

# An assistive handwashing system with emotional intelligence

Luyuan Lin

University of Waterloo

*Supervisor:*  
*Jesse Hoey*

July 30, 2014

# Agenda

## 1 Problem Statement

- Motivation
- Objectives

## 2 Basic Concepts

- Affect Control Theory (ACT)
- The BayesACT Framework

## 3 System Design

## 4 Experimental Results

## 5 Discussion

# Problem Statement - Motivation

## The COACH system



- An assistive system helping people with dementia (e.g. Alzheimer's Disease) completing daily activities
- Works well for some persons, but not as well for others
- Only considers functional states of users

# Problem Statement - Objectives

To augment the COACH system with an emotional reasoning engine so that the augmented system:

- is designed in a portable and extensible way
- runs in real-time from the perspective of the user group
- provides a level of functional assistance
- produces the prompts according to the emotional state of a user

Using Emotional Intelligence in Assitive Systems is the future direction.

## Affect Control Theory (ACT)

- represents emotions as EPA vectors, where E stands for “evaluation”, P stands for “potency”, and A stands for “activity”
- describes social events by an Actor-Behaviour-Object grammar
- “fundamentals” of identities and behaviours; shared between people within a same culture
- “transient impressions”: emotional feelings caused by a specific event

## Affect Control Theory (ACT)

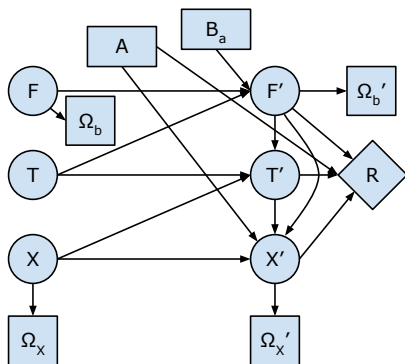
- represents emotions as EPA vectors, where E stands for “evaluation”, P stands for “potency”, and A stands for “activity”
- describes social events by an Actor-Behaviour-Object grammar
- “fundamentals” of identities and behaviours; shared between people within a same culture
- “transient impressions”: emotional feelings caused by a specific event

## The ACT Principle

Actors work to experience transient impressions that are consistent with their fundamental sentiments.

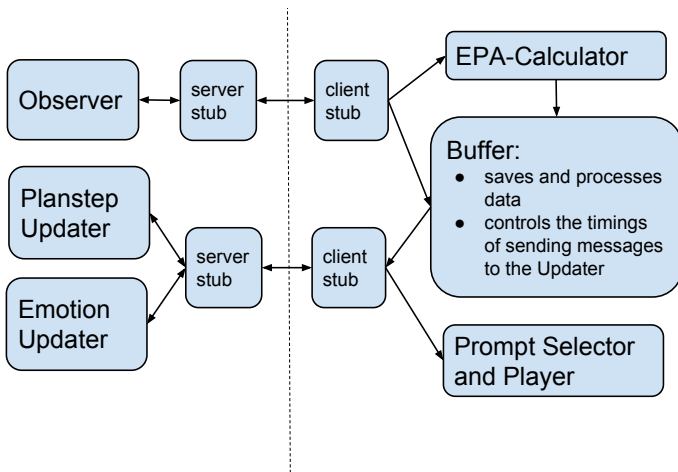
# Concepts - BayesACT

- A Bayesian version of the ACT theory
- Extends the ACT with POMDP model
- Uses a “turn-taking” model and represents state variables for Agent, Behaviour and Client (ABC)



- States  $S = \{X, F, T\}$ , where  $F = \{F_{ij}\}$ ,  $T = \{T_{ij}\}$ ,  $i \in \{a, b, c\}$ ,  $j \in \{e, p, a\}$
- Note:  $F_c$  denotes the agent's belief of the client's identity
- Observations  $\Omega = \{\Omega_X, \Omega_b\}$
- Actions  $\{A, B_a\}$
- Calculate  $\{A, B_a\}$  based on  $\{X, F, T\}$

# Design - Overview





# Design - the Planstep and Emotion Updater

Use the BayesACT framework in the handwashing scenario

- Recall: BayesACT includes states  $S = \{X, F, T\}$ , observations  $\Omega = \{\Omega_x, \Omega_b\}$ , and agent actions  $\{A, B_a\}$

# Design - the Planstep and Emotion Updater

Use the BayesACT framework in the handwashing scenario

- Recall: BayesACT includes states  $S = \{X, F, T\}$ , observations  $\Omega = \{\Omega_x, \Omega_b\}$ , and agent actions  $\{A, B_a\}$
- In our hand-washing system,  $X = \{X_{turn}, X_{ps}, X_{aw}, X_{behav}\}$

# Design - the Planstep and Emotion Updater

Use the BayesACT framework in the handwashing scenario

- Recall: BayesACT includes states  $S = \{X, F, T\}$ , observations  $\Omega = \{\Omega_x, \Omega_b\}$ , and agent actions  $\{A, B_a\}$
- In our hand-washing system,  $X = \{X_{turn}, X_{ps}, X_{aw}, X_{behav}\}$
- Observations give evidence to the state variables
- Note: the “confidence” of  $\Omega_b$  can be specified by  $\gamma$ , which is the variance of a normal (Gaussian) distribution.

# Design - the Planstep and Emotion Updater

Use the BayesACT framework in the handwashing scenario

- Recall: BayesACT includes states  $S = \{X, F, T\}$ , observations  $\Omega = \{\Omega_x, \Omega_b\}$ , and agent actions  $\{A, B_a\}$
- In our hand-washing system,  $X = \{X_{turn}, X_{ps}, X_{aw}, X_{behav}\}$
- Observations give evidence to the state variables
- Note: the “confidence” of  $\Omega_b$  can be specified by  $\gamma$ , which is the variance of a normal (Gaussian) distribution.
- Compute  $X'_{ps}$  based on  $\Omega_x$ ,  $\Omega_b$  and  $\{X, F, T\}$

# Design - the Planstep and Emotion Updater

Use the BayesACT framework in the handwashing scenario

- Recall: BayesACT includes states  $S = \{X, F, T\}$ , observations  $\Omega = \{\Omega_x, \Omega_b\}$ , and agent actions  $\{A, B_a\}$
- In our hand-washing system,  $X = \{X_{turn}, X_{ps}, X_{aw}, X_{behav}\}$
- Observations give evidence to the state variables
- Note: the “confidence” of  $\Omega_b$  can be specified by  $\gamma$ , which is the variance of a normal (Gaussian) distribution.
- Compute  $X'_{ps}$  based on  $\Omega_x$ ,  $\Omega_b$  and  $\{X, F, T\}$
- Calculate  $\{A, B_a\}$  based on  $\{X, F, T\}$

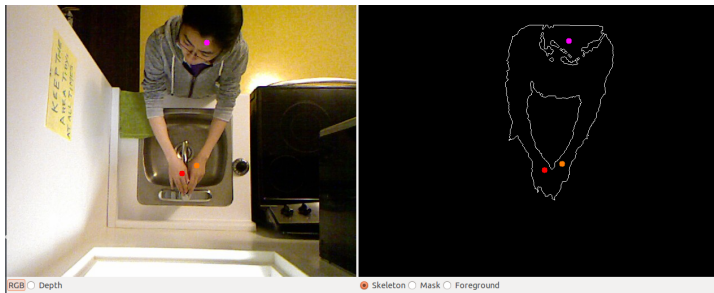
# Design - the EPA-Calculator

- Calculates affective meanings of user behaviours
- Feature Selection
  - analysis on facial expressions and speeches?
  - related research for this special application scenario hasn't been done yet

- Calculates affective meanings of user behaviours
- Feature Selection
  - analysis on facial expressions and speeches?
  - related research for this special application scenario hasn't been done yet
- Our approach
  - $E$  stays neutral (value = 0); its value is ignored in the reasoning engine
  - $P$  scaled from the expansiveness of the user's two hands
  - $A$  scaled from the moving speeds of the user's hands
  - Used piecewise linear interpolation method
- “Confidence” of  $\Omega_b$  can be specified in the reasoning engine

# Design - the Observer

- Step 1: Get the locations of the user's hands
  - bases on a body tracker implemented in previous work
  - obtains locations of body parts from depth images taken from an overhead perspective
  - was trained using partially labeled, unbalanced data, and is configurable and re-trainable

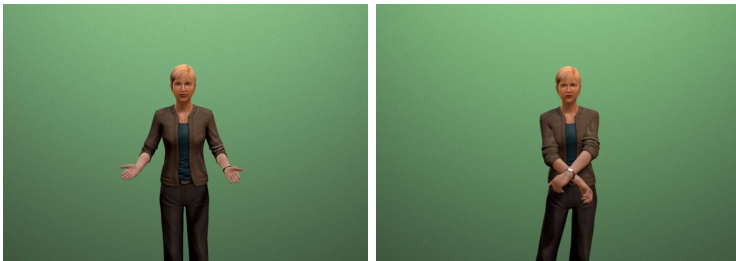




- Step 2: Map locations to user behaviours
  - if hands are close to an object, then there's high probability of performing the behaviour corresponded to the object
- Observation noise handled by the observation function in the reasoning engine

# Design - the Prompt Selector and Player

- The prompt dataset
  - 30 audio-visual prompts generated in previous study
  - created using the USC Virtual Human Toolkit
  - EPA values of videos evaluated by human raters



- A proper prompt is selected as the final prompt if it:
  - has the same propositional labels as the desired prompt
  - has the closest emotional (EPA) values as the desired prompt

# Experiments - Latency of the system

Average latency of the system

- 46.80ms for obtaining user behaviours
- 1.65s for calculating and updating functional and emotional beliefs
- 1.70s in total

The system runs in real-time from the perspective of its user group

# Experiments - Two laboratory tests

## Two laboratory tests

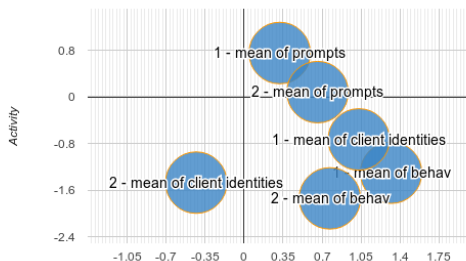
- washed hands while the system observed and assisted in real time
- acted more powerfully and more actively in the first test than the second
- link to video of test #2
- results shown in thesis

Another 15 tests were also run. Results are in the Appendix.

# Experiments - Conclusion

- Functionality performance
  - sometimes false positively recognizes an user behaviour
  - in general, is able to produce propositionally useful system prompts
- Emotionality performance

Comparison of  $P$  and  $A$  values in the two tests



Generally, user behaviours with higher  $P$  and higher  $A$  values lead to

- client identities with higher  $P$  and higher  $A$  values
- system prompts with lower  $P$  and higher  $A$  values
- as predicted by the Affect Control Theory

## Contribution

- Designed and implemented a prototypical hand-washing system that satisfies the objectives
- Tests also indicated a correlation between the EPA values of user behaviours, user identities, and system prompts

## Future Work

- Improve the EPA-Calculator
- Improve the prompt generation process
- Improve the Planstep- and Emotion- Updater
- Conduct clinical trials for the system

# Acknowledgement

This work is based on previous works of:

- Hoey, J., Schroder, T., & Alhothali, A. (2013, September). Bayesian affect control theory. In Affective Computing and Intelligent Interaction (ACII), 2013 Humaine Association Conference on (pp. 166-172). IEEE.
- Czarnuch, S., & Mihailidis, A. (2014 (in review)). Depth image hand tracking from an overhead perspective using partially labeled, unbalanced data: Development and real-world testing. IEEE Journal of Biomedical and Health Informatics.
- Malhotra, A., Yu, C., Schroder, T., & Hoey, J. (2014 (in review)). An exploratory study into the use of an emotionally aware cognitive assistant.

I'd like to take this opportunity to thank:

- Jesse Hoey
- James Tung and Peter van Beek
- Stephen Czarnuch, Aarti Malhotra and Celia Yu
- Families and friends, especially Chengbo Li, Xiao Yang, Enxun Wei and Luyi Lin

# Thank you!

- Questions?
- Comments?



## Model formulation

- The deflection  $\phi(F, T)$  between  $F$  and  $T$ :

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

- The probability of a post-action fundamental sentiment  $f'$ :

$$Pr(f'|f, t, x, b_a, \phi) \propto e^{-\phi(f', t') - \xi(f', f, b_a, x)} \quad (2)$$

where  $t'$  can be computed from  $\{f', t, x\}$  by empirically derived prediction equations of ACT.

- $Pr(x'|x, f', t', a)$ : how the application progresses
- $Pr(\omega_b|f)$  and  $Pr(\omega_x|x)$ : observation functions for the client behaviour sentiment and system state

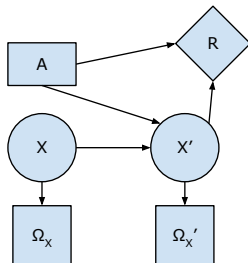
Update  $X'_{ps}$  based on  $\Omega_x$  and  $\{X_{ps}, X_{behav}, X_{aw}, F, T\}$

- *SampleXVar()* and *evalSampleXVar()*
- Pseudocode of *SampleXVar()* (on next page)
- $Pr : X_{behav} \rightarrow \Delta(\Omega_x)$  used in *evalSampleXVar()*

# Update $X'_{ps}$ based on $\Omega_x$ and $\{X_{ps}, X_{behav}, X_{aw}, F, T\}$ cont.

```
1: if Deflection(F, T) is high then
2:   threshold = high
3: else
4:   threshold = low
5: end if
6: if aw high then
7:   if prompted then
8:     if random_prob() < threshold then
9:       aw = low and not moving forward
10:    else if prompt wrong then
11:      aw = low and not moving forward
12:    else if likely then
13:      moving forward
14:    else if random_prob() < threshold then
15:      aw = low and not moving forward
16:    end if
17:  else
18:    if random_prob() < threshold then
19:      aw = low and not moving forward
20:    else
21:      aw stays high and moving forward
22:    end if
23:  end if
24: else
25:   if prompted then
26:     if random_prob() > threshold and
27:       prompt correct then
28:       move on and aw high
29:     else
30:       unlikely: aw high and not moving
31:       forward
32:     end if
33:   else
34:     unlikely: aw high and moving forward
35:   end if
```

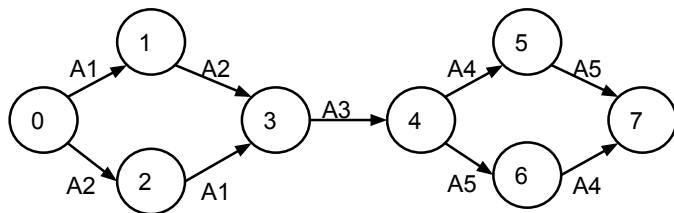
## Partially Observable Markov Decision Process (POMDP)



- A timeslice of a POMDP process
- Variables:  $\{ X, A, \Omega_X \}$
- $Pr : X \rightarrow \Delta(\Omega_X)$ ,  
 $Pr : X \times A \rightarrow \Delta(X)$
- Reward Function:  $R(A, X')$

# Solution - Representing “Functional States”

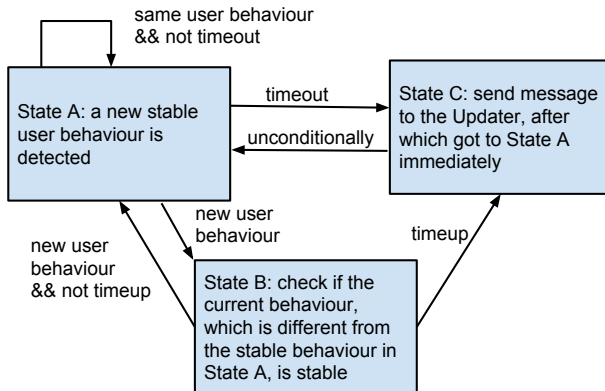
## Planstep Definition and Update Diagram



- Eight plansteps: (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”
- Five behaviours: A1 to A5 are “turn on water”, “put on soap”, “rinse hands”, “turn off water”, and “use towel”, respectively.

# Solution - The Buffer

- Between the Observer, the EPA-Calc, and the Reasoning Engine
- Controls timings of sending messages



- Smooths EPA values calculated by the Calculator

# Experiments - Parameter values used in laboratory experiments

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