uses probabilistic reasoning to build intelligent interactive systems have emerged. Psychsim [56] is an example of such interactive agents. In Psychsim, a POMDP model of psychological consistency theories was employed to estimate the relative value of actions. Various appraisal dimensions and a variety of influence factors such as consistency, self interest and "bias" was used. However, since the dimensions and influence factors were defined in an application-specific manner, it is not clear if they would generalise to other applications. A second example of such systems is FLAME [19]. FLAME combined fuzzy logic with reinforcement learning to achieve adaptivity. Emotional states and actions in FLAME were generated following application-dependent appraisal rules based on the OCC model [47] and a set of ad-hoc rules, respectively. Conati and Maclaren's work [14], which used a decision theoretic model and relied on sets of labelled emotions and rules from appraisal theories, is another example of an affectively intelligent interactive system. However, coming along with the advantage of easing interpretability and computability, and allowing for the encoding of detailed prior knowledge into applications, are the difficulties of generalization rule-based approaches have to face.

Different from rule-based systems, BayesACT [24] used a more general set of appraisal dimensions and describes identities and behaviours by values of these dimensions, regardless of their high-level interpretations. By operating completely in a dimensional space, BayesACT is able to "surmount computational issues, to assure scalability (the state space size only grows with the amount of state necessary to represent the application, not with the number of emotion labels), and to explicitly encode prior knowledge obtained empirically through a well-defined methodology" [24]. The authors of [24] also implemented and tested in simulations a Python program based on the BayesACT model. In their implementation, a class called *Agent* implements all the emotional parts, and subclasses of *Agent* implement the functional parts. Readers are referred to Section 3.6 in [24] (full version) for more details of their implementation. Despite the fact that the BayesACT model is easy to generalize for different applications, and thus can potentially be used as an emotional "plug-in" for systems that interact with humans, converters that map between actions (from both human users and the system) and dimensional values are still required for it to work in a real-world scenario. In other words, the BayesACT model did not tackle the problems of affect recognition and generation.

2.2.3 Hands Recognition Approaches

Noting the importance of accurately recognizing and tracking the user's hands in a handwashing system, this thesis reviews several typical hand tracking approaches in the following paragraphs. A formal analysis of and comparisons between these methods is beyond the scope of this thesis.

One approach of recognizing hands and other objects is based on skin-color [41]. In this approach, statistics-based color segmentation models and background subtraction were combined to identify objects, such as the user's hands and the towel, within the field of view. This approach is independent of time and thus is only able to recognize objects (rather than to track them). Another disadvantage of this method is that it is prone to noise, in the sense that skin colors of an object can change due to lighting-condition changes in the environment. A vision-based agent, which utilized this approach to recognize the user's hands and the towel, has been developed to assist persons with dementia during handwashing [7]. The locations of the user's hands and the towel are extracted by the agent for each frame, and the locations of other objects (e.g. the soap) are predefined in the approach. The user's hand behaviours are then detected by comparing the hands' locations with other objects' locations, and are used to update the beliefs of where the user is at during the handwashing process.

Another method to track hand locations is to use flocks of features [25]. A flock is a loose collection of features, or members. The features are characteristics of the local appearance of the object that is to be tracked. An approximate Bayesian sequential tracking method that uses flocks of color specks was used in a previous work of Hoey et al. [25]. To allow for long-term tracking of multiple objects, the authors also used a combination of three mixed-state particle filters [27] with data-driven proposals [45] and simple interactions to enable reinitialization after a track is lost. The method is robust to partial occlusions, distractors and shape changes. It is also able to consistently track objects over long sequences; as opposed to the previous ones which are independent of time.

Different from the above approaches, Czarnuch and Mihailidis utilized the depth information of images to track human bodies [16]. A C program has been developed and tested by them to track the user's body parts (such as head and hands) from an overhead perspective. The program has no prerequisites on environmental conditions or physical settings of the system, other than that images should be grabbed with depth information from an overhead perspective, which is easy to satisfy by mounting a RGBD camera above the sink area. The tracker was trained using partially labeled, unbalanced data, and has been shown by the authors in their paper that it is able to recognize and track the user's hands. The tracker is also configurable and re-trainable. One could recollect data and retrain the tracker if the tracker is to run in an environment different from the one which it was trained in. All these desirable features of the tracker make it a good candidate to design and develop our hand-washing system upon. More approaches on tracking human body parts using depth information of images can be found in Czarnuch and Mihailidis's paper [16], and Shotton et al.'s work [62].

Chapter 3

System Design

Our goal is to design an extensible system that can assist people with dementia during a hand-washing process by assessing their states and provide instructions accordingly. The system should combine affect recognition of behaviours, affective reasoning during interactions, and affective signals generation with AI techniques. This chapter illustrates how our system is designed as an integration of independent components.

People with certain types of cognitive disabilities have trouble in accomplishing daily tasks. For example, persons with dementia tend to forget where they are in a handwashing task, and without a caregiver's reminder, they might not be able to proceed. However, the need of human assistance in everyday tasks can cause the persons with dementia to feel unconfident and depressed. The frustrating caring jobs could also add a burden to families.

An assistive handwashing system is a cognitive intelligent system that can monitor user's behaviours and give proper prompts at certain times. Basically, the system observes the outside world and analyzes the observations over time. At each time step, the system updates its belief states about where the user is at in the handwashing task, and if needed, gives out proper prompts based on these belief states. One limitation of previously-built handwashing systems is that they did not consider users' emotional state changes during the interactions. In other words, the previous systems work functionally, but not emotionally. This defect could limit the effectiveness of the systems when expressing their instructions to human users. The handwashing system we design is able to compute both the user's functional states (i.e. where the user is at during the handwashing process) and emotional states (i.e. the user's identities and emotional states at each time step). The prompts it gives out are produced based on both of the above two states.

3.1 Overview

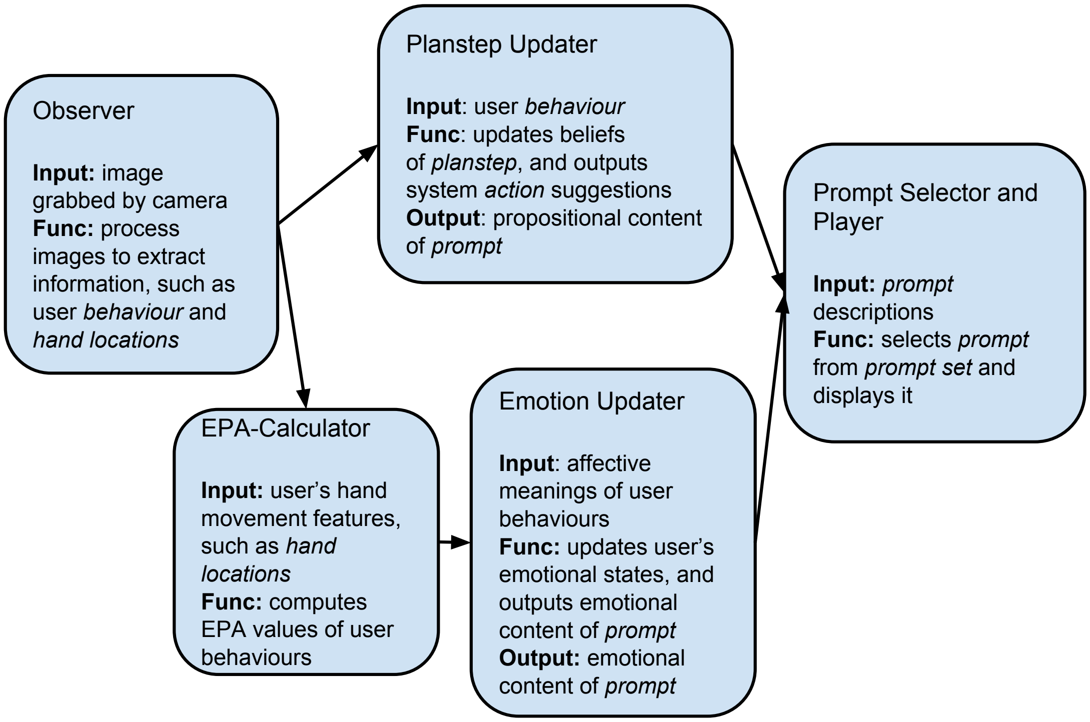

Planstep Updater
Input: user *behaviour*
Func: updates beliefs of *planstep*, and outputs system *action* suggestions
Output: propositional content of *prompt*
Observer
Input: image grabbed by camera
Func: process images to extract information, such as user *behaviour* and *hand locations*
Prompt Selector and Player
Input: *prompt* descriptions
Func: selects *prompt* from *prompt set* and displays it
Emotion Updater
Input: affective meanings of user behaviours
Func: updates user's emotional states, and outputs emotional content of *prompt*
Output: emotional content of *prompt*
EPA-Calculator
Input: user's hand movement features, such as *hand locations*
Func: computes EPA values of user behaviours

Figure 3.1: Functionality analysis on the system

Figure 3.1 describes the essential components of the hand-washing system based on a functionality analysis on the system. It shows also the data flow between components in the hand-washing system. The user's movements are observed and parsed in the **Observer** to extract information, such as user behaviours (e.g. "turning on water") and hand locations (i.e. the coordinates of the user's hands). User behaviours are then fed into the **Planstep Updater**, where beliefs (probability distributions) of where the user is at in the hand-washing process is updated and contents of instructions guiding the user to proceed is decided. On the other hand, hand movement features (such as hand locations) are fed into the **EPA-Calculator**, where emotion interpretations of the user's behaviours are computed. The **Emotion Updater** updates the beliefs of the user's emotional state basing on emotional interpretations of user behaviours, and produces the emotional content of the desired system action. Finally, the **Prompt Selector and Player** of the system selects/generates and displays proper prompts basing on the descriptions.

In the figure and our descriptions, planstep is used to denote the user's functional state and emotion is used to denote the user's emotional state. Prompt, which is an instructional message displayed by the system to the user, is used interchangeably with system action in our descriptions. The rest of this section analyzes the input, output, functionality and design difficulties of these components. How exactly each component is designed in our approach is explained. For easier description, "the **Output Part**" is used alternatively with "Prompt Selector and Player" to refer to the same component.

3.2 The Planstep and Emotion Updaters

3.2.1 The Planstep Updater

Planstep denotes the functional state of a user (i.e. how much has he/she accomplished) during the handwashing task. A set of subtasks of handwashing are defined, and each planstep is a combination of the completeness of these subtasks. For each timestep, the system checks if particular behaviours have been performed and updates planstep beliefs if particular subtasks are accomplished. A general definition of subtasks involved in a handwashing process uses three indicators: whether the water is on (on/off), the soap is on user's hand (dirty/soapy/clean), and the user's hands are wet (dry/wet). The corresponding planstep update diagram of this definition is shown in Figure 3.2.

Estimating if a subtask has been accomplished in the handwashing process is the most essential part in the planstep update. This can be broken down into two subproblems: (1) to check if an user behaviour has been performed, which is solved in the Observer; (2) to decide if a subtask has been accomplished accordingly.

Usually, sensors are put around the sink area to collect supportive evidence on recognizing user behaviours. One typical approach is vision-based (e.g. [25, 41, 16]): a camera is mounted above the sink to capture images of the area. For each video frame, by analyzing on the captured images, hands are tracked and their coordinates are obtained. By comparing the positions of the user's hands with the predefined ones of objects (such as soap, towel, sink and water), a system is able to check if the user's hands are within the neighbourhood of a certain object. If a certain area is detected, then the corresponding user behaviour is implied. For example, if the user's hands are detected to be in the neighbourhood of the soap, then he/she is believed to be putting on some soap on his/her hands.

Note that even if the hand locations are extracted without errors (which in reality is impossible), this location-based approach still can not claim to have 100% accuracy in

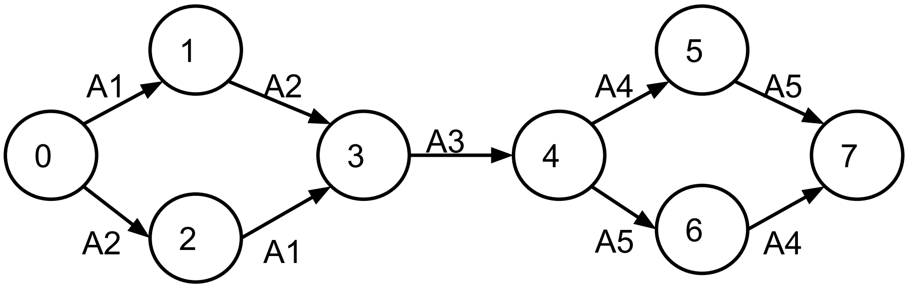


Figure 3.2: A planstep update diagram

Integers 0 to 7 denote the eight plansteps this subtask partition defines, which are (0) "off/dirty/dry", (1) "on/dirty/dry", (2) "off/soapy/dry", (3) "on/soapy/dry", (4) "on/clean/wet", (5) "off/clean/wet", (6) "on/clean/dry", (7) "off/clean/dry", respectively. A1 to A5 represent five behaviours completing the subtasks. A1 to A5 are "turn on water", "put on soap", "rinse hands", "turn off water", and "use towel", respectively.

detecting user behaviours. This is because having put one's hand in a particular position does not necessarily imply that the person is performing a certain behaviour. For example, the user can simultaneously put his hand on the tap while doing nothing, where a behaviour of "turning on water" or "turning off water" would be false positively detected by this approach. Adding more sensors (such as pressure sensors for detecting waterflow) might increase the accuracy of user behaviour detection; however, too many sensors is undesirable in our system design. Similarly, due to the subjective nature of the problem, having performed a behaviour does not necessarily imply the completeness of a subtask. For example, even though the behaviour of "using the towel" is detected, the dryness of the user's hands is still not predictable without further information.

Given the partially observable nature of user behaviours and the non-deterministic relationship between a user performing a behaviour and a user accomplishing a subtask, the POMDP model is used to design the Planstep Updater. To model the observation noise and the uncertainty between user behaviours taken and subtask completeness, a probability distribution is associated with each observation and possible behaviour. The probability distribution associated with observations gives the probability that one user behaviour is detected while another user behaviour is actually being performed. The probability distribution associated with behaviours, on the other hand, gives the probability that a subtask has been completed by the user when a certain behaviour is performed. With these two probability distributions, the Planstep Updater is able to update planstep beliefs based on user behaviour observations. Given the planstep of a user, the functional content of the system prompts is then decided based on a policy.

3.2.2 The Emotion Updater

The Emotion Updater in our approach is designed based on the BayesACT model. Three-dimensional vectors that represent *evaluation*, *potency*, and *activity* respectively (i.e. EPA vectors) are chosen to describe affective meanings in our work. Basic concepts of both of BayesACT and ACT are introduced in Section 2.1 of this thesis. As explained there, ACT can serve as a general psychological principle of micro-regulation of interpersonal interactions and BayesACT is able to learn the identities of users from their behaviours.

According to ACT, large deflections between the immediate impressions that the system prompts have on the user and the user's fundamental sentiments of him/herself and the hand-washing system would cause the user's attention to shift away from the premium communication objectives (i.e. to have his/her hands washed). Thus, it is important for the system to act in a way that is aligned with the user's emotional states. For example, if the user thinks of himself as the "boss" in their interaction, a more polite suggestion rather than an order should be given to achieve effective communication. Unfortunately, diseases can cause identity shifts [37], and the identity of a person with dementia is not obvious at all [46, 58]. Therefore, the user's identity should be learned from his/her behaviours. This thesis designs the Emotion Updater based on the BayesACT model. It represents the identity belief of the user as a probability distribution and updates it using observation functions, where the observations are emotional interpretations (i.e. EPA values) of user behaviours. The Emotion Updater is able to learn the user's identity in theory and in simulation, but not yet evaluated in practice. This thesis only briefly addresses this learning of the identity in the experimental results.

3.2.3 As One Single Reasoning Engine

The two updaters together decide the functionalities of the Observer and the two together form the functionality center of the hand-washing system. Logically, they both update belief states of the system at each timestep, and compute prompt descriptions based on the beliefs. From an engineering perspective, the two can be combined and developed as a single reasoning engine. In our approach, the two updaters are implemented on the basis of the existing BayesAct framework (implemented in [24]). Figure 3.3 shows how this thesis designs the Planstep and Emotion Updater as a single reasoning engine based on the BayesAct framework. Some parameters that need to be set when using this engine are described in Table 3.1.

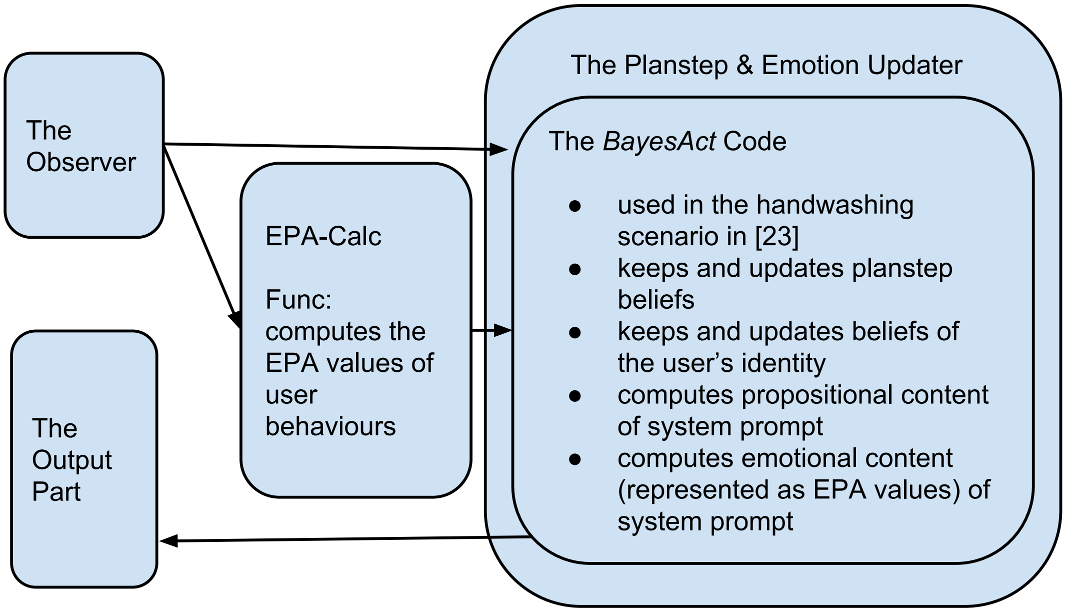

The Planstep & Emotion Updater
The *BayesAct* Code
● used in the handwashing scenario in [23]
● keeps and updates planstep beliefs
● keeps and updates beliefs of the user's identity
● computes propositional content of system prompt
● computes emotional content (represented as EPA values) of system prompt
The Observer
EPA-Calc
Func: computes the EPA values of user behaviours
The Output Part

Figure 3.3: Design the planstep and emotion updaters based on the BayesAct code

Recall that the BayesACT model includes states $S = \{X, F, T\}$, observations $\Omega = \{\Omega_x, \Omega_b\}$, and agent actions $\{A, B_a\}$. In our hand-washing system, $X = \{X_{turn}, X_{ps}, X_{aw}, X_{bahav}\}$, where X_{turn} describes whether the agent or the client is acting at this time, and X_{ps}, X_{aw} and X_{bahav} represents the current planstep, awareness and behaviour of the client. The observation Ω_x gives evidence to the system about X, including evidences indicating which party is currently acting, and what behaviour the client is currently performing if it is currently the client's turn. The system action A denotes the propositional content of a system message. For example, it can be "to instruct the user to rinse his hands". The observation Ω_b gives evidence to the system about f_b, which is an attribute in F representing the estimation of the affective meanings of the client's behaviours (when it is the client's turn). The observation function for the client behaviour sentiment $Pr(\Omega_b|f_b)$ is defined. It allows one to specify the "confidence" or "reliability" of the different components of Ω_b by γ, which is the variance of a normal (Gaussian) distribution (see Table 3.1 and also the start of Section 4.2). B_a, whose value is an EPA vector, denotes how the message should be expressed. For example, it can be "state the instructional message with a friendly tone".

A definition of eight plansteps is used to describe different functional states of the user

Table 3.1: Some of the parameters that need to be set when using the BayesAct engine

param	default value	meaning
β_a^0	0.01	initial identity variance for agent (larger means agent is more uncertain of its own identity)
β_c^0	0.01	initial identity variance for client (larger means agent is more uncertain of the client's identity)
γ	$(1.0, 1.0, 1.0)$	model noise variance of the E, P, and A values of user behaviours (larger means agent is more uncertain of the input)
N	300	number of samples used in the computation
f_a^0	$[1.5, 0.51, 0.45]$	the agent's initial belief of its own identity
f_c^0	$[1.59, 0.79, -0.88]$	the agent's initial belief of the client's identity

in a hand-washing process. The corresponding planstep update diagram of this definition is shown in Figure 3.2. The current planstep belief X_{ps} is a discrete variable with eight values (states), where the i-th state denotes the probability of the system currently being at planstep i. Two probability distributions are used to compute the X_{ps} transitions given an observation of the user's behaviour: the distribution $Pr : X_{behav} \rightarrow \Delta(\Omega_x)$ of the user's behaviour over behavioural observations, and the distribution $Pr : (X_{ps}, X_{behav}) \rightarrow \Delta(X_{ps})$ of the user's functional state moves from one planstep to another after a certain behaviour is performed. X_{aw} is a binary variable. It describes if the user is aware or not. A variable D is used to describe the current emotional deflection in the interaction and to infer how responsive a person is to a prompt. D corresponds to the differences between F and T. The dynamics in the system are:

— If the user is aware (i.e. has a high X_{aw} value), then if there is no prompt from the agent, the user will advance stochastically to the next planstep with a probability that is dependent on the current observation of user behaviour and the deflection D. If the user does not advance, she loses awareness (i.e. has a low X_{aw} value).

— If the user is aware (i.e. has a high X_{aw} value) and is prompted, and the deflection D is high, then a prompt will likely confuse the user and cause him/her have a low X_{aw} value. Again, this happens stochastically.

— If the user is not aware (i.e. has a low X_{aw} value), then if there is a prompt from the agent, and the deflection D is low, the user will likely follow the prompt, which causes the X_{aw} value to rise. Otherwise (i.e. there is no prompt, or the deflection D is high), the user will not do anything (or do something other than the one prompted) with high probability.

3.3 The EPA-Calculator

The BayesACT model does not tackle the problems of affect recognition and generation. Thus, for it to be used, an "input mapper" measuring the EPA values of user behaviours, as well as an "output mapper" mapping the propositional and EPA-vector descriptions of the system's guidance messages into concrete prompts are needed. The "input mapper" corresponds to the EPA-Calculator discussed in this subsection, and the "output mapper" corresponds to the Output Part of the system, which will be discussed in Section 3.5.

The problem of affect recognition from human behaviours, including affect recognition from bodily movements, facial expressions and sentences spoken, is a difficult machine learning problem. Chapter 2 of this thesis reviewed some of the previous work in recognizing affect from user behaviours; however, constrained by the specialty of our application scenario — to assist people with dementia (i.e. special user group) during hand-washing process (i.e. special use case where user's verbal and facial expressions can not be obtained easily and/or clearly) few of the previous approaches fit into our system naturally.

Recognition studies from facial expression analysis, which are relatively mature approaches, are not suitable for our application scenario. There are several reasons for this. Firstly, most facial expression analysis methods are based on data sets consisting of acted expressions; even though for approaches where spontaneous expressions are used, the expressions are performed by normal, healthy persons rather than persons with dementia, the user group of our hand-washing system. Since diseases could cause physical changes (e.g. persons with dementia might have fewer facial expressions), the performance of the facial expression analysis methods claimed to have high accuracies elsewhere might not remain the same when applied to our application. Moreover, most existing approaches analyse clear front-view facial expressions, which are difficult to obtain during the handwashing process.

Studies have as well examined acoustic-based methods that recognize affect by analyzing acoustic features, words spoken, or even sentence structures. However, speech recognition for elderly persons is harder than for younger people, and dementia can cause additional difficulties. Also the ambient noises (e.g. water running) can be a possible problem. Thus it is not feasible to collect user voices clearly during the hand-washing process, not to mention that extracting and analyzing words selected and sentence structures from audio recordings is itself a big challenge.

One possible approach to deal with the user-group constraints of our application is to select features by cooperating with sociologists, psychologists and physiologists, and to collect and label data. However, the approach requires a large amount of time, effort and expertise. With its focus on building a prototypical system integrating emotional intelligence, this thesis project leaves this for future work.

In our approach, the expansiveness between user's hands and the velocity of the user moving his/her hands are the features chosen in the system for computing affective meanings (potency and activity, to be specific) of user behaviours. This selection is supported by previous studies. For example, in the work of Beck and his colleagues [5], the authors derived a relationship between motion parameters and affective state, and showed that velocity and expansiveness correlate with arousal.

Threshold-based approaches are used to map expansiveness and velocity levels to potency and activity values of the EPA vectors representing a user's behaviours. The evaluation value of the EPA vector remains as default values in this prototypical approach. The calculated EPA representations are then fed into the Emotion Updater where the system's belief states about the user's identities are updated. Based on the belief states, the Emotion Updater then produces the affective meanings required for system prompts according to the principle of minimizing deflection. The EPA-Calculator is designed and implemented as an independent component from the Emotion Updater, which makes it easy to improve the calculator's performances by adding more features or employing new machine learning models without affecting the design of the Emotion Updater and other components in the system.

3.4 The Observer

The Observer perceives the world by sensors. What sensors should be mounted and what information should be extracted depends on the input requirements of the Planstep Updater and the EPA-Calculator. Concluded from the discussions in previous sections, the functionality of the Observer is to detect user behaviours (required by the Planstep Updater) and to compute expansiveness and velocity of the user's hand movements (required by the EPA-Calculator). Our design uses location-based methods to produce these features from hand locations. Figure 3.4 shows how this is done.

As shown in Figure 3.4, location-based methods detect user behaviours by comparing the locations of the user's hands with those of the objects we are interested in (as in [7, 16]). With pre-defined coordinates of certain objects (e.g. the soap, the towel, etc), a user behaviour (e.g. "using the towel") is said to be detected if the corresponding object (e.g. the towel) is close to the user's hands. If multiple objects are close to the user's hands, then the closest one is said to be selected. Apparently, the range at which the hands are

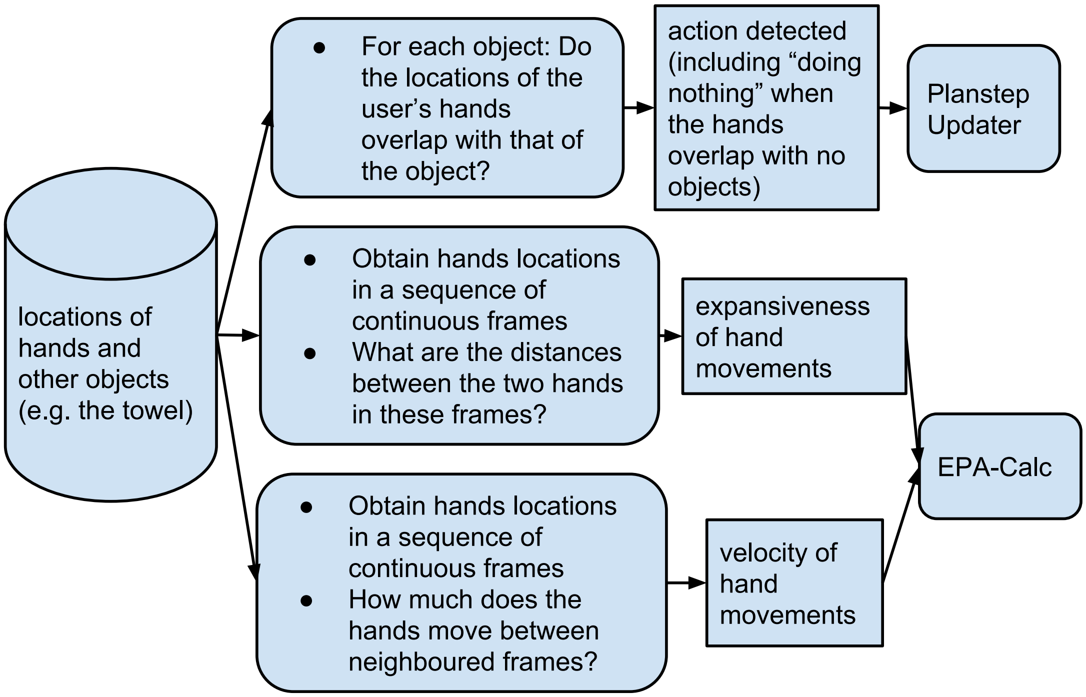

For each object: Do the locations of the user's hands overlap with that of the object?
action detected (including "doing nothing" when the hands overlap with no objects)
Planstep Updater
locations of hands and other objects (e.g. the towel)
Obtain hands locations in a sequence of continuous frames
What are the distances between the two hands in these frames?
expansiveness of hand movements
EPA-Calc
Obtain hands locations in a sequence of continuous frames
How much does the hands move between neighboured frames?
velocity of hand movements

Figure 3.4: How the location info of hands and other objects are used in the system

considered “close enough” to an object should be designed carefully. For example, the ranges should be different for different objects.

To be more sophisticated, as opposed to being a constant value, the range associated with an object in a location-based method could even be implemented as a probability distribution describing the probability of the user's hands touching the object given the distance between the coordinates of hands and the object. In such a case, comparisons between distances would become comparisons between probabilities, and the object that is closest to the user's hands would become the one that has the highest probability of being touched by the user. In the work of Hoey and Poupart [22], the authors described the probability distribution of user behaviours given hands locations directly as a POMDP observation function. In their approach, given observations of continuous x-y-z positions of the user's hands, the probability distribution of user behaviours was computed by the observation function, which was some Gaussian distribution.

Location-based approaches of detecting user behaviours have their limitations, in the sense that observing the user's hands at a position does not necessarily imply that the user is performing a certain behaviour. Including context information, such as which planstep the person is currently at, or utilizing additional sensors, such as ones that can detect water flow changes, might improve the accuracy of behaviour detection for location-based approaches. However, requiring too many sensors is undesirable for a system assisting people with everyday tasks. Luckily though, since the Planstep and Emotion Updater is designed and implemented as a POMDP in our system, all this uncertainty of observations is gracefully handled.

As mentioned above, essentially, the Observer needs to recognize the user's hands from grabbed images and obtain the coordinates of the hands. Chapter 2 of this thesis reviewed some previous work in tracking human hands, among which Czarnuch and Mihailidis's approach utilized the depth information of images [16]. Their tracker is able to recognize and track the user's hands accurately and is a suitable base for the Observer component of our hand-washing system. Chapter 4 of this thesis gives a more detailed explanation on the accuracy of the tracker recognizing user's hands. This thesis designs the Observer component of the system as an extension to the tracker developed by Czarnuch and Mihailidis in their work. The coordinates of a set of objects, which are the soap, the towel, the taps, the sink and the running water, are predefined.

The Observer of our system uses an RGBD camera mounted above the sink area to grab images and extract features of the user's hand movements. A 4D camera is better than a normal camera in the sense that it is able to get depth information of objects as well. With this additional information provided, image analyzers are able to achieve more accurate results. In fact, it would be difficult for the analyzers to differentiate the vertical distances between the user's hands and certain objects without depth information, in which case the user would be detected as "using the soap" whenever his/her hands and the soap are projected at the same place on the x-y plane (regardless of what the z values are).

3.5 The Output Part

A prompt is an instructional message the handwashing system gives out suggesting what the user should do next. As discussed before, in order to convey the intent of the instructions effectively to its user, the system should carefully decide how it should express the instructions. That being said, a prompt is defined with two components: the proposition and the emotion. The propositional part represents "what behaviour the user should perform next", while the emotional part indicates "how the instructional message should be expressed". Two prompts are considered identical only when they contain the same propositional content and are expressed with the same emotion. For example, the audio messages "please turn on the water" and "please put on some soap" are different prompts, since they contain different propositional contents. The audio messages "put on some soap now you slowpoke!" and "could you please put on some soap?" are different prompts as well, since they have different emotional interpretations.

While the main functionalities of the updaters are to update the belief states of the user's planstep and emotional state, they also compute the propositional (i.e. what instructions should be given) and emotional (i.e. how the instructions should be expressed) descriptions of prompts. The descriptions are then fed into the output part, where final prompts are either selected from predefined prompt sets or generated dynamically. This falls into the category of affect generation. The output part also displays the final prompts if needed. Note that prompts might not be needed at every timestep. In fact, if the user is performing smoothly by himself/herself, he/she might even get interrupted and confused by prompts. For easier descriptions, an "empty prompt" is defined to describe situations where no prompt is needed.

Dynamic generation of affective messages is a large challenge, as exemplified by the literature on embodied conversational agents (see [11, 44], and Chapter 2 of this thesis). Our approach selects the final prompt to display to the user from a set of pre-generated and evaluated prompts. The four most essential questions involved in this approach are: (1) Deciding the format of the pre-generated prompts: should they be video, audio, or textual prompt? (2) Designing the prompts, e.g. the words used in the prompts. If the prompts are audio or video prompts, the tones how the messages are stated should be carefully designed as well. Character gestures and other details might require consideration as well if video prompts are used. (3) Labeling the generated prompts. While it might be easy to label the propositional content of the prompts, labeling the emotional interpretations of the prompts might require additional effort. (4) Selecting the prompt to display based on the propositional and emotional descriptions at each timestep.

The effectiveness of different prompt formats depends on many variables, including personal preference and physical states of the users. For example, some users might find that videos with more information encoded are more helpful, while some others might not even look at the screen. The performance of prompt formats depends on the users' physical states as well. Ideally, the system should learn users' responsiveness to different prompt formats and select the most appropriate ones, or allow users to set prompt format preferences themselves. The format of audio-visual is used for the purpose of demonstration in our prototype. The format of audio-visual was used in the previous work on COACH system as well [40].

Prompt design is very important as it directly affects the usefulness of the prompt. The user would have difficulty understanding the system's intent if an ambiguous prompt is displayed. Aside from stating its intent clearly, the system should also encode emotion in the prompts. For example, different character gestures should be used to infer to different emotional interpretations. The more prompts with different emotional interpretations are generated, the more choices from which the final prompt is selected are provided and the more accurate results are expected. However, the number of pre-generated prompts is limited. Thus, the number of prompts pre-generated and the emotional interpretation differences between the prompts both need to be carefully decided.

This thesis uses the set of audio-visual prompts generated and evaluated in Malhotra's study [38] as a prompt dataset and selects the most proper prompt from it. Malhotra designed 30 emotionally aligned audio-visual prompts that could be used by a cognitive assistive system in a handwashing scenario in her study. An empirical survey was conducted by Malhotra to get the prompts evaluated in terms of EPA values. Readers can refer to Chapter 2 for more details of Malhotra's study. Given the set of pre-generated video prompts with both propositional and emotional labels, and the propositional and emotional descriptions of desired prompts computed by the Planstep- and Emotion-Updater, the Output Part is able to select a most proper prompt from the set by choosing the prompt with the same propositional labels and the closest emotional (EPA) values from the pre-evaluated set. The difficulty of the selection then lies in the definitions of emotional closeness between prompts. The emotional closeness of two prompts is related to the Euclidean distance between their emotional labels (i.e. the EPA vectors representing their emotional interpretations). Readers can refer to Chapter 4.4 of this thesis for more details of the algorithm used in our approach to choose the final prompt from the prompt dataset.

After a most proper prompt is chosen, the Output Part displays it out to the user. The sub-component that serves as a multi-media player in the Output Part is designed as a VLC[1] SDK application. VLC SDK is a mature and easy-to-use media framework that can be embedded into systems to provide multimedia capabilities for the applications. Minor implementation details, such as the timings to display prompts (e.g. the minimum time length between displaying two prompts), should be decided carefully.

3.6 Communication between Components

In our system implementation, if an open source package is used, it is used as is (of course, with some interface wrappers if necessary), which means that it has not been rewritten in languages different from their original implementations or been rewritten into APIs. There are several reasons for this. Firstly, the original implementations are usually stable. These widely distributed open source packages have already been tested by the developers who developed them and a large number of users who used them. Secondly, it is easier to extend or maintain the handwashing system by using the open source packages as is. In this way, one can easily update the system by replacing the plug-ins to newer versions (possibly with some changes to the interfaces), whenever bugs are found and fixed in the packages, better performance is achieved by newer implementations of the packages, or more suitable packages are found. Thirdly, it saves time and effort to use the packages as what they are. To sum up, using existing packages without modification is safe, easy to maintain, extend and switch to other packages if needed, and effort-saving.

Figure 3.5 shows an overview of how the system is designed as independent components. Note that different components of our hand-washing system utilize different techniques (e.g. developed using different languages) and work differently (e.g. run as an independent thread v.s. serving as an API or component). Specifically, the Observer component is designed as a hand-tracker based on Czarnuch's tracker [16], where the original tracker was implemented in C. The Planstep- and Emotion-Updater is designed based on the BayesAct framework [24] which was written in Python. Therefore, coordination between these components is a problem.

The server-client model is utilized in our approach. A server stub and a client stub are developed on both communication parties to encode, send, receive, and decode messages for the two parities. Figure 3.6 shows an example of employing the server-client model in a

[1] An introduction to the VLC SDK can be found via https://wiki.videolan.org/LibVLC/.

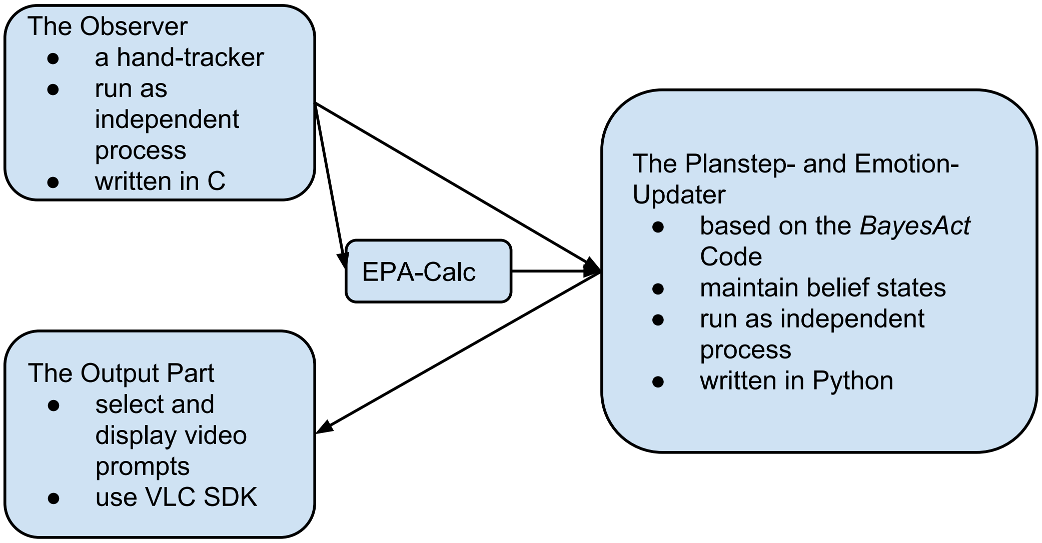

The Observer
● a hand-tracker
● run as independent process
● written in C
EPA-Calc
The Planstep- and Emotion-Updater
● based on the *BayesAct* Code
● maintain belief states
● run as independent process
● written in Python
The Output Part
● select and display video prompts
● use VLC SDK

Figure 3.5: Data flow and logical relationship between components

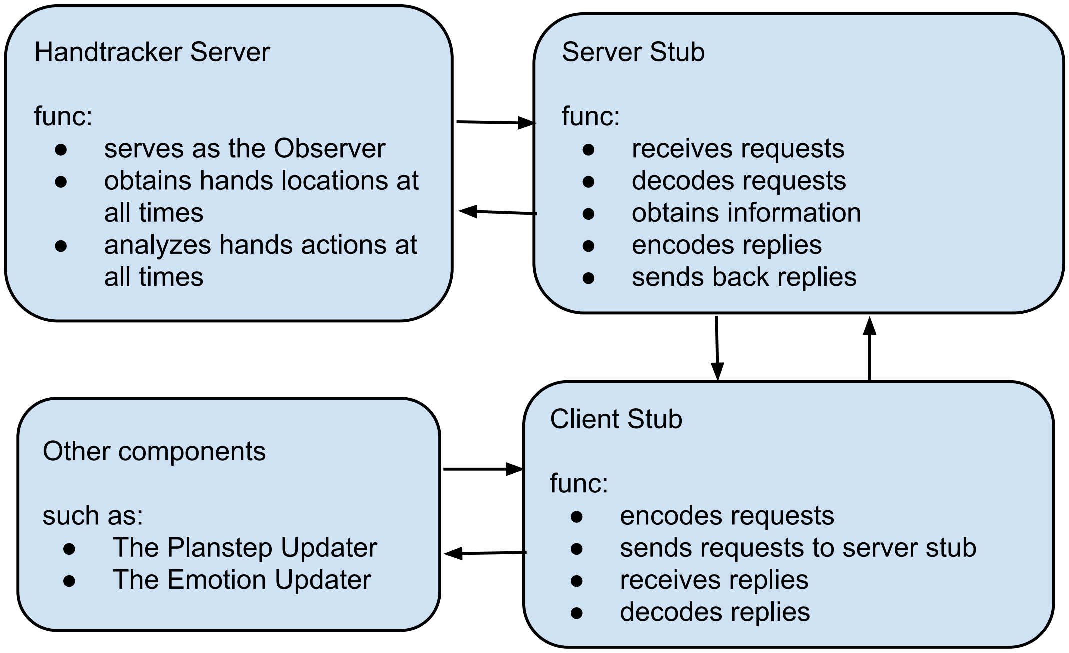


Figure 3.6: Communication between components using server-client model

scenario where a Hand-tracker Server that tracks the user's hands at all times is employed as the Observer component. The definitions of the request and response messages used in the communications should be shared by and accessible to both the the server and the client stubs. One way to achieve this is to define message structures on both the client and the server side, and manually ensure that the definitions are identical. This approach is easy to implement; but is vulnerable to bugs and is difficult to maintain. Another approach is to defined the message structures within a single "header" file accessible to both of the two sides. The "header" file should be referred to in the implementation of both the client and the server side. This approach is safer than the first one in the sense that no manual effort is needed to ensure the consistency of messages definitions on the two sides. However, it is infeasible to share same "header" files between two components implemented with different languages. A more sophisticated approach, which is also our approach, is to use Google's protocol buffer mechanism[2], which allows one to define message structures at one place, and to use them at another place. By compiling and automatically generating the definitions in another language, this mechanism ensures the consistency between message definitions shared by both of the communicating sides. The mechanism is also convenient to utilize. It is able to define data attributes in a structure as "repeated", "optional" and "required" fields, which is beneficial to define complicated data types. Useful accessible functions to data fields are provided by the mechanism. Furthermore, since the consistency between message definitions are automatically ensured, message structures defined using this mechanism are extensible.

The timings of the client requests and the server replies are carefully decided. The components shown in Figure 3.5 work with different frequency: the Observer recognizes user's hands, obtains and analyzes hand locations for each frame grabbed by the sensor at a rate about 24 frames/second , while the Updater works as a turn-taking model between the agent (i.e. the system) and the client (i.e. the human user). Our solution to solve this problem is to use a buffer between the Observer and the Updater. Information (such as hand locations) obtained from the Observer is saved and analyzed in the buffer. EPA values of user behaviours computed by the EPA-Calculator are stored as well and can be temporally smoothed in the buffer. The algorithm the Buffer uses to temporally smooth the EPA values of user behaviours is explained in Section 4.2. Messages are sent to the Updater for state updates only when certain conditions are met.

The following three aspects should be carefully designed for the buffer: (1) the conditions which would trigger the buffer to transfer between states; (2) the buffer's size, i.e. the information processed for how many frames can be handled by the buffer; (3) the policy

[2] Introduction to the mechanism can be accessed via https://developers.google.com/protocol-buffers/.

describing how overflowed messages are dealt with. The state machine shown in Figure 3.7 illustrates how the buffer is designed in this thesis. As shown in the figure, there are three states of the buffer. State A represents the state where the user performs a same user behaviour within neighbouring frames. Two conditions can trigger the buffer's state to transfer from State A to another state: (1) If a user behaviour different from the current stable behaviour is detected, then the state of the buffer would transfer to State B. If the new behaviour is found to last for less than a *timeup* period, the state of the buffer would change back to State A, with the stable behaviour unchanged. (2) If the buffer has been in State A for a *timeout* period, then the buffer would change to State C, where the state of the buffer would change back to State A immediately after messages encoding the current stable user behaviour are sent to the updaters. State B of the buffer checks if a newly detected user behaviour is a stable one. Similarly, there are two conditions that can trigger the buffer's state to transfer from State B to another state: (1) If the new user behaviour different from the current stable behaviour is found to last for at least a *timeup* period, then the new user behaviour is considered stable. In that case, the state of the buffer would change to State C, where the buffer's state would change to State A immediately after messages encoding this new stable behaviour are sent to the updaters. (2) Otherwise, the buffer's state would change back to State A with the current stable user behaviour non-changed.

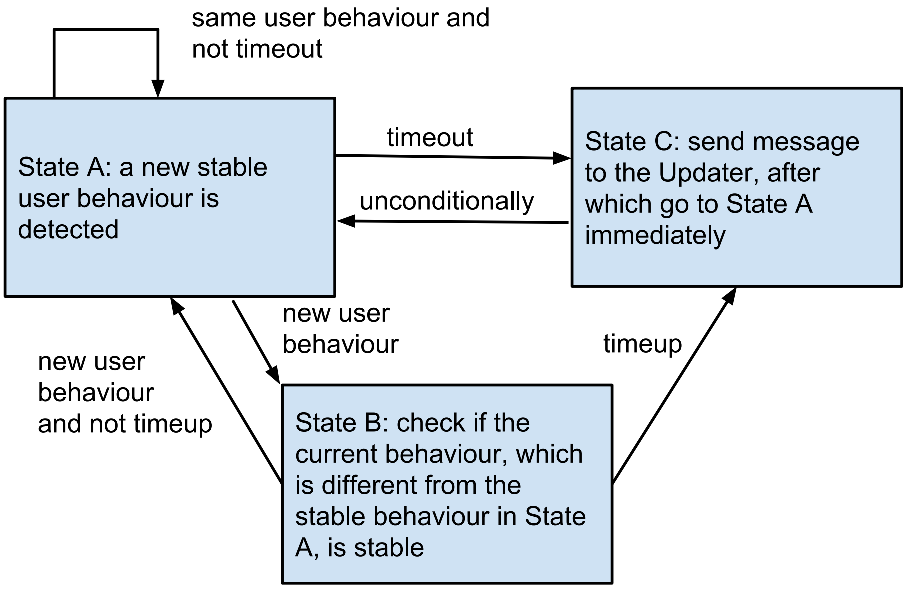

same user behaviour and not timeout
State A: a new stable user behaviour is detected
timeout
unconditionally
State C: send message to the Updater, after which go to State A immediately
new user behaviour
new user behaviour and not timeup
timeup
State B: check if the current behaviour, which is different from the stable behaviour in State A, is stable

Figure 3.7: State transitions of the buffer between the Observer and the Updater

3.7 Summary

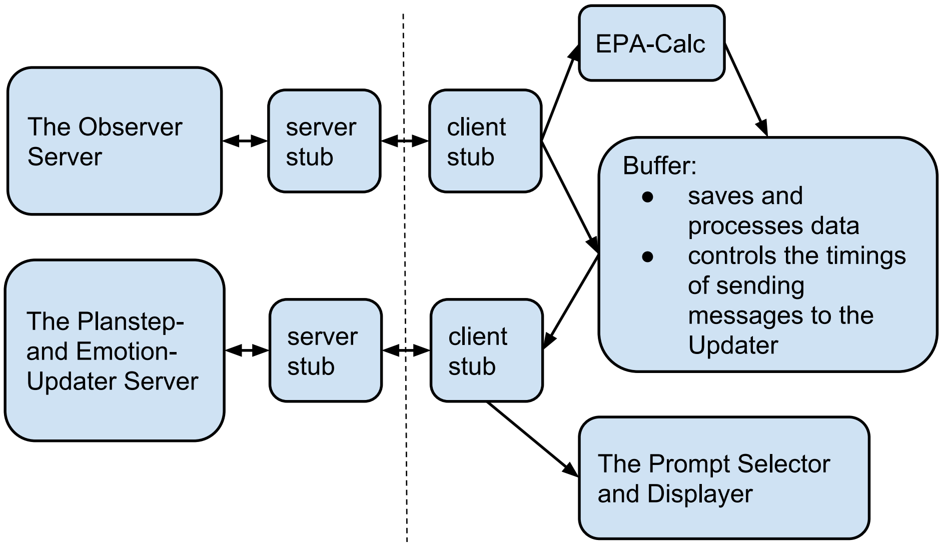

The Observer Server
server stub
client stub
EPA-Calc
Buffer:
● saves and processes data
● controls the timings of sending messages to the Updater
The Planstep- and Emotion-Updater Server
server stub
client stub
The Prompt Selector and Displayer

Figure 3.8: System design with server-client models

To sum up, the system is designed as shown in Figure 3.8. The Observer component of the system is designed based on Czarnuch's tracker [16], and is used as a hand-tracker server (see Sections 3.4 and 3.6). The Planstep- and Emotion-Updater is designed based on the BayesAct framework (implemented in [24]), and is used as an Updater server (see Sections 3.2 and 3.6. The rest of the components are designed as client components which communicate with the servers through stubs. Among all the client components, an EPA-Calculator (see Section 3.3) and a Buffer (see Section 3.6) sit between the two servers. The EPA-Calculator consumes user's hand locations and computes the EPA values of user behaviours. The Buffer stores and re-processes user behaviours and their EPA values, and controls the timings at which messages are sent to the Updater server for state changes. The final system prompt is selected and displayed by the Output Part (see Section 3.5).

Being designed as an integration of independent components, the system is easy to maintain and extend. For example, the Observer Component can be extended to obtain more features useful in computing EPA values for user behaviours without many changes to other components. This system design is portable as well. Since this design enables component communications across processes and multi-languages, one can easily use the same system design (possibly with some minor changes) in other application scenarios.

Chapter 4

Implementation

The system was implemented by developing new modules and combining both newly-developed components and existing packages together. Most of the code was written in C/C++, while some of it used Python. Server-stubs and client-stubs were employed, and Google's protocol buffer mechanism was used as the way to define the request and response messages shared by the two communicating parties. Open source libraries, such as ZeroMQ[1] and libVLC (i.e. VLC SDK), were utilized as well. Each module in the system was designed and implemented in an independent, efficient, and extensible way. The rest of this section describes the system implementation in detail.

4.1 The Planstep and Emotion Updater: a BayesACT reasoning engine

We implemented the Planstep- and Emotion- Updater on the basis of the BayesAct program developed by Hoey et al. [24]. In their program, a BayesAct framework that models emotional state changes during human interactions was implemented. Based on the framework, this thesis implemented a subclass of class *Agent* that simulates the actions of an automated assistant in a hand-washing scenario. The subclass is called *Assistant*. Class *Assistant* has an attribute field denoting the values of observed behaviours, and has methods that update belief states, including X and the relationships between X and Y (or, F

[1] ZeroMQ is a messaging library, which allows you to design a complex communication system without much effort. It creates an API that looks a lot like sockets, and feels the same, and gives you the messaging styles you want. More information of ZeroMQ can be accessed at http://zeromq.org/.

Table 4.1: Mapping between action suggested in BayesAct and prompt content representations in Prompt-Selector

Action description	Representation number in BayesAct	Representation number in Prompt-Selector
no prompts needed	0	0
ask the user to put on soap	1	2
ask the user to rinse hands	3	3
ask the user to turn on water	2	1
ask the user to turn off water	2	4
ask the user to to dry hands	4	5
tell the user all has been done	N/A	6
Invalid/undefined prompts	N/A	-1

and T)), based on observations. A function to get an estimation of the "current most-likely planstep" was defined in *Assistant* as well. The function returns the planstep that has the highest probabilities in the planstep distribution. POMDP observation functions were also defined in *Assistant*.

Noting the fact that an integer denoting the propositional messages contained in a prompt is used by both the BayesAct and the Prompt-Selector (the numbers come along with a set of predefined video prompts that the selector selects from), consistency between the two encoding systems must be assured. To achieve this, a converter on the server-stub side was implemented. The converter maps propositional descriptions of prompts returned by the server to their corresponding representations used by Prompt-Selector. Since "turning on water" and "turning off water" are denoted by same integers in BayesAct, additional information such as estimation of the current most-likely planstep is needed for the conversion. The converter is called before the server-stub packs and sends replies back to the client-stub. Relationships between the encoding systems of representing propositions used by BayesAct and the Prompt-Selector are illustrated in Table 4.1.

As far as the propositional content generation policy is concerned, two options exist: a POMCP policy and a heuristic policy. The POMCP policy computes a utility function which looks several steps further to produce prompts, while the heuristic policy constructs prompts based on mappings between the estimation of the "current most-likely planstep" and system actions. The mappings in the heuristic policy are obtained by heuristic knowledge and are shown in Table 4.2. Considering the fact that the heuristic policy runs faster and is sufficient for our prototype system, which is used to demonstrate the feasibility of including emotional intelligence in a practical system, the heuristic policy was chosen in

Table 4.2: Action suggestions basing on the current most-likely planstep (heuristic policy)

Current Most-likely Planstep				Actions Suggested
Number	Soap(dirty/soapy/clean)	Water(on/off)	Hand(wet/dry)	
0	dirty	off	dry	turn on water
1	dirty	on	dry	put on some soap
2	soapy	off	dry	turn on water
3	soapy	on	dry	rinse hands
4	clean	on	wet	turn off water
5	clean	off	wet	use towel
6	clean	on	dry	turn off water
7	clean	off	dry	N/A

our approach.

As discussed in the previous chapter, BayesAct is treated as a server in our implementation. Both a server-stub and a client-stub are developed. The server-stub listens to client requests at all times, and if there are any, processes them. The request processing tasks include: decoding the requests, passing arguments encoded in the request to the server for it to update belief states, and packing and replying the prompt descriptions obtained from the server to the client-stub. The client-stub packs and sends user hand-actions and EPA values of these actions to the server-stub, and waits for replies. When a reply is received, the client-stub decodes it, and converts the information accordingly, e.g. passes the information to a Prompt-Selector, where appropriate video prompts are selected.

```
message BayesactRequest {
  required double evaluation = 1 [default = 0.0];
  required double potency = 2 [default = 0.0];
  required double activity = 3 [default = 0.0];
  required int32 hand_action = 4 [default = -1];
 }

 message BayesactResponse {
  required double evaluation = 1 [default = 0.0];
  required double potency = 2 [default = 0.0];
  required double activity = 3 [default = 0.0];
  required int32 prompt = 4 [default = -1]; // propositional representation
  required bool is_done = 5 [default = false]; // if reaches the last planstep
 }
```

Definitions of request and response messages shared between the two communicating parties are defined using Google's protocol buffers (shown above), and the stubs are implemented using ZeroMQ libraries. The stubs bind or connect to given addresses in their constructors, and with the help of ZeroMQ, they are able to easily send and receive messages to/from each other. Among all the benefits that using protocol buffers and ZeroMQ brings us, the language-neutral advantage is one that is worthy of particular attention: it allows us to combine easily the BayesAct server and its stub with all other components together as a whole system, where the former ones were implemented in python, while the latter ones were implemented using C/C++.

4.2 The EPA-Calculator and the Buffer: computing and temporally smoothing EPAs

The hands' coordinates obtained from the hand-tracker server are fed into an EPA-Calculator, which represents a user behaviours as EPA values. In our prototypical approach, the *Evaluation* of the user's behaviour in all situations are computed as a neutral value and is considered as an uninformative observation in the Emotion Updater. This is done through defining the observation function for the client behaviour sentiment $Pr(\Omega_b|f_b)$ (see subsection 3.2.3). To let the system neglect the E value of the user's behaviour input to the Emotion Updater, we set $\gamma_e = +\infty$, where γ_e is the attribute in γ describing the "confidence" or "reliability" of the observation of *Evaluation* of the user's behaviour. The *Potency* and *Activity* are computed in the EPA-Calculator based on the distances between the user's two hands within a same frame and the distances that the user's hands have moved between neighbouring frames, respectively. The E, P, and A values computed for user behaviours in the EPA-Calculator are temporally smoothed in the Buffer before being fed into the Planstep and Emotion Updater. The "confidence" of the P and A values computed can be expressed by setting different γ values as well.

A parameter n is used to compute the P and A values of the user's behaviour. The average distance between the user's two hands in a set of n neighbouring frames is scaled to the P value of the user's behaviour in the EPA-Calculator. The average distance $Dist[i]$

of frames $i-n+1, i-n+2, ..., i$ can be computed using the following formula:

$$Dist[i] = \frac{1}{n} \sum_{k=i-n+1}^{i} dist(positions[k,0], positions[k,1]) \tag{4.1}$$

where function $dist : Point \times Point \rightarrow \mathbb{R}$ computes the distance between two points, and $positions[k,0]$ and $positions[k,1]$ are the coordinates of the user's left/right hand in frame k. A piecewise linear interpolation method is used to map the average distance $Dist[i]$ to the $P[i]$, which is the P value of the user's behaviour at frame i. Two same-length ascendingly sorted arrays of thresholds, *potency* and *distance*, are defined. The first and the last elements of array *potency* are -4.3 and 4.3[2], respectively. And the first and last elements of array *distance* are $-\infty$ and $+\infty$, respectively. The algorithm used in the mapping between $Dist[i]$ and the $P[i]$ is as following:

$$P[i] = (Dist[i] - distance[k-1]) * \frac{potency[k] - potency[k-1]}{distance[k] - distance[k-1]} + potency[k-1] \tag{4.2}$$

where $distance[k] \geq Dist[i] > distance[k-1]$.

A set of P values (say, from $P[i]$ to $P[j]$, $i \leq j$) are temporally smoothed in the Buffer to compute the final P value that is fed into the Emotion Updater. The smoothing algorithm ensures that the influence of the $P[k], i \leq k \leq j$, decays with time. The smoothing algorithm used in the Buffer can be illustrated by the following formula:

$$P = \sum_{k=i}^{j} (\frac{alpha}{alpha+1})^{j-k} * \frac{1}{alpha+1} * P[k] \tag{4.3}$$

where $alpha \geq 0$. If $alpha = 0$, then $P = P[j]$, which means that no temporal smoothing is used to compute the final P value.

Similarly, the A value of the user's behaviour are calculated based on the average of the distances his/her hands move between n neighbouring frames. The average movement $Diff[i]$ during frames $i-n+1, i-n+2, ..., i$ can be computed by the following formula:

$$Diff[i] = \frac{1}{n-1} \sum_{k=i-n+2}^{i} maxDiff(positions[k], positions[k-1]) \tag{4.4}$$

where $position[k]$ contains a pair of coordinates representing the user's left and right hand locations respectively in frame k, and function $maxDiff : (Point_a, Point_b) \times pair(Point_c,$

[2] Recall that *potency* is a real number within range $[-4.3, 4.3]$

$Point_d) \rightarrow \mathbb{R}$ returns $max(dist(Point_a, Point_c), dist(Point_c, Point_d))$. Piecewise linear interpolation method is used to map the average distance $\mathit{Diff}[i]$ to the $A[i]$, which is the A value of the user's behaviour at frame i. Two same-length ascendingly sorted arrays of thresholds, *activity* and *difference*, are defined. The first and the last elements of array *activity* are -4.3 and 4.3[3], respectively. And the first and last elements of array *difference* are $-\infty$ and $+\infty$, respectively. The mapping algorithm between $\mathit{Diff}[i]$ and $A[i]$ is as following:

$$A[i] = (\mathit{Diff}[i] - \mathit{difference}[k-1]) * \frac{\mathit{activity}[k] - \mathit{activity}[k-1]}{\mathit{difference}[k] - \mathit{difference}[k-1]} + \mathit{activity}[k-1] \quad (4.5)$$

where $\mathit{difference}[k] \geq \mathit{Diff}[i] > \mathit{difference}[k-1]$.

A set of A values (say, from $A[i]$ to $A[j]$, $i \leq j$) are temporally smoothed in the Buffer to compute the final A value that is fed into the Emotion Updater. The smoothing algorithm ensures that the influence of the $A[k]$, $i \leq k \leq j$, decays with time. The smoothing algorithm used in the Buffer can be illustrated by the following formula:

$$A = \sum_{k=i}^{j} (\frac{alpha}{alpha+1})^{j-k} * \frac{1}{alpha+1} * A[k] \quad (4.6)$$

where $alpha \geq 0$. If $alpha = 0$, then $A = A[j]$, which means that no temporal smoothing is used to compute the final A value.

Note that in the computations of P and A, variables n, *potency*, *distance*, *activity* and *difference* are used in the EPA-Calculator and *alpha* is used in the Buffer. The values of these variables need to be set when running the system. Moreover, the number of values used to compute the final P and A values needs to be set in the Buffer as well. Chapter 5 described a way to set the thresholds *potency*, *distance*, *activity* and *difference* by statistical results from experiments. In this prototypical approach, the unweighted means of distances were used and scaled to P's and A's; weighted average, and/or more other features can be included as indicators for the EPA values of user behaviours in future approaches.

4.3 The Observer: an extension to existing hand-tracker

Similar to the Planstep and Emotion Updater, we implemented the Observer as an extension to the tracker Czarnuch developed [16]. We added a server-stub and client-stub and

[3] Recall that *activity* is a real number within range $[-4.3, 4.3]$

made several changes to parameter values and other details. We explain in the following paragraphs the algorithms used in Czarnuch's tracker, the changes we made to the original program, and the implementation details of the server and client stubs. Note that a Kinect[4] camera is mounted above the sink when the Observer is in use.

In the original program, decision trees were trained on a set of images that were manually annotated to optimize key parameters. The trained tracker is then used to classify body parts in depth images grabbed from an overhead perspective. The tracker first generates a random decision forest using a simple depth feature to provide intermediate multiclass probability density functions (PDF) for each sampled image pixel. It then proposes final body part positions by aggregating the information contained in the underlying PDF. As explained by Czarnuch [16], the pre-trained decision trees included in their original program are able to classify body parts (including head and hands) as long as the camera is mounted from an overhead perspective to the objects and areas of interest.

After body parts (e.g. hands) are classified, the original tracker uses a location-based method to identify "hand behaviours". It first checks which pre-defined areas the user's hands are currently inside of. If multiple areas are detected, then a set of rules, such as comparing the distances from the areas' centers and current hand-locations, are applied to decide the "winner" area that the user's hands fall in at the moment. For example, suppose a "soap area" is defined as (s_x, s_y, s_z, s_r), where (s_x, s_y, s_z) are the coordinates of the area center in world space and s_r represents the radius of this area (i.e. the area is defined as within a spherical surface). If the current left hand-location is detected as at point (l_x, l_y, l_z), where $(l_x - s_x)^2 + (l_y - s_y)^2 + (l_z - s_z)^2 \leqslant s_r^2$, then this left hand is considered to overlap with the "soap area", which implies a behaviour of "using the soap". If the user's left hand overlaps with multiple areas, then the one with a closer center to the hand's location becomes the winning area. This same "detection of area and behaviour" process is performed for both of the user's hands and for all pre-defined areas. Note that in this approach, the user's two hands can be detected as performing different behaviours at the same time — one using the soap within the "soap area" and the other one doing nothing in particular at the sink. In the Observer we developed, when different areas/behaviours are detected for the user's two hands, a function comparing the two and returning a single winning area/action is called. More details about this comparison function are explained in the following paragraphs.

Figure 4.1 shows that the original tracker can recognize and return hand-positions with enough accuracy in our application scenario. Having this result, we did not retrain the

[4] Kinect is a line of motion sensing input devices by Microsoft for Xbox 360 and Xbox One video game consoles and Windows PCs. It is able to capture depth information of images. See http://en.wikipedia.org/wiki/Kinect for more details.

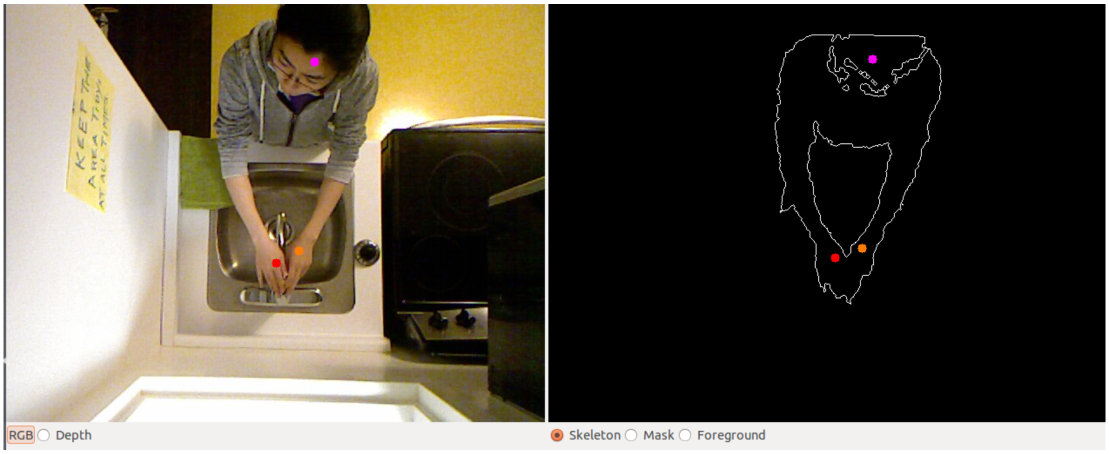


Figure 4.1: Screenshot of Czarnuch's tracker recognizing user's head and hands

model in our approach. However, if researchers were to retry our experiments, or to use Czarnuch's program in their own applications, they might need to recollect data and retrain the model. Fortunately, Czarnuch included both data collection and data training modes in his program.

In our Observer, the positions and coverages of the interested areas are assigned in a configuration file. To be specific, coordinates and radiuses of seven areas, including AWAY, SINK, SOAP, WATER, Left_TAP, Right_TAP, TOWEL are defined. Since there was only one tap involved in the laboratory setting of our experiments (see Figure4.1), the coordinates of Left_TAP (and Right_TAP) are actually used to model the positions of the user's left (and right) hand when he/she tries to turn on/off the water. Priority levels to user behaviours are defined as well (see Table 4.3). In the Observer, when different areas/behaviours are detected for the user's two hands, a function comparing the priority levels of two and returning the one with higher priority is called. Moreover, noticing that both the Observer and the Planstep and Emotion Updater represent areas/actions by integers, we implemented a converter function to convert between the two encoding systems to ensure that the two encoding systems are consistent with each other. Table 4.3 gives an overview of the relationships between areas and the user behaviours implied by them, along with the priority levels assigned to and the representations numbers used for these areas.

As discussed previously, the Observer is treated as an observation server in our implementation. Both a server-stub and a client-stub were developed. The server-stub listens to client requests at all times, and if there are any, processes them, which includes decoding

Table 4.3: Relationships between areas and the actions implied

Area Name	Action Implied	Priority Level	Representation Number used in the Observer	Representation Number used in the Updater
N/A	undefined action	0	0	0
AWAY	doing nothing	1	1	0
SINK	doing nothing	2	2	0
SOAP	putting on soap	6	3	1
WATER	rinsing hands	4	5	3
Left_TAP	turning on/off water	5	4	2
Right_TAP	turning on/off water	5	4	2
TOWEL	drying hands	3	6	4

the requests, asking the server for current hand-coordinates and hand-actions, and packing and replying to the client-stub the answers. The client-stub sends requests (for current user behaviours and hand locations) to the server-stub, and waits for replies. When a reply is received, the client-stub decodes it, and converts the information accordingly (e.g. maps the representation numbers of hand-actions used by the hand-tracker to the ones used by BayesAct).

Definitions of request and response messages should be shared between the two communicating parties. With the benefit of being language-neutral, platform-neutral, and easily extendable, Google's protocol buffers were used to define the message structures (shown below).

```
message HandTrackerRequest {
  optional int32 timestamp = 1 [default = -1]; // -1 means this field is not used
}

message HandTrackerResponse {
  message HandPosition {
    required float x = 1 [default = 0.0];
    required float y = 2 [default = 0.0];
    required float z = 3 [default = 0.0];
  }
  required HandPosition left_hand_position = 1;
  required HandPosition right_hand_position = 2;
  required int32 action = 3 [default = 0];
  optional int32 timestamp = 4 [default = -1];
}
```

The server-stub and client-stub can be easily implemented using ZeroMQ libraries. The stubs only need to bind or connect to given addresses in their constructors, and with the help of ZeroMQ, they are able to easily send and receive messages to each other. Of course, encoding and decoding of messages should be processed according to the message definitions above before sending out and after receiving messages, respectively. The client-stub also needs to map the representation numbers of hand-actions used by the hand-tracker to the ones used by BayesAct after decoding messages.

A function called processRequestsIdle() is implemented on the server side to have the server-stub listen to client-stub requests at all times. It checks the existence of requests, and processes them accordingly if needed. The function, along with another function called handTrackerIdle(), which observes and returns hand-locations and hand-actions, are registered as the server's idle callbacks.

4.4 The Output Part: Prompt Selector and Player

Both propositional and emotional prompt descriptions are passed into a Prompt Selector, where a most appropriate video prompt is selected from a set of pre-generated and rated prompts. After this final prompt is selected, it will be sent to a Prompt Player, which was implemented using VLC SDK, for display. This sub-section illustrates the implementation details of the Prompt Selector and Player in order.

The set of pre-generated and rated video prompts in Malhotra's survey [38] serves as the prompt dataset for our Prompt Selector. Malhotra created thirty prompts using the USC Virtual Human Toolkit[5]. Figure 4.2 with the two screenshots of prompts instructing the user to use some soap gives a general idea of what the video clips look like. Note that the message contents of the two prompts are the same (i.e. they intend to instruct the user to do same actions); it is the way how these messages were expressed that differ them apart. The character in the first screenshot was suggesting to the user to "try putting on some soap" with widely-open hands and a kind smiling face, giving the impression of a nice, a little bit dominant and active lady. On the contrary, the character in the second screenshot was stating "If you want to put on some soap, there is a soap pump lying around." with

[5] See http://ict.usc.edu/prototypes/vhtoolkit/

her hands crossed and her face unhappy. Different from the first one, the second character gives us the impression of someone who is more passive[6]. In fact, the survey results align with our intuitive impressions: the first video prompt was rated $[1.56, 1.17, 1.35]$ for its EPA values by participants, while the second one got a rate of $[-0.94, -0.67, 0]$ as its EPA values in the same survey. The EPA values rated for all the thirty prompts are illustrated in the appendix.

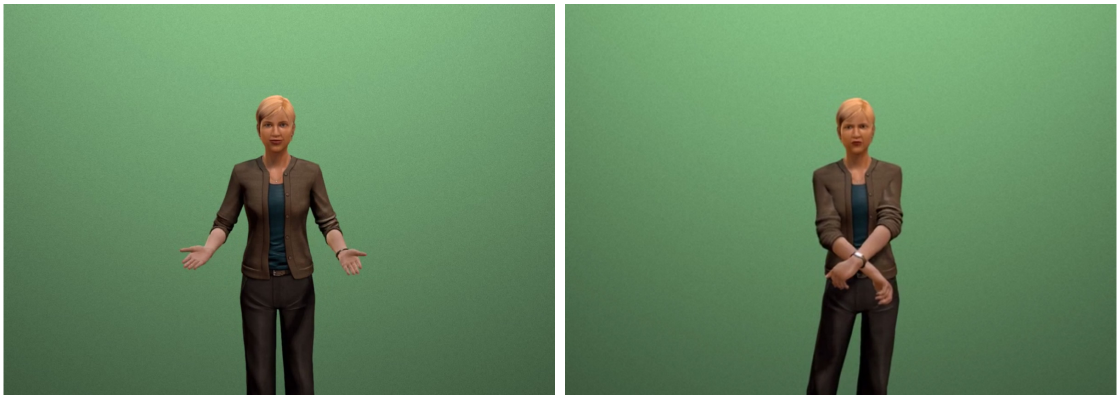


Figure 4.2: Screenshots of two video prompts stating same propositional messages

Each of the prompts in the dataset of the Prompt Selector has two labels: one states the propositional content of the prompt while the other one describes it emotionally. The propositional labels are assigned according to the intent of the prompts and are represented by integers. The mappings from “intent of prompts” to “propositional labels” are stated in Table 4.1. The emotional labels, on the other hand, are defined as the EPA values participants rated in the survey. For example, the prompt from which the first screenshot is labelled as $[1.56, 1.17, 1.35]$ for its emotional annotation.

To utilize the pre-generated and evaluated prompts as a dataset from which the most proper prompt is selected, a file describing all these prompts (e.g. what the labels of each prompt is) is needed. When the system starts, it reads the file and stores the descriptions of prompts and their labels in memory for later use. This thesis project uses a csv file with headers “filename”, “prompt_number”, “evaluation”, “potency”, “activity” to save information of the prompts. Among all the columns, “prompt_number” is the integral propositional label of a prompt and “evaluation”, “potency”, “activity” are three real numbers representing the EPA values of the same prompt.

[6] According to ACT, the second prompt might work better than the first prompt for a pessimist.

Given the propositional and emotional descriptions of desired prompts (obtained from the BayesAct server component), the Prompt Selector is able to select the most proper prompts in the dataset via a distance-based method. To illustrate how the selector works, we first define the distance between two vectors of EPA values as the weighted Euclidean distance between them:

$$distance((E_1, P_1, A_1), (E_2, P_2, A_2)) = \sqrt{w_E(E_1 - E_2)^2 + w_P(P_1 - P_2)^2 + w_A(A_1 - A_2)^2} \tag{4.7}$$

where w_E, w_P and w_A are weights for the three dimensions respectively. We then define the emotional distance between two prompts as the distance between their emotional annotations, i.e. the EPA values assigned to them. When given the propositional and emotional descriptions of the desired prompt, the selector selects out the video prompt that simultaneously has the same propositional label as and the minimal emotional distance to that desired prompt. In our implementation, the weights w_E, w_P and w_A are assigned values $\{1, 1, 1\}$; other values and even other definitions of the distances can be tried out in future improvements.

After the proper prompt is selected, a Prompt Player is used to display the video prompt. The Prompt Player was implemented using libVLC (VLC SDK), a mature and easy-to-use media framework that can be embedded into systems to provide multimedia capabilities for the applications.

Chapter 5

Experimental Results

As defined at the end of the Chapter 1, the four objectives of this thesis are to augment the COACH system with an emotional reasoning engine based on BayesACT so that the augmented system: (1) is designed in a portable and extensible way; (2) runs in real-time from the perspective of the user group; (3) provides at least a level of functional assistance of as high quality as the COACH; (4) is able to tune the prompts in some positive way according to the emotional state of a user. It has been shown in Chapter 3 and Chapter 4 that the system we developed is easy to extend. Experiments conducted on the system show that an average latency of 46.79ms is caused by the Observer component of the system, 0.009ms by the Buffer, and 1.65s by the Updater. The overall average latency of the system is 1.70s, with the maximum latency being 1.86ms and the minimum being 1.56ms. The results show that the system runs in real-time from the perspective of its user group.

We demonstrate in this section by laboratory based tests that the system is also able to provide a level of functional assistance and to produce system prompts that have encoded to some extent the emotional state of the user. The tests were conducted on a PC running 64-bit Ubuntu 12.04 LTS, with AMD FX(tm)-6300 Six-Core Processor 6 and NVIDIA GeForce GTX 650 Ti Graphics Card. A $Kinect^{TM}$ camera was mounted above the sink area and was the only sensor of the system.

5.1 Parameter Setup

This section explains how the values of threshold variables *distance* and *difference* used in the EPA-Calculator were assigned based on statistical results obtained from experiments

Table 5.1: Parameter values used in laboratory experiments

Param.	**Value**	**Defined in which component**
n	10	EPA-Calc, see Section 4.2
$distance$	$\{-\infty, 0, 8, 40, 128, 160, +\infty\}$	EPA-Calc, see Section 4.2
$potency$	$\{-4.3, -4.3, 0, 1, 2, 4.3, 4.3\}$	EPA-Calc, see Section 4.2
$difference$	$\{-\infty, 0, 3.5, 17.5, 35, 70, +\infty\}$	EPA-Calc, see Section 4.2
$activity$	$\{-4.3, -4.3, -2, -1, 0, 4.3, 4.3\}$	EPA-Calc, see Section 4.2
$alpha$	0	Buffer, see Section 4.2
$timeout$	300	Buffer, see Section 3.6
$timeup$	1	Buffer, see Section 3.6
β_a^0	0.001	Updater, see Subsection 3.2.3
β_c^0	2.0	Updater, see Subsection 3.2.3
γ	$(100000, 1.0, 0.5)$	Updater, see Subsection 3.2.3
N	2000	Updater, see Subsection 3.2.3
f_a^0	$[1.5, 0.51, 0.45]$	Updater, see Subsection 3.2.3
f_c^0	Different in each test	Updater, see Subsection 3.2.3

as follows. Videos were recorded while a person washed her hands with different emotions in nine complete hand-washing trials. A total number of 13,703 frames were extracted from the videos. For each frame, the distance between the person's two hands was computed. For each pair of neighbouring frames, the distances that the person's hands moved between the frames were calculated as well. Histograms of the two "distances" are shown in Figure 5.1. Note that in around 69% of the frames, the distances between the user's hands falls into the range of $(8, 40]$. Note also that in around 70% of frames, the distances the user's hands have moved from their positions in the last frames falls into the range of $(3.5, 17.5]$. Based on analysis of the distributions of the two "distances", we assigned the values of variables used in the EPA-Calculator as following: $distance = \{-\infty, 0, 8, 40, 128, 160, +\infty\}$, $potency = \{-4.3, -4.3, 0, 1, 2, 4.3, 4.3\}$, $difference = \{-\infty, 0, 3.5, 17.5, 35, 70, +\infty\}$, and $activity = \{-4.3, -4.3, -2, -1, 0, 4.3, 4.3\}$. Table 5.1 shows an overview of the values assigned to important variables in our tests of the system. Note that the variance of normal distribution γ, which specifies the "reliability" of the different components of Ω_b (see Figure 2.2), is set to $(10000, 1.0, 0.5)$. This means that the E value computed for the user's behaviour is considered uninformative in the reasoning engine, and that the P value is somewhat reliable and A the most reliable. See Subsection 3.2.3 and Section 4.2 for more explanation of the parameter γ.

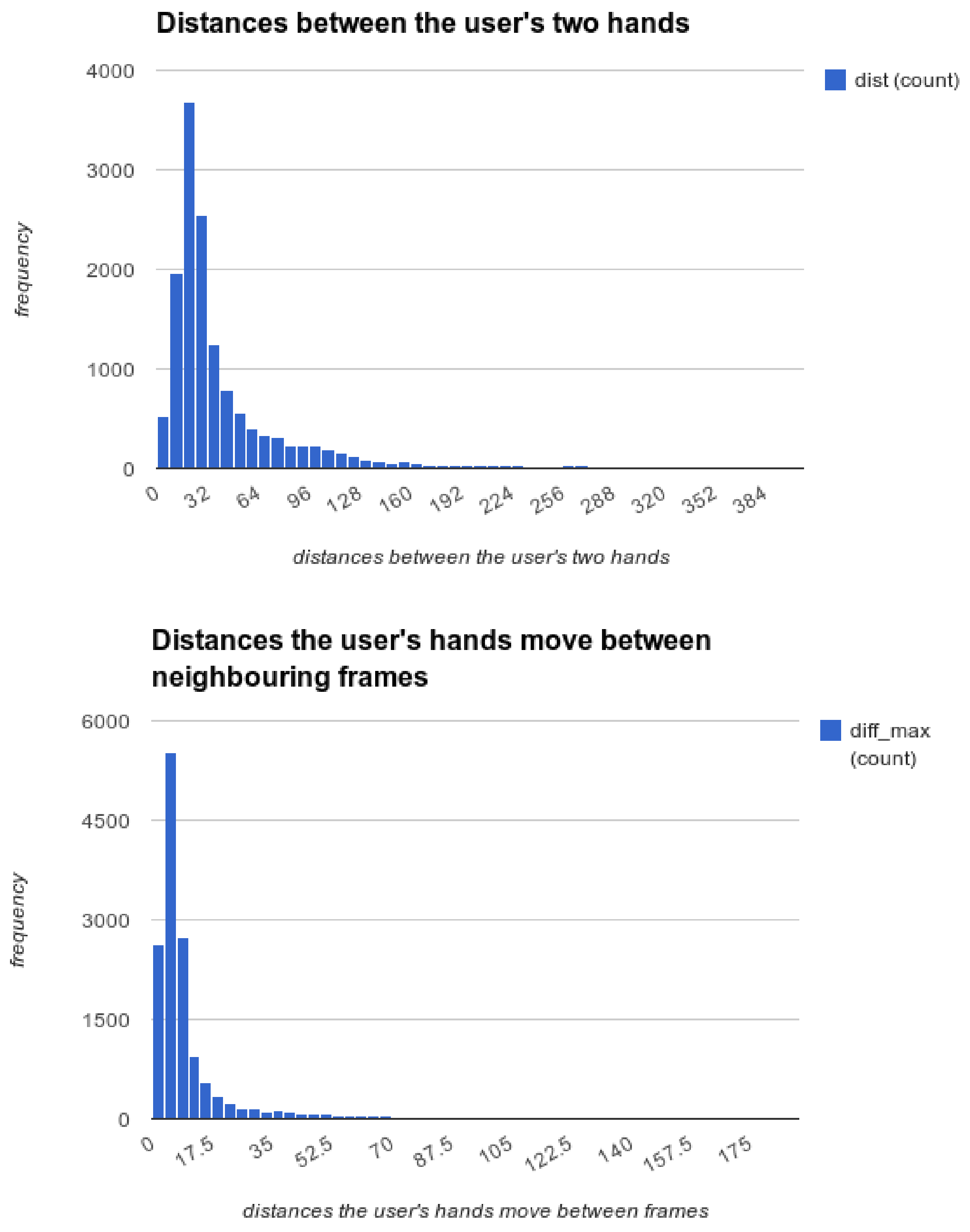
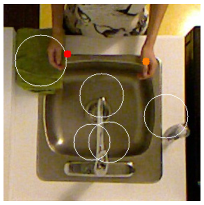
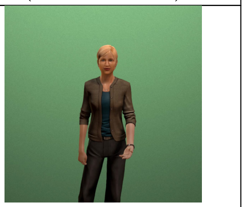


Figure 5.1: Histograms

5.2 Overview of Two Laboratory Tests

Tables 5.2 and 5.3 show the state changes of the system in two laboratory tests. In both of these tests, an actor was washing her hands while the system observes and assists her in real time. The difference between the two was that the actor acted more powerfully (with her hands more "open") and more actively (with her hands moving more quickly) in the first test than in the second one. f_c^0 was set to $[1.61, 0.84, -0.87]$[1] in test #1, and was set to $[-0.64, -0.43, -1.81]$[2] in test #2. Recall that f_c denotes the agent's belief of the client's identity, and f_c^0 denotes the initial value of this belief. Except for the columns "Time" and "User Behav. (screenshot)", all data in the table were computed by the system. The column "Planstep Belief" uses the definition of plansteps shown in Table 4.2. The video prompts displayed by the system during the tests are described in the table by screenshots and the avatar's lines. As shown in the tables, the Planstep and Emotion Updater (i.e. the BayesAct reasoning engine) updated its states for a total of 8 times in both test #1 and test #2. We explain using the tables in the remaining of this section that the system is able to work both functionally and emotionally by looking at the experimental results of the two tests more closely.

Table 5.2: State changes in test #1 of the system

Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t1		TOWEL $\begin{bmatrix} 0 \\ 1.86 \\ -1.7 \end{bmatrix}$	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00] most likely planstep=0	$\begin{bmatrix} 1.7 \\ 1.41 \\ -1.39 \end{bmatrix}$	"turn on water" $\begin{bmatrix} 1.82 \\ 0.22 \\ 0.47 \end{bmatrix}$	 "Hello I am so glad to have you here. Please turn on the water."

[1] Obtained using INTERACT. It is close to the EPA value of an identity of "elder".

[2] Obtained using INTERACT. It is close to the EPA value of an identity of a "lonesome elder".

Table 5.2: State changes in test #1 of the system

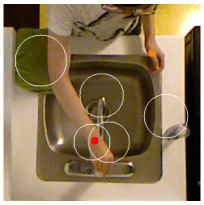
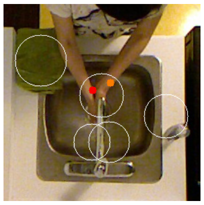
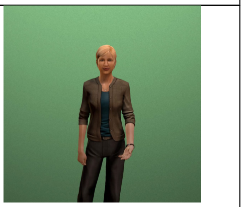
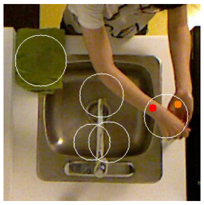
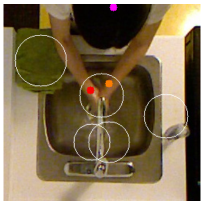
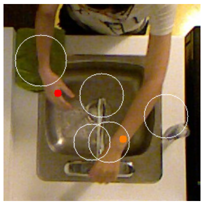
Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t2		TAP $\begin{bmatrix} 0 \\ 1.68 \\ -0.58 \end{bmatrix}$	[0.26, 0.74, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00] most likely planstep=1	$\begin{bmatrix} 2.73 \\ 1.14 \\ -1.03 \end{bmatrix}$	N/A	No Prompt
t3		RINSE $\begin{bmatrix} 0 \\ 1.49 \\ -0.16 \end{bmatrix}$	[0.27, 0.73, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00] most likely planstep=1	$\begin{bmatrix} 2.67 \\ 1.21 \\ -0.72 \end{bmatrix}$	“use some soap” $\begin{bmatrix} 1.51 \\ 0.12 \\ 0.52 \end{bmatrix}$	 “You are washing your hands. Please use the soap.”
t4		SOAP $\begin{bmatrix} 0 \\ 0.73 \\ -1.52 \end{bmatrix}$	[0.00, 0.01, 0.35, 0.64, 0.00, 0.00, 0.00, 0.00] most likely planstep=3	$\begin{bmatrix} 2.57 \\ 0.69 \\ -0.66 \end{bmatrix}$	N/A	No Prompt
t5		RINSE $\begin{bmatrix} 0 \\ 0.23 \\ -1.87 \end{bmatrix}$	[0.00, 0.00, 0.01, 0.02, 0.97, 0.00, 0.00, 0.00] most likely planstep=4	$\begin{bmatrix} 2.92 \\ 0.7 \\ -0.43 \end{bmatrix}$	N/A	No Prompt
t6		TAP $\begin{bmatrix} 0 \\ 1.79 \\ -1.84 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.00, 0.18, 0.00, 0.82, 0.00] most likely planstep=6	$\begin{bmatrix} 3.21 \\ 0.98 \\ -0.47 \end{bmatrix}$	N/A	No Prompt

Table 5.2: State changes in test #1 of the system

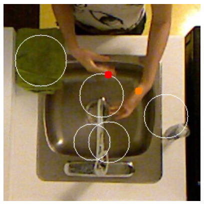
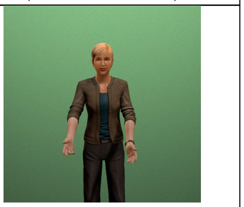
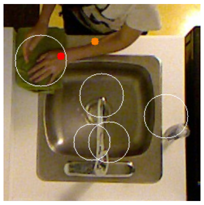
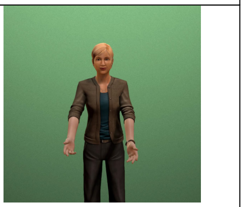
Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t7		RINSE $\begin{bmatrix}0\\1.69\\-1.6\end{bmatrix}$	[0.00, 0.00, 0.00, 0.00, 0.21, 0.00, 0.79, 0.00] most likely planstep=6	$\begin{bmatrix}3.27\\1.07\\-0.58\end{bmatrix}$	“use towel” $\begin{bmatrix}1.77\\0.16\\1\end{bmatrix}$	 “Can I get your hands dried up?”
t8		TOWEL $\begin{bmatrix}0\\1.08\\-1.16\end{bmatrix}$	[0.00, 0.00, 0.00, 0.00, 0.00, 0.14, 0.00, 0.86] most likely planstep=7	$\begin{bmatrix}3.31\\1.04\\-0.57\end{bmatrix}$	“all done” $\begin{bmatrix}1.55\\0.38\\0.87\end{bmatrix}$	 “Can you come back soon?”

Table 5.3: State changes in test #2 of the system

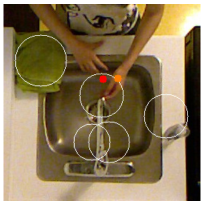
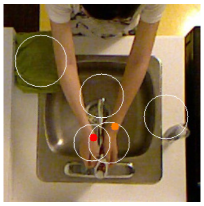
Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t1		RINSE $\begin{bmatrix}0\\0.29\\-1.86\end{bmatrix}$	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00] most likely planstep=0	$\begin{bmatrix}-0.87\\-0.28\\-2.11\end{bmatrix}$	“turn on water” $\begin{bmatrix}0.87\\0.85\\0.27\end{bmatrix}$	 “I want you to turn the water on.”
t2		TAP $\begin{bmatrix}0\\1.49\\-1.63\end{bmatrix}$	[0.46, 0.54, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00] most likely planstep=1	$\begin{bmatrix}-0.02\\-0.47\\-1.84\end{bmatrix}$	N/A	No Prompt

Table 5.3: State changes in test #2 of the system

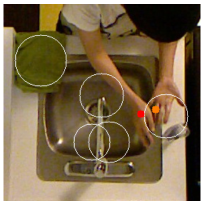
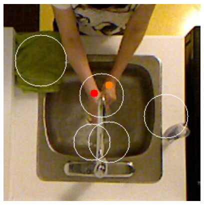
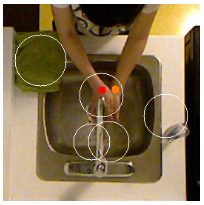
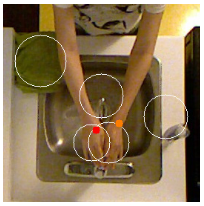
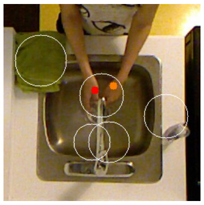
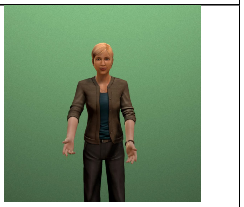
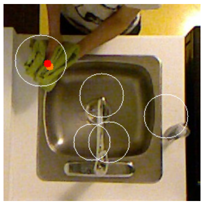
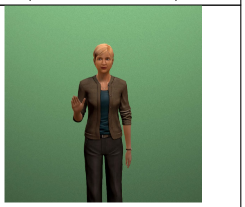
Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t3		SOAP $\begin{bmatrix} 0 \\ 1.25 \\ -1.74 \end{bmatrix}$	[0.02, 0.02, 0.24, 0.73, 0.00, 0.00, 0.00, 0.00] most likely planstep=3	$\begin{bmatrix} 1.2 \\ -0.37 \\ -1.46 \end{bmatrix}$	N/A	No Prompt
t4		RINSE $\begin{bmatrix} 0 \\ 0.05 \\ -1.85 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.00] most likely planstep=4	$\begin{bmatrix} 1.66 \\ -0.46 \\ -1.32 \end{bmatrix}$	N/A	No Prompt
t5		RINSE $\begin{bmatrix} 0 \\ 0.32 \\ -1.95 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.00] most likely planstep=4	$\begin{bmatrix} 1.61 \\ -0.45 \\ -1.33 \end{bmatrix}$	"turn off water" $\begin{bmatrix} 1.88 \\ 0.75 \\ 0.38 \end{bmatrix}$	 "Try turning off the water."
t6		TAP $\begin{bmatrix} 0 \\ 1.22 \\ -1.75 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.01, 0.18, 0.00, 0.81, 0.00] most likely planstep=6	$\begin{bmatrix} 1.84 \\ -0.48 \\ -1.25 \end{bmatrix}$	N/A	No Prompt
t7		RINSE $\begin{bmatrix} 0 \\ 1.03 \\ -1.73 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.01, 0.29, 0.00, 0.70, 0.00] most likely planstep=6	$\begin{bmatrix} 1.77 \\ -0.46 \\ -1.25 \end{bmatrix}$	"use towel" $\begin{bmatrix} 1.68 \\ 0.82 \\ -0.16 \end{bmatrix}$	 "Can I get your hands dried up?"

Table 5.3: State changes in test #2 of the system

Time	User Behav. (screenshot)	Behav. prop./epa	Planstep Belief	f_c	Prompt: prop./epa	Avatar (screenshot)
t8		TOWEL $\begin{bmatrix} 0 \\ 0.48 \\ -1.42 \end{bmatrix}$	[0.00, 0.00, 0.00, 0.00, 0.00, 0.09, 0.00, 0.91] most likely planstep=7	$\begin{bmatrix} 1.83 \\ -0.45 \\ -1.24 \end{bmatrix}$	"all done" $\begin{bmatrix} 1.5 \\ 0.6 \\ -0.03 \end{bmatrix}$	 "Good bye. Hope to see you soon."

5.3 Functionality Performance of the System

To make sure that the hand-washing system works well functionally, the following three aspects should be assured: (1) the system recognizes user-behaviours correctly; (2) it updates its beliefs of plansteps appropriately; (3) it gives out helpful prompts based on the belief states (i.e. the prompts should be propositionally useful). Recall that the first aspect depends on the performance of the Observer, and the last two aspects depend on that of the Planstep Updater. The column "user behaviour screenshot" in Tables 5.2 and 5.3 illustrates some screenshots of the system recognizing the actor's hands. As shown in the figures, the Observer extended from the original tracker is able to extract the positions of the user's hands when the user is turning on/off the tap, using soap, rinsing hands, and using towel. At each timestep, the Planstep Updater updates its planstep beliefs and computes a most likely planstep. Depending on the "most-likely planstep" computed, the Planstep Updater then decides (using the heuristic policy) on the propositional content of the system prompt at that time. For a certain most-likely planstep, the propositional content of the system prompt is deterministic. The policy based on which the propositional content of a system prompt is determined is shown in Table 4.2.

Table 5.2 shows how the system's planstep beliefs and prompts were changed according to the user's behaviours in test #1. At time t1, the actor was about to start washing her hands, with her right hand accidently being in the region of the towel. Observing this, the system thought the actor was using the towel, and concluded that the actor needed some instructions. The system suggested the user to turn on the water at t1. At time t2, the actor turned on the water. The system updated its planstep beliefs accordingly and got a most-likely planstep of 1. The system did not perform any prompts at t2, since it believed that the user had a high level of awareness (because she "followed" the system instructions) and was able to continue the handwashing task properly by herself. At time t3, the actor tried to rinse her hands. The system detected this behaviour of the actor and suggested for her to put on some soap first. Then, at times t4, t5, and t6, the system updated its planstep beliefs and did not perform any prompts when the actor put on some soap (at t4), rinsed her hands (at t5) and turned off the water (at t6). At t7, the actor was detected as rinsing her hands again, while in fact she was just moving her hands from the tap area to the towel area. Believing that the actor had a low awareness level and was in need of some assistance, the system suggested the user to proceed with using the towel. At time t8, the actor finally completed the handwashing task, and the system prompted an "all done" message to indicate the accomplishment.

Table 5.2 shows how the system's planstep beliefs and prompts were changed according to the user's behaviours in test #2. At time t1, the actor was detected as rinsing her hands while she was moving her hands towards the tap to turn on the water. The actor received an instruction suggesting for her to turn on the water from the system. At time t2, the actor "followed" the system's instruction and turned on the water. The system updated its planstep beliefs accordingly and got a most-likely planstep of 1. The system did not perform any prompts at t2, since it believed that the user had a high level of awareness and was able to continue the handwashing task properly by herself. At times t3 and t4, the actor put on some soap and started rinsing her hands. The system updated its planstep beliefs and did not perform any prompts. At t5, the actor continued rinsing her hands. Believing that the actor having been performing the same behaviour for too long a time, which implied a low awareness level of the actor, the system suggested the user to proceed with turning off the water. The actor followed the system's instruction and turned off the water at time t6. At time t7, the actor performed the behaviour of rinsing her hands again. Believing that the actor had a low awareness level and was in need of some assistance, the system instructed the user to use the towel. At time t8, the actor finally completed the handwashing task, and the system prompted an "all done" message to indicate the accomplishment.

As shown in Tables 5.2 and 5.3, though the system sometimes might false positively recognize a user behaviour (e.g. thinking the actor was using the towel at t1 in test #1), in general, it is able to produce propositionally useful system prompts in the two tests. The defection of falsely recognizing noise behaviours can be ameliorated by increasing the value of the parameter *timeup*, which is defined in the Buffer and is designed to handle (to some extent) behaviour noises. Readers can find state change details in a total of 17 tests, including the two shown in Tables 5.2 and 5.3 and 15 other runs, conducted on the system in Appendix A of this thesis.

5.4 Emotionality Performance of the System

To demonstrate that the system is able to work emotionally, this subsection compares the average of EPA values computed for user behaviours, for f_c's, and for system prompts in tests #1 and #2. Figure 5.2 shows a simple comparison of the EPA values in the two aforementioned tests.

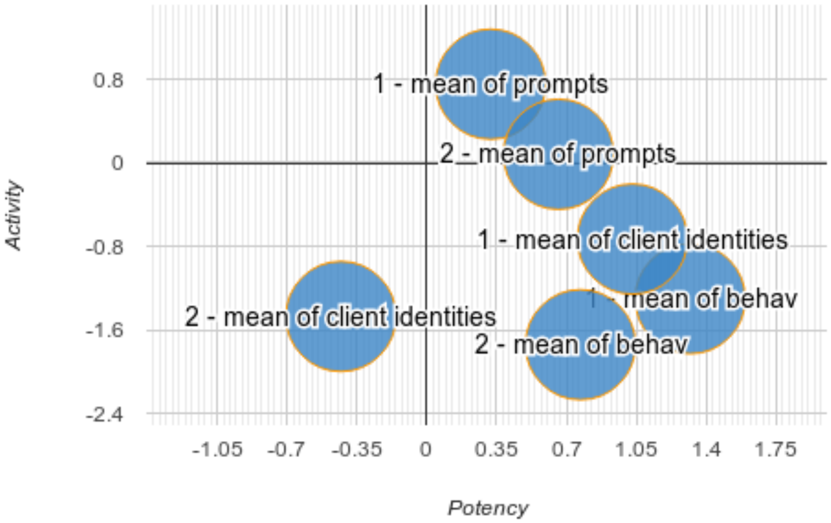


Figure 5.2: Compare P and A values in the two tests

As shown in Tables 5.2 and 5.3, throughout the tests, the user behaviours in the first test generally had larger P and larger A values than those in the second test. The P and A values computed for user behaviours in test #1 reached an average of $[1.32, -1.3]$, while that in the second test was $[0.77, -1.74]$. This phenomenon accords well with how the actor acted in the two tests, which illustrates that the P and A values of user behaviours computed by the EPA-Calculator are reasonable.

As also shown in Tables 5.2 and 5.3, throughout the tests, the f_c's in the first test generally had larger P and larger A values than those in the second test, and the system prompts in the first test generally had smaller P and higher A values. The mean of the EPA values of f_c's in the two tests were $[2.8, 1.03, -0.73]$ and $[1.13, -0.43, -1.47]$, respectively. And the mean of the EPA values of system prompts in the two tests were [1.62, 0.32, 0.75] and [1.53, 0.66, 0.08]. Note that prompts with lower P values and higher A values are produced for identities with higher P values and higher A values. This correlation makes sense since people who think of themselves as powerful persons tend to expect respect from others in interactions (i.e. prompts should be expressed to them with low potency levels), and that active people are likely to interact better with persons who are active as well — these “intuitions” are born out with BayesACT simulations, and thus are in accord with the predictions of Affect Control Theory.

To check if this correlation between the P and A values of identities and system prompts is just a coincidence, a total of 17 tests (including the two described above) of the system were conducted. Readers can find state change details for the 17 tests in Appendix A of this thesis. Among all the 17 tests, the actor performed more powerfully and more actively in 10 of them, and performed less powerfully and less actively in the other 7 runs. For all these tests, the parameter settings shown in Table 5.1 were used. We see a variability in results of the 17 tests; but generally, user behaviours with higher P and higher A values lead to: (1) client identities with higher P and higher A values, and (2) system prompts with lower P and higher A values.

To sum up, the test results showed that the system is able to compute reasonable P and A values for user behaviours, to update its beliefs of the user's identity based on behaviours performed by the user and itself, and to produce system prompts accordingly. The tests also indicated that user behaviours with higher P and higher A values may lead to f_c's with higher P and higher A values and system prompts with lower P and higher A values.

Chapter 6

Discussion

6.1 Contribution

Research in the area of emotional intelligence generally covers the following four aspects: (1) *recognition of affective states*, (2) *generation of affectively modulated signals*, (3) *psychological study of human emotions*, and (4) *computationally modelling affective HCIs.* While studies covering one or more of these aspects have been conducted, few assistive systems that have integrated all the four pieces together have been implemented. This thesis is one of the exploratory works in this area and proposed a solution to integrating emotional intelligence with a cognitive intelligent assistive system.

This thesis defined four objectives at the beginning and sought solutions to achieve it. It reviewed previous work in all the four aspects of emotional intelligence, and designed and implemented a prototypical hand-washing system aimed at assisting people with dementia to complete hand-washing tasks successfully. As an integration of independent components, the hand-washing system is extensible and portable. The hand-washing system is able to run in real-time from the perspective of the user group, and has been shown by laboratory tests that it is capable of providing a level of functional assistance and producing system prompts that have encoded to some extent the emotional state of the user.

Approaches that can be taken to recognize affective meanings of user behaviours in the application scenario, and the difficulties that lie in those approaches have been discussed in the thesis. The author pointed out that constrained by the specialty of the hand-washing scenario and the user group of this application, vision-based approaches focusing on facial expression analysis and acoustic-based approaches, which are the two most common approaches in affect recognition, are not necessarily the most suitable solutions for recognizing affective meanings of user behaviours for this particular application. The thesis finally took an initial threshold-based approach of recognizing affective meanings of user behaviours from hand movements — to be more specific, the expansiveness between user's hands and the velocity of the user moving his/her hands. The tracker that was used to locate the user's hands was designed and implemented as an extension to an existing human body tracker [16].

The affective reasoning during interactions are implemented in a reasoning engine where the belief state of the user's affective identities is updated basing upon BayesACT. The engine also maintains a belief state of how much the user has completed in the hand-washing process. Recommendation of prompts described in both functional (i.e. the content of an instructional message) and emotional (i.e. how the instructional message should be expressed) dimensions are produced by the engine based on the belief states and certain policies. The engine was designed and implemented on the basis of the existing BayesAct framework [24].

To enable the prompting system to display affectively modulated prompts to users, the thesis also reviewed techniques used in affective signal generation. Since dynamic generation of prompts is relatively difficult and requires much computational resources, the thesis took an approach of selecting the final prompt from a set of pre-generated and evaluated prompts. This thesis summarized that the four most essential questions involved in designing a prompt dataset and choosing a most appropriate one from it are: (1) Deciding the format of the prompts: should they be video, audio, or textual prompt? (2) Designing the prompts, e.g. the words used in the prompts. If the prompts are audio or video prompts, the tones how the messages are stated should be carefully designed as well. Character gestures and other details might require considerations as well if video prompts are used. (3) Labelling the generated prompts. (4) Selecting the prompt to display based on the propositional and emotional descriptions of recommended prompts produced by the reasoning engine. After these questions were discussed, the hand-washing system was designed and implemented in a way to select the final prompts displayed to users from a set of prompts generated and evaluated in previous work [38]. The final prompts were selected based on two conditions: (1) it should have same functional meanings as the recommended prompt by the engine, and (2) among all the prompts in the set that satisfy the first condition, it should have the smallest emotional distance, whose definition was defined in the thesis, to the recommended prompt.

Preliminary experiments where the system monitors an actor washing her hands and gives prompts were conducted. The results of two tests, where the actor behaved less powerfully and less actively in the second test than in the first, were described and compared in detail in the thesis. Fifteen further tests of the system were conducted as well. A simple comparison of the averages of EPA values for user behaviours, the user's identities and system prompts are provided in the thesis. Detailed state changes in those experiments are provided in the appendix of the thesis. The results of the tests showed that user behaviours are roughly recognized by the system, the EPA values computed for the user behaviours are reasonable. The tests also showed that the system is able to update its beliefs of planstep and emotional state of the user, and is able to produce accordingly system prompts both functionally and emotionally. The tests also indicated that user behaviours with higher P and higher A values are more likely to lead to identity beliefs for the user with higher P and higher A values and system prompts with lower P and higher A values. However, since the E value, which is an important component in representing sentiments, of user's behaviours were assumed to be neutral in our system, the correctness of this correlation still requires further investigations.

6.2 Future Work

This thesis is an initial work of integrating emotional intelligence with intelligent cognitive assistants. Our prototypical system is a first approach in this exploratory area and focuses on the integration work of combining different pieces of emotional intelligences with real-world functional assistive systems. The system may be improved in the following multiple directions:

1. Improving the EPA-Calculator

   Currently, the system computes P and A values of user behaviours based on the expansiveness of the user's hands and the velocity of the user's hand movements. The E value of user behaviours were assumed to be neutral in the system. Since E is an important component in representing sentiments, the validity of the correlation between sentiments of user behaviours, user identities and system prompts is limited. In the future, the EPA-Calculator can be improved to compute the A values for user behaviours as well. One possible approache to achieve this is to choose more sophisticated features/indicators to recognize the user's emotional states by cooperating with sociologists, psychologists, and physiologists, and studying the relationships between the behaviours of persons with AD and their emotion changes. If new features are chosen, new sensors and analyzers that obtain the selected features from observations might need to be added to the system .

Data collection and labelling is needed as well to implement a sophisticated EPA-Calculator. Data can be collected by recording videos of a person's hand movements while he/she is washing his/her hands. Ideally, the behaviours of persons with dementia should be recorded in order to achieve better performance for the calculator. The number of video recordings should be designed carefully. It is desirable that the recordings cover all possibilities that how a person's emotion changes during the handwashing process, though, unfortunately, it is infeasible to achieve this in reality due to the subjective nature of emotional experiences. After videos recording people's hand movements while they are washing their hands are collected, surveys should be conducted to have these labelled. Note that videos should be cut into shorter clips before they are labelled. This process is called data segmentation and is itself an open-ended problem. With proper segmentation, the user's emotional states (or the general emotional impressions formed by the clip) should remain stable throughout a video clip. For each short clip, a single EPA vector representing emotional impression the participant has on the clip is collected and is associated with each frame within the clip. The time length of the video clips should not be either too long nor too short. If the video clips are too long, more than one emotion are likely to be present in the same clip. If the video clips are too short, human raters might not be able to differentiate the emotional impressions presented by one clip from that presented by another. A solution that avoids the difficult data segmentation problem is to present the whole videos to participants, and let the participants split and assign new EPA values to clips when they feel a new emotional impression is formed. This approach is easier to implement, with the risks of people partitioning the videos differently and unreasonably. Other aspects of the survey, such as participant eligibility for the survey, should be designed carefully as well. If the data collected and labelled is not enough, which often happens in medical research, statistical methods, such as bootstrapping, can be applied.

Even though labelling was conducted on video clips, the labels are applied to frames within the videos, and models mapping from features extracted from the frames to the labels are trained. This is to avoid the heavy computational burden caused by learning directly from videos. Note that several frames are produced from a single video clip, and not all of them need to be included in the training process. Decisions should be made on when, how many, and what frames to cut from the video clips for training. The shorter the time period between two neighbour frames, the more accurate the result would be. However, it would require too many computational resources if neighbouring frames are too close to each other. If the frames are grabbed at a frequency higher than that at which frames are processed by the classifier, loss of data would occur as well.

2. Improving the prompt generation process

   Currently, the system selected final prompts displayed to users from a set of 30 audio-visual prompts generated and evaluated in Malhotra's work [38]. The prompts were evaluated in terms of EPA values in a survey where participants were normal healthy persons. More prompts with different formats and contents can be created, rated and added to the prompt dataset in the future. Moreover, if the system were to be evaluated in clinical trials (as opposed to in the laboratory environment), the prompt dataset should be rated by persons with AD rather than normal healthy persons. Approaches that can dynamically generate prompts could be tried as well in future works.

3. Improving the Planstep- and Emotion-Updater

   Currently, the belief states updater of the system is implemented as a POMDP. It assumes that the user and the system take turns in the interactions. Future work can improve the system by breaking through the turn-taking limits. The user's identity is learned by a BayesACT model in the current approach. Further investigation of identities in Alzheimer's disease and how BayesACT can be used to provide more effective prompting can be taken in future work as well.

4. Conducting clinical trials for the system

   With training and testing data collected from clinical trials, and data labelled and prompts evaluated by persons with AD, preliminary tests of the system in clinical environments can be conducted in future works.

Being designed and implemented as an integration of independent components, the system is extensible and portable, which makes it possible to improve one or more of the components with a few, if any, minor changes to other components.

References

[1] Nalini Ambady and Robert Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological bulletin*, 111(2):256, 1992.

[2] Theologos Athanaselis, Stelios Bakamidis, Ioannis Dologlou, Roddy Cowie, Ellen Douglas-Cowie, and Cate Cox. Asr for emotional speech: clarifying the issues and enhancing performance. *Neural Networks*, 18(4):437–444, 2005.

[3] Marian Stewart Bartlett, Gwen Littlewort, Mark Frank, Claudia Lainscsek, Ian Fasel, and Javier Movellan. Recognizing facial expression: machine learning and application to spontaneous behavior. In *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, volume 2, pages 568–573. IEEE, 2005.

[4] Anton Batliner, Kerstin Fischer, Richard Huber, Jörg Spilker, and Elmar Nöth. How to find trouble in communication. *Speech communication*, 40(1):117–143, 2003.

[5] Aryel Beck, Antoine Hiolle, Alexandre Mazel, and Lola Cañamero. Interpretation of emotional body language displayed by robots. In *Proceedings of the 3rd international workshop on affective interaction in natural environments*, pages 37–42. ACM, 2010.

[6] Ray L Birdwhistell. *Kinesics and context: Essays on body motion communication.* University of Pennsylvania press, 2011.

[7] Jennifer Boger, Pascal Poupart, Jesse Hoey, Craig Boutilier, Geoff Fernie, and Alex Mihailidis. A decision-theoretic approach to task assistance for persons with dementia. In *IJCAI*, pages 1293–1299. Citeseer, 2005.

[8] Hana Boukricha, Ipke Wachsmuth, A Hofstatter, and Karl Grammer. Pleasure-arousal-dominance driven facial expression simulation. In *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009. 3rd International Conference on*, pages 1–7. IEEE, 2009.

[9] Andrew E Budson and Paul R Solomon. *Memory loss: A practical guide for clinicians.* Elsevier Health Sciences, 2011.

[10] T Duy Bui. Creating emotions and facial expressions for embodied agents. 2004.

[11] Justine Cassell. *Embodied conversational agents.* MIT press, 2000.

[12] Ginevra Castellano, Santiago D Villalba, and Antonio Camurri. Recognising human emotions from body movement and gesture dynamics. In *Affective computing and intelligent interaction*, pages 71–82. Springer, 2007.

[13] Wen-Sheng Chu, Fernando De La Torre, and Jeffery F Cohn. Selective transfer machine for personalized facial action unit detection. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 3515–3522. IEEE, 2013.

[14] Cristina Conati and Heather Maclaren. Empirically building and evaluating a probabilistic model of user affect. *User Modeling and User-Adapted Interaction*, 19(3):267–303, 2009.

[15] Mark Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, and viewpoint dependence. *Journal of nonverbal behavior*, 28(2):117–139, 2004.

[16] Stephen Czarnuch and Alex Mihailidis. Depth image hand tracking from an overhead perspective using partially labeled, unbalanced data: Development and real-world testing. *under review*, 2014.

[17] Sidney D'Mello and Art Graesser. Automatic detection of learner's affect from gross body language. *Applied Artificial Intelligence*, 23(2):123–150, 2009.

[18] Paul Ekman. Are there basic emotions? 1992.

[19] Magy Seif El-Nasr, John Yen, and Thomas R Ioerger. Flamefuzzy logic adaptive model of emotions. *Autonomous Agents and Multi-agent systems*, 3(3):219–257, 2000.

[20] Johnny RJ Fontaine, Klaus R Scherer, Etienne B Roesch, and Phoebe C Ellsworth. The world of emotions is not two-dimensional. *Psychological science*, 18(12):1050–1057, 2007.

[21] Joseph C Hager, Paul Ekman, and Wallace V Friesen. Facial action coding system. *Salt Lake City, UT: A Human Face*, 2002.

[22] Jesse Hoey and Pascal Poupart. Solving pomdps with continuous or large discrete observation spaces. In *IJCAI*, pages 1332–1338, 2005.

[23] Jesse Hoey, Pascal Poupart, Axel von Bertoldi, Tammy Craig, Craig Boutilier, and Alex Mihailidis. Automated handwashing assistance for persons with dementia using video and a partially observable Markov decision process. *Computer Vision and Image Understanding*, 114(5):503–519, 2010.

[24] Jesse Hoey, Tobias Schroder, and Areej Alhothali. Bayesian affect control theory. In *Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on*, pages 166–172. IEEE, 2013.

[25] Jesse Hoey, A von Bertoldi, P Poupart, and A Mihailidis. Tracking using flocks of features, with application to assisted handwashing. In *British Machine Vision Conference*, pages 367–376, 2006.

[26] Jesse Hoey, Xiao Yang, Eduardo Quintana, and Jesús Favela. Lacasa: Location and context-aware safety assistant. In *Pervasive Computing Technologies for Healthcare (PervasiveHealth), 2012 6th International Conference on*, pages 171–174. IEEE, 2012.

[27] Michael Isard and Andrew Blake. A mixed-state condensation tracker with automatic model-switching. In *Computer Vision, 1998. Sixth International Conference on*, pages 107–112. IEEE, 1998.

[28] Michelle Karg, Kolja Kuhnlenz, and Martin Buss. Recognition of affect based on gait patterns. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 40(4):1050–1061, 2010.

[29] Michelle Karg, A Samadani, Rob Gorbet, K Kuhnlenz, Jesse Hoey, and Dana Kulic. Body movements for affective expression: A survey of automatic recognition and generation. 2013.

[30] Jeffrey A Kaye, Shoshana A Maxwell, Nora Mattek, Tamara L Hayes, Hiroko Dodge, Misha Pavel, Holly B Jimison, Katherine Wild, Linda Boise, and Tracy A Zitzelberger. Intelligent systems for assessing aging changes: home-based, unobtrusive, and continuous assessment of aging. *The Journals of Gerontology Series B: Psychological Sciences and Social Sciences*, 66(suppl 1):i180–i190, 2011.

[31] Isaac V Kerlow. *The art of 3D: computer animation and effects*. John Wiley & Sons, 2004.

[32] Andrea Kleinsmith and Nadia Bianchi-Berthouze. Affective body expression perception and recognition: A survey. *Affective Computing, IEEE Transactions on*, 4(1):15–33, 2013.

[33] Oh-Wook Kwon, Kwokleung Chan, Jiucang Hao, and Te-Won Lee. Emotion recognition by speech signals. In *INTERSPEECH*, 2003.

[34] John Lasseter. Principles of traditional animation applied to 3d computer animation. In *ACM Siggraph Computer Graphics*, volume 21, pages 35–44. ACM, 1987.

[35] Chul Min Lee and Shrikanth S Narayanan. Toward detecting emotions in spoken dialogs. *Speech and Audio Processing, IEEE Transactions on*, 13(2):293–303, 2005.

[36] Michael Lewis, Jeannette M Haviland-Jones, and Lisa Feldman Barrett. *Handbook of emotions*. Guilford Press, 2010.

[37] Kathryn J Lively and Carrie L Smith. Identity and illness. In *Handbook of the Sociology of Health, Illness, and Healing*, pages 505–525. Springer, 2011.

[38] Aarti Malhotra, Celia Yu, Tobias Schröder, and Jesse Hoey. An exploratory study into the use of an emotionally aware cognitive assistant. *under review*, 2014.

[39] Iris B Mauss and Michael D Robinson. Measures of emotion: A review. *Cognition and emotion*, 23(2):209–237, 2009.

[40] Alex Mihailidis, Jennifer N Boger, Tammy Craig, and Jesse Hoey. The coach prompting system to assist older adults with dementia through handwashing: An efficacy study. *BMC Geriatrics*, 8(1):28, 2008.

[41] Alex Mihailidis, Brent Carmichael, and Jennifer Boger. The use of computer vision in an intelligent environment to support aging-in-place, safety, and independence in the home. *Information Technology in Biomedicine, IEEE Transactions on*, 8(3):238–247, 2004.

[42] George E Monahan. State of the art — a survey of partially observable Markov decision processes: Theory, Models, and Algorithms. *Management Science*, 28(1):1–16, 1982.

[43] Mihalis A Nicolaou, Hatice Gunes, and Maja Pantic. Continuous prediction of spontaneous affect from multiple cues and modalities in valence-arousal space. *Affective Computing, IEEE Transactions on*, 2(2):92–105, 2011.

[44] Radosław Niewiadomski, Sylwia Julia Hyniewska, and Catherine Pelachaud. Computational models of expressive behaviors for a virtual agent. *Social Emotions in Nature and Artifact*, page 143, 2013.

[45] Kenji Okuma, Ali Taleghani, Nando De Freitas, James J Little, and David G Lowe. A boosted particle filter: Multitarget detection and tracking. In *Computer Vision-ECCV 2004*, pages 28–39. Springer, 2004.

[46] Celia J Orona. Temporality and identity loss due to alzheimer's disease. *Social Science & Medicine*, 30(11):1247–1256, 1990.

[47] Andrew Ortony. *The cognitive structure of emotions.* Cambridge university press, 1990.

[48] Charles Egerton Osgood. *Cross-cultural universals of affective meaning.* University of Illinois Press, 1975.

[49] Maja Pantic and Leon JM Rothkrantz. Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE*, 91(9):1370–1390, 2003.

[50] Maja Pantic, Michel Valstar, Ron Rademaker, and Ludo Maat. Web-based database for facial expression analysis. In *Multimedia and Expo, 2005. ICME 2005. IEEE International Conference on*, pages 5–pp. IEEE, 2005.

[51] Christian Peters, Thomas Hermann, Sven Wachsmuth, and Jesse Hoey. Automatic task assistance for people with cognitive disabilities in brushing teeth—a user study with the tebra system. *ACM Transactions on Accessible Computing (TACCESS)*, 5(4):10, 2014.

[52] Matthai Philipose, Kenneth P Fishkin, Mike Perkowitz, Donald J Patterson, Dieter Fox, Henry Kautz, and Dirk Hahnel. Inferring activities from interactions with objects. *Pervasive Computing, IEEE*, 3(4):50–57, 2004.

[53] Rosalind W Picard. *Affective computing.* MIT press, 2000.

[54] Martha E Pollack. Intelligent technology for an aging population: The use of AI to assist elders with cognitive impairment. *AI magazine*, 26(2):9, 2005.

[55] Pascal Poupart. An introduction to fully and partially observable Markov decision processes. *Decision Theory Models for Applications in Artificial Intelligence: Concepts and Solutions, IGI Global*, pages 33–62, 2011.

[56] David V Pynadath and Stacy C Marsella. Psychsim: Modeling theory of mind with decision-theoretic agents. In *IJCAI*, volume 5, pages 1181–1186, 2005.

[57] Dawn T Robinson, Lynn Smith-Lovin, and Allison K Wisecup. *Affect control theory.* Springer, 2006.

[58] Donna Rose Addis and Lynette Tippett. Memory of myself: Autobiographical memory and identity in alzheimer's disease. *Memory*, 12(1):56–74, 2004.

[59] Klaus R Scherer. What are emotions? and how can they be measured? *Social Science Information*, 44(4):695–729, 2005.

[60] Wolfgang Scholl. The socio-emotional basis of human interaction and communication: How we construct our social world. *Social Science Information*, 52(1):3–33, 2013.

[61] Tobias Schröder, Janine Netzel, Carsten C Schermuly, and Wolfgang Scholl. Culture-constrained affective consistency of interpersonal behavior: A test of affect control theory with nonverbal expressions. *Social Psychology*, 44(1):47, 2013.

[62] Jamie Shotton, Toby Sharp, Alex Kipman, Andrew Fitzgibbon, Mark Finocchio, Andrew Blake, Mat Cook, and Richard Moore. Real-time human pose recognition in parts from single depth images. *Communications of the ACM*, 56(1):116–124, 2013.

[63] Lucy Suchman. *Human-machine reconfigurations: Plans and situated actions.* Cambridge University Press, 2007.

[64] Jianhua Tao and Tieniu Tan. Affective computing: A review. In *Affective Computing and Intelligent Interaction*, pages 981–995. Springer, 2005.

[65] Lijun Yin, Xiaozhou Wei, Yi Sun, Jun Wang, and Matthew J Rosato. A 3d facial expression database for facial behavior research. In *Automatic face and gesture recognition, 2006. FGR 2006. 7th international conference on*, pages 211–216. IEEE, 2006.

[66] Zhihong Zeng, Maja Pantic, Glenn I Roisman, and Thomas S Huang. A survey of affect recognition methods: Audio, visual, and spontaneous expressions. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 31(1):39–58, 2009.

APPENDIX

Appendix A

Detailed Test Results

This appendix shows the EPA values rated by participants for the thirty prompts in the prompt dataset. It also shows detailed state changes of the 17 tests conducted on the system. The columns "time" in the tables indicate timesteps in the Planstep and Emotion Updater, where "time = 0" means at the time of initialization. The columns "behav-prop." in the tables use the definition of behaviours shown in column "Representation Number used in the Updater" in Table 4.3, and the columns of "ps_beliefs" use the definition of plansteps shown in Table 4.2. Finally, the columns "prompt-prop." in the tables use the definition of prompts shown in column "Representation number in Prompt-Selector" in Table 4.1. If "prompt-epa = 0", then it means that there was no actual prompt displayed at that timestep. No matter whether a prompt was displayed or not at a timestep, the value of "prompt-epa" at that timestep was produced by the updater based on the belief states at that time, and thus is included in calculating the mean of prompt-epa's for that run. For all these tests, the parameter settings as shown in Table 5.1 were used.

Table A.1: EPA values rated to prompts

#	EPA and type (expectations)	Sentence said by avatar	Purpose	EPA values (rated)
1	EPA+++_Discipline	Hi there, good to see you. Let's get started. Try turning on the water.	turn on water	[2.50, 1.78, 1.11]

2	EPA+–_Request	Hello, I am so glad to have you here. Please turn on the water.	turn on water	$[2.50, 1.06, 1.00]$
3	EPA-++_Bossy	Hi. Let's start washing your hands. Turn on the water.	turn on water	$[-0.28, 1.33, 1.06]$
4	EPA—_Bum	Hey. Came to wash your hands. Turn on the water if you want.	turn on water	$[-2.28, -0.22, 0.06]$
5	EPA+-+_Childlike	I want you to turn the water on.	turn on water	$[-0.22, 0.94, 0.89]$
6	EPA+++_Discipline	Try putting on some soap.	put on some soap	$[1.56, 1.17, 1.35]$
7	EPA+–_Request	You are washing your hands. Please use the soap.	put on some soap	$[1.22, 0.89, 0.72]$
8	EPA-++_Bossy	Now use the soap.	put on some soap	$[-1.50, 1.67, 1.44]$
9	EPA—_Bum	If you want to put on some soap, there is a soap pump lying around.	put on some soap	$[-0.94, -0.67, 0.00]$
10	EPA+-+_Childlike	I want you to put on some soap.	put on some soap	$[0.39, 1.78, 1.50]$
11	EPA+++_Discipline	Try rinsing your hands.	rinse hands	$[0.44, 0.33, 1.28]$
12	EPA+–_Request	Please rinse your hands.	rinse hands	$[2.28, 1.33, 1.11]$
13	EPA-++_Bossy	Rinse your hands now.	rinse hands	$[-1.83, 1.83, 2.00]$
14	EPA—_Bum	Rinse your hands if you want.	rinse hands	$[-1.50, -1.56, -0.11]$
15	EPA+-+_Childlike	Can I get your hands rinsed?	rinse hands	$[1.22, 0.56, 1.11]$
16	EPA+++_Discipline	Try turning off the water.	turn off water	$[1.11, 0.94, 0.89]$
17	EPA+–_Request	Please turn the water off. Thank you.	turn off water	$[2.50, 1.22, 0.94]$

18	EPA-++_Bossy	Will you turn off the water now?	turn off water	$[-1.94, 1.56, 1.67]$
19	EPA—_Bum	Turn the water off when done.	turn off water	$[-0.17, 1.28, 0.72]$
20	EPA+-+_Childlike	I want you to turn off the water.	turn off water	$[0.24, 1.47, 1.29]$
21	EPA+++_Discipline	Good job. Try using the towel to dry your hands.	dry up hands	$[2.28, 1.22, 1.33]$
22	EPA+–_Request	You are doing great. Please dry your hands using the towel.	dry up hands	$[2.89, 1.56, 0.50]$
23	EPA-++_Bossy	Now dry your hands.	dry up hands	$[-1.44, 1.61, 1.28]$
24	EPA—_Bum	There is a towel somewhere to dry your hands.	dry up hands	$[-1.39, -1.61, -0.44]$
25	EPA+-+_Childlike	Can I get your hands dried up?	dry up hands	$[1.44, 1.11, 1.17]$
26	EPA+++_Discipline	Goodbye. Hope to see you soon.	indicating all is done	$[2.11, 0.44, 0.11]$
27	EPA+–_Request	Please come back. I shall wait for you.	indicating all is done	$[1.50, -0.11, 0.11]$
28	EPA-++_Bossy	You are done. Leave now.	indicating all is done	$[-2.67, 1.94, 1.44]$
29	EPA—_Bum	Will see you whatever.	indicating all is done	$[-2.67, -0.83, 0.22]$
30	EPA+-+_Childlike	Can you come back soon?	indicating all is done	$[1.06, 0.12, 0.47]$

Table A.2: Experiment results of run 1

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.59, 0.83, -0.87]		
1	4	[0, 3.71, -0.84]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.51, 1.96, -1.14]	1	[1.84, 0.55, 0.41]
2	2	[0, 0.84, -0.34]	1	[0.27, 0.73, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.70, 0.92, -0.49]	0	[1.76, 0.62, 0.57]
3	4	[0, 1.59, -0.09]	1	[0.29, 0.71, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.74, 1.07, -0.48]	2	[1.77, 0.36, 0.39]
4	1	[0, 1.63, 1.78]	3	[0.00, 0.01, 0.11, 0.88, 0.00, 0.00, 0.00, 0.00]	[1.59, 0.99, -0.06]	0	[1.9, 0.47, 0.25]
5	3	[0, 1.02, -1.63]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[1.57, 0.97, 0.05]	0	[1.94, 0.45, 0.35]
6	2	[0, 1.82, -1.47]	6	[0.00, 0.00, 0.00, 0.00, 0.26, 0.00, 0.74, 0.00]	[1.61, 0.97, 0.03]	0	[2.05, 0.45, 0.56]
7	4	[0, 1.99, -1.57]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.05, 0.01, 0.94]	[1.62, 1.00, 0.04]	6	[1.96, 0.95, 0.37]
average		[0, 1.8, -0.59]			[1.62, 1.12, -0.29]		[1.89, 0.55, 0.41]

Table A.3: Experiment results of run 2

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.58, 0.81, -0.94]		
1	4	[0, 1.9, -0.21]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.16, 1.42, -0.85]	1	[1.58, 0.24, 0.01]
2	2	[0, 0.91, -0.31]	1	[0.28, 0.72, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.29, 0.96, -0.25]	0	[1.79, 0.54, 0.59]
3	4	[0, 1.14, -0.53]	1	[0.22, 0.78, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.38, 0.91, -0.35]	2	[1.85, 0.33, 0.72]
4	1	[0, 1.15, 1.29]	3	[0.00, 0.00, 0.15, 0.84, 0.00, 0.00, 0.00, 0.00]	[2.54, 0.96, 0.05]	0	[1.67, 0.41, 0.37]
5	3	[0, 0.61, -1.73]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[2.46, 0.68, -0.38]	0	[1.77, 0.54, 0.81]
6	2	[0, 1.9, -1.53]	6	[0.00, 0.00, 0.00, 0.00, 0.26, 0.00, 0.74, 0.00]	[2.49, 0.60, -0.57]	0	[1.69, 0.59, 0.71]
7	4	[0, 1.96, -1.75]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.09, 0.01, 0.89]	[2.75, 0.57, -0.78]	6	[1.64, 0.42, 0.53]
average		[0, 1.37, -0.68]			[2.30, 0.87, -0.45]		[1.71, 0.44, 0.53]

Table A.4: Experiment results of run 3

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.52, 0.85, -0.9]		
1	2	[0, 1.89, -0.81]	1	[0.25, 0.75, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.49, 1.17, -1.66]	0	[1.75, 0.3, 0.44]
2	4	[0, 1.69, -1.40]	1	[0.23, 0.77, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.50, 1.18, -1.39]	2	[2.01, 0.99, 0.82]
3	2	[0, 1.54, -0.52]	1	[0.20, 0.80, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.60, 1.22, -1.15]	2	[2, 0.76, 0.81]
4	3	[0, 1.69, -1.31]	1	[0.13, 0.87, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.62, 1.26, -1.14]	2	[1.89, 0.38, 0.45]
5	1	[0, 0.7, -1.55]	3	[0.00, 0.02, 0.00, 0.98, 0.00, 0.00, 0.00, 0.00]	[1.29, 1.11, -0.79]	0	[2.04, 0.74, 0.42]
6	3	[0, 0.25, -1.91]	4	[0.00, 0.00, 0.00, 0.03, 0.97, 0.00, 0.00, 0.00]	[0.77, 0.94, -0.54]	0	[1.84, 0.78, 0.11]
7	3	[0, 0.62, -1.98]	4	[0.00, 0.00, 0.00, 0.05, 0.95, 0.00, 0.00, 0.00]	[0.77, 0.93, -0.55]	4	[2.18, 0.67, -0.01]
8	2	[0, 1.61, -0.48]	6	[0.00, 0.00, 0.00, 0.00, 0.20, 0.00, 0.80, 0.00]	[0.51, 0.89, -0.42]	0	[1.77, 0.89, 0.1]
9	4	[0, 1.79, -1.18]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.15, 0.01, 0.84]	[0.43, 0.87, -0.42]	6	[1.69, 0.7, -0.05]
10	4	[0, 0.36, -1.25]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.12, 0.02, 0.86]	[0.35, 0.83, -0.42]	6	[1.81, 0.61, 0.73]
average		[0, 1.89, -0.81]			[1.51, 1.01, -1.28]		[1.75, 0.3, 0.44]

Table A.5: Experiment results of run 4

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.54, 0.78, -0.89]		
1	2	[0, 0.7, -0.61]	1	[0.28, 0.72, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.84, 0.00, -0.86]	0	[1.48, 0.38, 0.44]
2	4	[0, 1.16, -0.64]	1	[0.23, 0.77, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.77, 0.23, -0.71]	2	[1.63, 0.35, 0.73]
3	1	[0, 1.17, 1.01]	3	[0.00, 0.01, 0.09, 0.90, 0.00, 0.00, 0.00, 0.00]	[2.95, 0.48, -0.08]	0	[1.56, 0.51, 0.86]
4	3	[0, 0.95, -1.68]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[3.39, 0.59, -0.24]	0	[1.43, 0.48, 0.57]
5	2	[0, 1.81, -1.60]	6	[0.00, 0.00, 0.00, 0.00, 0.16, 0.00, 0.84, 0.00]	[3.56, 0.77, -0.21]	0	[1.23, 0.25, 0.65]
6	4	[0, 2.36, -1.60]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.03, 0.01, 0.96]	[3.78, 0.96, -0.14]	6	[1.45, 0.44, 0.99]
average		[0, 1.36, -0.85]			[3.21, 0.51, -0.38]		[1.46, 0.4, 0.71]

Table A.6: Experiment results of run 5

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.59, 0.80, -0.87]		
1	4	[0, 1.14, -1.13]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.57, 0.89, -1.05]	1	[1.99, 0.5, 0.41]
2	2	[0, 1.29, 0.04]	1	[0.34, 0.66, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.98, 0.88, -0.46]	0	[1.49, 0.33, 0.39]
3	3	[0, 1.43, 0.44]	1	[0.41, 0.59, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[3.07, 0.83, -0.27]	2	[1.58, 0.52, 0.83]
4	1	[0, 0.86, -1.41]	2	[0.00, 0.00, 0.53, 0.46, 0.00, 0.00, 0.00, 0.00]	[3.15, 0.78, -0.19]	0	[1.67, 0.28, 0.83]
5	3	[0, 0.2, -1.88]	4	[0.00, 0.00, 0.01, 0.01, 0.98, 0.00, 0.00, 0.00]	[2.65, 0.38, -0.56]	0	[1.62, 0.26, 0.81]
6	2	[0, 1.78, -1.80]	6	[0.00, 0.00, 0.00, 0.00, 0.38, 0.00, 0.62, 0.00]	[2.46, 0.39, -0.61]	0	[1.77, 0.57, 0.93]
7	3	[0, 1.21, -1.60]	6	[0.00, 0.00, 0.00, 0.00, 0.42, 0.00, 0.58, 0.00]	[2.45, 0.42, -0.64]	5	[1.62, 0.75, 0.49]
8	4	[0, 1.11, -1.07]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.23, 0.00, 0.77]	[2.45, 0.39, -0.66]	6	[1.87, 0.63, 0.84]
average		[0, 1.13, -1.05]			[2.60, 0.62, -0.55]		[1.7, 0.48, 0.69]

Table A.7: Experiment results of run 6

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.71, 0.84, -0.89]		
1	4	[0, 1.39, -1.40]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.44, 0.98, -1.21]	1	[1.96, 0.61, 0.43]
2	2	[0, 1.35, -0.38]	1	[0.24, 0.76, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.13, 0.63, -0.68]	0	[1.71, 0.9, 0.91]
3	3	[0, 1.36, 0.40]	1	[0.23, 0.77, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.10, 0.54, -0.42]	2	[1.74, 0.49, 0.43]
4	1	[0, 0.55, -1.47]	3	[0.00, 0.01, 0.33, 0.66, 0.00, 0.00, 0.00, 0.00]	[2.25, 0.42, -0.49]	0	[1.46, 0.27, 0.24]
5	3	[0, 0.17, -1.83]	4	[0.00, 0.00, 0.01, 0.01, 0.98, 0.00, 0.00, 0.00]	[2.15, 0.31, -0.58]	0	[1.9, 0.56, 0.84]
6	2	[0, 1.69, -1.72]	6	[0.00, 0.00, 0.00, 0.00, 0.16, 0.00, 0.84, 0.00]	[2.30, 0.39, -0.58]	0	[1.76, 0.48, 0.7]
7	3	[0, 1.67, -1.57]	6	[0.00, 0.00, 0.00, 0.00, 0.18, 0.00, 0.82, 0.00]	[2.29, 0.45, -0.58]	5	[1.93, 0.48, 0.7]
8	4	[0, 1.07, -1.10]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.98, 0.02, 0.00]	[2.28, 0.49, -0.55]	6	[1.75, 0.7, 0.96]
average		[0, 1.16, -1.13]			[2.12, 0.53, -0.63]		[1.78, 0.56, 0.65]

Table A.8: Experiment results of run 7

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.61, 0.84, -0.87]		
1	4	[0, 1.86, -1.70]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.70, 1.41, -1.39]	1	[1.82, 0.22, 0.47]
2	2	[0, 1.68, -0.58]	1	[0.26, 0.74, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.73, 1.14, -1.03]	0	[1.59, 0.15, 0.5]
3	3	[0, 1.49, -0.16]	1	[0.27, 0.73, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.67, 1.21, -0.72]	2	[1.51, 0.12, 0.52]
4	1	[0, 0.73, -1.52]	3	[0.00, 0.01, 0.35, 0.64, 0.00, 0.00, 0.00, 0.00]	[2.57, 0.69, -0.66]	0	[1.7, 0.85, 1.07]
5	3	[0, 0.23, -1.87]	4	[0.00, 0.00, 0.01, 0.02, 0.97, 0.00, 0.00, 0.00]	[2.92, 0.70, -0.43]	0	[1.64, 0.26, 0.87]
6	2	[0, 1.79, -1.84]	6	[0.00, 0.00, 0.00, 0.00, 0.18, 0.00, 0.82, 0.00]	[3.21, 0.98, -0.47]	0	[1.4, 0.43, 0.73]
7	3	[0, 1.69, -1.60]	6	[0.00, 0.00, 0.00, 0.00, 0.21, 0.00, 0.79, 0.00]	[3.27, 1.07, -0.58]	5	[1.77, 0.16, 1]
8	4	[0, 1.08, -1.16]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.14, 0.00, 0.86]	[3.31, 1.04, -0.57]	6	[1.55, 0.38, 0.87]
average		[0, 1.32, -1.30]			[2.80, 1.03, -0.73]		[1.62, 0.32, 0.75]

Table A.9: Experiment results of run 8

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.56, 0.79, -0.89]		
1	2	[0, 3.05, -0.89]	1	[0.30, 0.70, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.33, 1.69, 0.22]	0	[2.04, 0.16, 0.58]
2	1	[0, 1.46, -1.54]	3	[0.01, 0.03, 0.12, 0.84, 0.00, 0.00, 0.00, 0.00]	[2.84, 1.07, 0.01]	0	[1.82, 0.5, 0.71]
3	3	[0, 0.28, -1.71]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[2.72, 0.63, -1.36]	0	[1.65, 0.62, 0.74]
4	3	[0, 0.49, -1.87]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[2.68, 0.55, -1.45]	4	[1.7, 0.39, 0.53]
5	2	[0, 0.71, -0.83]	6	[0.00, 0.00, 0.00, 0.00, 0.26, 0.00, 0.74, 0.00]	[1.97, 0.61, -1.34]	0	[1.74, 0.61, 0.5]
6	4	[0, 1.75, -1.30]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.12, 0.01, 0.86]	[1.82, 0.68, -1.25]	6	[1.69, 0.39, 0.2]
average		[0, 1.29, -1.36]			[2.39, 0.87, -0.86]		[1.77, 0.44, 0.54]

Table A.10: Experiment results of run 9

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.61, 0.83, -0.9]		
1	2	[0, 2.02, -0.83]	1	[0.17, 0.83, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.36, 1.23, -1]	0	[1.88, 0.54, 0.55]
2	1	[0, 1.56, -1.63]	3	[0.00, 0.01, 0.07, 0.92, 0.00, 0.00, 0.00, 0.00]	[2.32, 0.99, -1.18]	0	[1.58, 0.63, 0.7]
3	3	[0, 0.32, -1.74]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[2.31, 0.52, -1.12]	0	[1.66, 0.6, 0.54]
4	3	[0, 0.52, -1.95]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[2.16, 0.68, -1.19]	4	[1.88, 0.49, 0.58]
5	2	[0, 0.86, -0.63]	6	[0.00, 0.00, 0.00, 0.00, 0.14, 0.00, 0.86, 0.00]	[2.32, 0.38, -1.03]	0	[1.73, 0.47, 0.74]
6	4	[0, 1.64, -1.39]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.11, 0.01, 0.88]	[2.35, 0.40, -0.97]	6	[1.66, 0.68, 0.65]
average		[0, 1.15, -1.36]			[2.30, 0.70, -1.08]		[1.73, 0.57, 0.63]

Table A.11: Experiment results of run 10

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[1.51, 0.82, -0.89]		
1	2	[0, 1.96, -0.61]	1	[0.25, 0.75, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.38, 0.97, -0.11]	0	[1.46, 0.33, 0.33]
2	1	[0, 1.46, -1.59]	3	[0.00, 0.01, 0.28, 0.70, 0.00, 0.00, 0.00, 0.00]	[1.96, 0.22, -0.39]	0	[1.72, 0.75, 0.21]
3	3	[0, 0.33, -1.81]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[2.27, -0.01, -0.62]	0	[1.63, 0.47, 0.36]
4	3	[0, 0.58, -1.88]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[2.28, 0.01, -0.8]	4	[1.75, 0.89, 0.38]
5	2	[0, 0.72, -0.93]	6	[0.00, 0.00, 0.00, 0.00, 0.13, 0.00, 0.87, 0.00]	[2.58, 0.06, -0.7]	0	[1.55, 0.25, 0.25]
6	4	[0, 1.74, -1.49]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.03, 0.01, 0.97]	[2.73, 0.15, -0.78]	6	[1.21, 0.59, 0.73]
average		[0, 1.13, -1.38]			[2.37, 0.24, -0.57]		[1.55, 0.55, 0.38]

Table A.12: Experiment results of run 11

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.70, -0.43, -1.77]		
1	3	[0, 0.22, -1.69]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.92, -0.36, -2.12]	1	[1.44, 1.26, -0.32]
2	2	[0, 0.32, -1.12]	1	[0.22, 0.78, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[0.25, 0.03, -1.13]	0	[1.42, 0.64, -0.3]
3	3	[0, 0.37, -0.62]	1	[0.18, 0.82, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.04, 0.12, -0.86]	2	[1.17, 0.7, 0.23]
4	1	[0, 0.45, -1.68]	3	[0.00, 0.01, 0.17, 0.81, 0.00, 0.00, 0.00, 0.00]	[1.03, -0.11, -0.89]	0	[1.97, 0.57, -0.23]
5	3	[0, 0.37, -1.95]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[1.46, -0.24, -0.97]	0	[1.41, 0.75, 0.33]
6	3	[0, 0.35, -1.90]	4	[0.00, 0.00, 0.01, 0.01, 0.99, 0.00, 0.00, 0.00]	[1.47, -0.24, -1.02]	4	[1.68, 0.87, 0.51]
7	2	[0, 0.42, -1.41]	6	[0.00, 0.00, 0.00, 0.00, 0.18, 0.00, 0.81, 0.00]	[1.64, -0.34, -1.11]	0	[1.48, 0.79, 0.15]
8	3	[0, 1.21, -1.67]	6	[0.00, 0.00, 0.00, 0.00, 0.18, 0.00, 0.82, 0.00]	[1.60, -0.31, -1.11]	5	[1.51, 0.82, -0.01]
9	4	[0, 0.1, -1.59]	7	[0.00, 0.00, 0.00, 0.00, 0.01, 0.96, 0.04, 0.00]	[1.63, -0.33, -1.13]	6	[1.47, 0.6, 0.3]
average		[0, 0.42, -1.51]			[0.90, -0.20, -1.15]		[1.5, 0.78, 0.07]

Table A.13: Experiment results of run 12

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.59, -0.45, -1.71]		
1	3	[0, 0.33, -1.95]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.61, -0.28, -2.27]	1	[1.1, 0.52, 0.21]
2	2	[0, 0.44, -1.19]	1	[0.22, 0.78, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.01, 0.12, -1.32]	0	[1.69, 0.11, 0.19]
3	3	[0, 0.48, -0.77]	1	[0.22, 0.78, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.00, 0.16, -1.16]	2	[1.82, 0.82, 0.05]
4	1	[0, 0.67, -1.68]	3	[0.00, 0.00, 0.45, 0.55, 0.00, 0.00, 0.00, 0.00]	[1.25, 0.06, -0.72]	0	[1.62, 0.81, 0.03]
5	3	[0, 0.41, -2.01]	4	[0.00, 0.00, 0.01, 0.02, 0.97, 0.00, 0.00, 0.00]	[1.18, -0.09, -0.62]	0	[1.69, 0.83, 0.47]
6	3	[0, 0.37, -2.00]	4	[0.00, 0.00, 0.01, 0.01, 0.98, 0.00, 0.00, 0.00]	[1.14, -0.06, -0.83]	4	[1.98, 0.93, 0.34]
7	2	[0, 0.55, -1.41]	6	[0.00, 0.00, 0.01, 0.04, 0.21, 0.00, 0.74, 0.00]	[1.22, -0.07, -0.68]	0	[1.78, 0.83, 0.49]
8	3	[0, 1.21, -1.69]	4	[0.00, 0.00, 0.00, 0.01, 0.88, 0.00, 0.12, 0.00]	[1.41, 0.01, -0.8]	0	[1.85, 0.42, 0.66]
9	4	[0, -0.13, -1.58]	5	[0.00, 0.00, 0.00, 0.00, 0.01, 0.94, 0.00, 0.04]	[1.47, 0.02, -0.74]	0	[1.85, 0.45, 0.33]
average		[0, 0.48, -1.59]			[1.01, -0.01, -1.02]		[1.71, 0.64, 0.31]

Table A.14: Experiment results of run 13

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.71, -0.40, -1.75]		
1	0	[0, 0.69, -1.56]	0	[0.00, 0.19, 0.81, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.74, -0.01, -2.22]	1	[0.97, 0.81, 0.15]
2	2	[0, 1.33, -1.09]	1	[0.07, 0.92, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.49, 0.24, -1.15]	0	[1.78, 0.52, 0.72]
3	3	[0, 1.54, -1.63]	1	[0.10, 0.90, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.55, 0.31, -1.12]	2	[1.45, 0.71, 0.15]
4	3	[0, 0.27, -1.88]	1	[0.11, 0.89, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[2.14, 0.24, -1.48]	2	[1.81, 0.58, 0.65]
5	1	[0, 1.35, -1.51]	3	[0.00, 0.01, 0.00, 0.99, 0.00, 0.00, 0.00, 0.00]	[3.05, 0.39, -0.87]	0	[1.65, 0.65, 0.93]
6	1	[0, 0.44, -1.96]	3	[0.00, 0.02, 0.00, 0.98, 0.00, 0.00, 0.00, 0.00]	[3.09, 0.28, -1.04]	3	[1.49, 0.36, 0.91]
7	3	[0, 0.96, -1.74]	4	[0.00, 0.00, 0.00, 0.01, 0.99, 0.00, 0.00, 0.00]	[2.82, 0.57, -0.64]	0	[1.58, 0.68, 0.61]
8	3	[0, 0.37, -1.81]	4	[0.00, 0.00, 0.00, 0.00, 1.00, 0.00, 0.00, 0.00]	[2.69, 0.54, -0.71]	4	[1.42, 0.74, 0.74]
9	2	[0, 1.24, -1.79]	6	[0.00, 0.00, 0.00, 0.00, 0.27, 0.00, 0.73, 0.00]	[2.51, 0.61, -0.57]	0	[1.64, 0.5, 1.01]
10	4	[0, 1.68, -0.99]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.10, 0.00, 0.89]	[2.50, 0.68, -0.47]	6	[1.5, 0.06, 0.7]
11	4	[0, 0.28, -1.63]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.09, 0.00, 0.90]	[2.47, 0.66, -0.47]	6	[1.77, 0.44, 0.31]
average		[0, 0.92, -1.60]			[2.33, 0.41, -0.98]		[1.55, 0.55, 0.63]

Table A.15: Experiment results of run 14

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.72, -0.46, -1.84]		
1	2	[0, 1.23, -1.37]	1	[0.18, 0.82, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[0.28, -0.55, -2.55]	0	[1.02, 0.87, -0.44]
2	1	[0, 0.87, -1.63]	3	[0.01, 0.05, 0.00, 0.94, 0.00, 0.00, 0.00, 0.00]	[0.62, -0.18, -1.74]	0	[1.66, 0.5, 0.01]
3	3	[0, 0.15, -1.83]	4	[0.00, 0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.00]	[0.73, -0.26, -1.59]	0	[1.87, 0.66, -0.17]
4	2	[0, 1.06, -1.62]	6	[0.00, 0.00, 0.00, 0.00, 0.12, 0.00, 0.88, 0.00]	[0.83, -0.29, -1.61]	0	[1.55, 0.45, 0.57]
5	3	[0, 1.21, -1.70]	6	[0.00, 0.00, 0.00, 0.00, 0.12, 0.00, 0.88, 0.00]	[0.84, -0.26, -1.62]	5	[1.34, 0.68, 0.15]
6	4	[0, 0.22, -1.60]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.02, 0.01, 0.97]	[0.83, -0.25, -1.58]	6	[1.81, 0.31, 0.06]
average		[0, 0.79, -1.63]			[0.69, -0.30, -1.78]		[1.54, 0.58, 0.03]

Table A.16: Experiment results of run 15

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.57, -0.39, -1.75]		
1	2	[0, 1.26, -1.40]	1	[0.27, 0.73, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[1.12, 0.14, -1.49]	0	[1.72, 0.65, 0.3]
2	1	[0, 0.86, -1.68]	3	[0.00, 0.01, 0.34, 0.64, 0.00, 0.00, 0.00, 0.00]	[1.94, 0.38, -1.26]	0	[1.76, 0.76, 0.48]
3	3	[0, 0.23, -1.87]	4	[0.00, 0.00, 0.01, 0.02, 0.97, 0.00, 0.00, 0.00]	[1.15, 0.29, -1.69]	0	[1.81, 0.8, 0.37]
4	2	[0, 1.12, -1.69]	6	[0.00, 0.00, 0.00, 0.00, 0.33, 0.00, 0.67, 0.00]	[1.23, 0.25, -1.88]	0	[1.53, 0.39, -0.1]
5	3	[0, 1.22, -1.70]	6	[0.00, 0.00, 0.00, 0.00, 0.29, 0.00, 0.71, 0.00]	[1.06, 0.27, -1.84]	5	[1.37, 0.92, 0.43]
6	4	[0, 0.24, -1.51]	7	[0.00, 0.00, 0.00, 0.00, 0.01, 0.00, 0.03, 0.96]	[0.51, 0.27, -1.66]	6	[1.69, 0.92, 0.43]
average		[0, 0.82, -1.64]			[1.17, 0.27, -1.64]		[1.64, 0.74, 0.32]

Table A.17: Experiment results of run 16

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.64, -0.43, -1.81]		
1	3	[0, 0.29, -1.86]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.87, -0.28, -2.11]	1	[0.87, 0.85, 0.27]
2	2	[0, 1.49, -1.63]	1	[0.46, 0.54, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.02, -0.47, -1.84]	0	[1.21, 0.57, -0.11]
3	1	[0, 1.25, -1.74]	3	[0.02, 0.02, 0.24, 0.73, 0.00, 0.00, 0.00, 0.00]	[1.20, -0.37, -1.46]	0	[1.79, 0.68, 0.03]
4	3	[0, 0.05, -1.85]	4	[0.00, 0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.00]	[1.66, -0.46, -1.32]	0	[1.69, 0.38, 0.23]
5	3	[0, 0.32, -1.95]	4	[0.00, 0.00, 0.00, 0.01, 0.98, 0.00, 0.00, 0.00]	[1.61, -0.45, -1.33]	4	[1.88, 0.75, 0.38]
6	2	[0, 1.22, -1.75]	6	[0.00, 0.00, 0.00, 0.01, 0.18, 0.00, 0.81, 0.00]	[1.84, -0.48, -1.25]	0	[1.59, 0.6, 0.04]
7	3	[0, 1.03, -1.73]	6	[0.00, 0.00, 0.00, 0.01, 0.29, 0.00, 0.70, 0.00]	[1.77, -0.46, -1.25]	5	[1.68, 0.82, -0.16]
8	4	[0, 0.48, -1.42]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.09, 0.00, 0.91]	[1.83, -0.45, -1.24]	6	[1.5, 0.6, -0.03]
average		[0, 0.77, -1.74]			[1.13, -0.43, -1.47]		[1.53, 0.66, 0.08]

Table A.18: Experiment results of run 17

time	user behav.		ps_belief		f_c	prompt	
	prop.	epa	value	probability distribution		prop.	epa
0					[-0.61, -0.38, -1.77]		
1	3	[0, 0.35, -1.88]	0	[1.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[-0.89, -0.07, -2.18]	1	[1.07, 0.81, 0.22]
2	2	[0, 1.5, -1.68]	1	[0.12, 0.88, 0.00, 0.00, 0.00, 0.00, 0.00, 0.00]	[0.16, -0.03, -1.13]	0	[1.25, 0.77, -0.06]
3	1	[0, 1.26, -1.70]	3	[0.01, 0.05, 0.01, 0.93, 0.00, 0.00, 0.00, 0.00]	[1.14, 0.64, -1.55]	0	[1.92, 0.37, -0.03]
4	3	[0, 0.19, -1.88]	4	[0.00, 0.00, 0.00, 0.02, 0.98, 0.00, 0.00, 0.00]	[1.91, 0.73, -1.72]	0	[1.8, 0.29, 0.99]
5	3	[0, 0.34, -1.86]	4	[0.00, 0.00, 0.00, 0.04, 0.96, 0.00, 0.00, 0.00]	[1.90, 0.75, -1.77]	4	[1.82, 0.26, 1]
6	2	[0, 1.22, -1.71]	6	[0.00, 0.00, 0.00, 0.00, 0.24, 0.00, 0.76, 0.00]	[2.12, 0.76, -1.79]	0	[1.48, 0.43, 0.21]
7	3	[0, 1.03, -1.71]	6	[0.00, 0.00, 0.00, 0.00, 0.26, 0.00, 0.74, 0.00]	[2.16, 0.77, -1.78]	5	[1.54, 0.09, 0.66]
8	4	[0, 0.4, -1.57]	7	[0.00, 0.00, 0.00, 0.00, 0.00, 0.14, 0.01, 0.85]	[2.27, 0.79, -1.73]	6	[1.59, 0.31, 0.74]
average		[0, 0.79, -1.75]			[1.35, 0.54, -1.71]		[1.56, 0.42, 0.47]