IJCAI 2007 Workshop

on

AI for Human Computing (AI4HC'07)

ORGANIZERS:

Thomas Huang
Anton Nijholt
Maja Pantic
Alex Pentland

CTIT
Centre for Telematics and Information Technology

HYDERABAD, INDIA 2007

AMI
AUGMENTED MULTI-PARTY INTERACTION

IJCAI 2007 Workshop
on
AI for Human Computing (AI4HC'07)

Despite the fact that the research on Human Computing is still in its pioneering phase, promising approaches have been reported in the literature in the last couple of years on automatic analysis of human behavior (including affective computing and socially-aware computing), smart environments and intelligent interfaces. This workshop focuses on these topics, which form the essence of Human Computing. It brings together experts from around the world working in these fields and provides a state-of-the-art overview of new paradigms and challenges in AI research on Human Computing.

The Workshop Program

- 09.00 – 09.15: Opening
- 09.15 – 10.00: **Invited Talk**, J.F. Cohn
  "Foundations of Human Computing: Facial Expression and Emotion"

- 10.00 – 10.15: Coffee break & Poster preparation
- 10.15 – 12.45: **Oral Session 1**: (per talk – 20 min presentation + 5 min discussion)
  1) ***Position Paper***, M. Pantic, A. Pentland, A. Nijholt and T. Huang
     "Machine Understanding of Human Behavior"
  2) M. den Uyl
     "Towards Embeddable Vision Architecture for Human Computing"
  3) A. Oikonomopoulos, I. Patras, M. Pantic and N. Paragios
     "Trajectory-based Representation of Human Action"
  4) V. Muthukumar, E. Regentova, J. Zheng, A. Ponzio, T. Wu and Z. Devlin
     "Human Gaze Differentiation for Man-Machine Interaction using a Hierarchical Solution for Eye Detection and Tracking"
  5) J. Broekens and P. Haazebroek
     "Emotion and Reinforcement: Affective Facial Expressions Facilitate Robot Learning"

- 12.45 – 13.30: **Poster Session** & Coffee break
  1) D.F. Adamatti, J.S. Sichman and H. Coelho
     "Virtual Players in RPG"
  2) Z. Ruttkay, D. Reidsma and A. Nijholt
     "Unexploited Dimensions of Virtual Humans"
  3) S. Rao, V. Kumar, M. Hatala and D. Gasevic
     "Mixed Initiative Interfaces to Recognize, Regulate, and Reflect Programming Styles"
  4) P. van der Vet, O. Kulyk, I. Wassink, W. Fikkert, H. Rauwerda, B. van Dijk, G. van der Veer, T. Breit and A. Nijholt
     "Smart environments for collaborative design, implementation, and interpretation of scientific experiments"

- 13.30 – 14.15: Lunch break
- 14.15 – 15.00: **Invited Talk**, J. Cassell
  "Making (Virtual) Friends and Influencing (Virtual) People: Building Rapport in Humans and Virtual Humans"

- 15.00 – 15.15: Coffee break
- 15.15 – 17.15: **Oral Session 2**: (per talk – 20 min presentation + 5 min discussion)
  1) R. Poppe and R. Rienks
     "Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications"
  2) H. Raghavan, O. Madani and R. Jones
     "When will a Human in the Loop Accelerate Learning?"
  3) V. Kumar
     "Capturing and Disseminating Shareable Learning Experiences"
  4) D. Sonntag, R. Engel, G. Herzog, A. Pfalzgraf, N. Pfleger, M. Romanelli and N. Reithinger
     "Smart Web Handheld – Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services"

- 17.15 – 17.30: Coffee break
- 17.30 – 18.15: **Panel discussion** led by the workshop organizers and the invited speakers
  (To encourage a discussion and to allow enough time for the preparation of a fruitful discussion, a list of intriguing questions regarding the current and new paradigms and challenges facing the researchers in the field of Human Computing will be assembled together with the workshop participants at least two weeks before the workshop.)

- 18.15 – 18.30: Closing

Table of Contents

Invited Talk

Foundations of Human Computing: Facial Expression and Emotion

Jeffrey F. Cohn

Departments of Psychology & Psychiatry, University of Pittsburgh
Adjunct Faculty, Robotics Institute, Carnegie Mellon University

http://www.pitt.edu/~jeffcohn

Many people believe that emotions and subjective feelings are one and the same and that a goal of human-centered computing is emotion recognition. The first belief is outdated; the second mistaken. For human-centered computing to succeed, a different way of thinking is needed. Emotions are species-typical patterns that evolved because of their value in addressing fundamental life tasks. Emotions consist of multiple components that may include intentions, action tendencies, appraisals, other cognitions, central and peripheral changes in physiology, and subjective feelings. Emotions are not directly observable, but are inferred from expressive behavior, self-report, physiological indicators, and context. This talk will focus on expressive behavior because of its coherence with other indicators and the depth of research on the facial expression of emotion in behavioral and computer science. Among the topics to be discussed are approaches to measurement, timing or dynamics, individual differences, dyadic interaction, and inference. The main argument that this talk will support is that design and implementation of perceptual user interfaces may be better informed by considering the complexity of emotion, its various indicators, measurement, individual differences, dyadic interaction, and problems of inference.

Jeffrey Cohn is Professor of Psychology and Psychiatry at the University of Pittsburgh and Adjunct Faculty at the Robotics Institute, Carnegie Mellon University, where he leads the Face Group together with Professor Takeo Kanade. He earned his PhD in Clinical Psychology from the University of Massachusetts in Amherst and completed Clinical Internship at the University of Maryland Medical Center. For the past 20 years, he has conducted investigations in the theory and science of emotion, depression, and nonverbal communication. He has co-led interdisciplinary and inter-institutional efforts to develop methods of automated analysis of facial expression and prosody and applied these tools to research in human emotion, communication, biometrics, and human-computer interaction. He has published over 90 papers on these topics. His research has been supported by grants from the National Institute of Mental Health, the National Institute of Child Health and Human Development, the National Science Foundation, and the Defense Advanced Research Projects Agency.

Foundations of Human Centered Computing: Facial Expression and Emotion[*]

Jeffrey F. Cohn
University of Pittsburgh
Department of Psychology, Pittsburgh, PA 15260
jeffcohn@cs.cmu.edu

Abstract

Many people believe that emotions and subjective feelings are one and the same and that a goal of human-centered computing is emotion recognition. The first belief is outdated; the second mistaken. For human-centered computing to succeed, a different way of thinking is needed.

Emotions are species-typical patterns that evolved because of their value in addressing fundamental life tasks (Ekman, 1992a). Emotions consist of multiple components that may include intentions, action tendencies, appraisals, other cognitions, central and peripheral changes in physiology, and subjective feelings. Emotions are not directly observable, but are inferred from expressive behavior, self-report, physiological indicators, and context. I focus on expressive behavior because of its coherence with other indicators and the depth of research on the facial expression of emotion in behavioral and computer science. In this paper, among the topics I include are approaches to measurement, timing or dynamics, individual differences, dyadic interaction, and inference. I propose that design and implementation of perceptual user interfaces may be better informed by considering the complexity of emotion, its various indicators, measurement, individual differences, dyadic interaction, and problems of inference.

1 Introduction

How can computers recognize human emotions? Is this even the correct question? By emotion, people often think of subjective feelings, but emotions are more than that and subjective feeling is in no sense essential. There is no *sin qua non* for emotion. Emotions are species-typical patterns consisting of multiple components that may include intentions, action tendencies, appraisals, other cognitions, neuromuscular and physiological changes, expressive behavior, and subjective feelings. None of these is necessary or sufficient. In human-human interaction, intentions and action tendencies often are more important than what an individual may be feeling. People may or may not be aware of what they're feeling, and feelings often come about some time late in the temporal unfolding of an emotion.

A goal of human-centered computing is computer systems that can unobtrusively perceive and understand human behavior in unstructured environments and respond appropriately. Much work has strived to recognize human emotions. This effort is informed by the importance of emotion to people's goals, strivings, adaptation, and quality of life (Ekman, 2003; Lazarus, 1991) at multiple levels of organization, from intra-personal to societal (Keltner & Haidt, 1999). Efforts at emotion recognition, however, are inherently flawed unless one recognizes that emotion – intentions, action tendencies, appraisals and other cognitions, physiological and neuromuscular changes, and feelings – is not an observable. Emotion can only be inferred from context, self-report, physiological indicators, and expressive behavior (see Figure 1). The focus of the current paper is on expressive behavior, in particular facial expression, and approaches to measurement, feature selection, individual differences, interpersonal regulation, and inference.

Facial expression is a useful place to begin when thinking about foundations of human computing. Facial expression has been a subject of keen study in behavioral science for more than a hundred years(Darwin, 1872/1998; Ekman & Rosenberg, 2005), and within the past 10 years considerable progress has been made in automatic analysis of facial expression from digital video input (Pantic & Patras, 2006; Pantic & Rothkrantz, 2000; Tian, Cohn, & Kanade, 2005).

Facial expression correlates moderately with self-reported emotion (Ekman & Rosenberg, 2005), pain (Prkachin, 1992), craving (Sayette et al., 2003) and emotion-related central and peripheral physiology (Davidson, Ekman, Saron, Senulis, & Friesen, 1990; Levenson, Ekman, & Friesen, 1990). Facial expression and self-reported emotion have similar underlying dimensions (e.g., positive and negative affect) (Watson & Tellegen, 1985) and serve interpersonal functions by conveying communicative intent, signaling

[*] A previous version of this paper was originally published in the Proc. ACM Int'l Conf. Multimodal Interface 2006 (Copyright © ACM Press).

affective information in social referencing, and contributing to the regulation of social interaction (Cohn & Elmore, 1988; Schmidt & Cohn, 2001). Cultural differences in how and when to express emotion emerge in infancy (Malatesta & Haviland, 1982; Oster et al., 1996). As a measure of trait affect and socialization, stability in facial expression emerges early in life (Cohn & Campbell, 1992). By adulthood, stability is moderately strong, comparable to what has been found for self-reported emotion (Cohn, Schmidt, Gross, & Ekman, 2002). Expressive changes in the face are a rich source of cues about intra- and interpersonal indicators and functions of emotion (Gottman, Levenson, & Woodin, 2001; Keltner & Haidt, 1999).

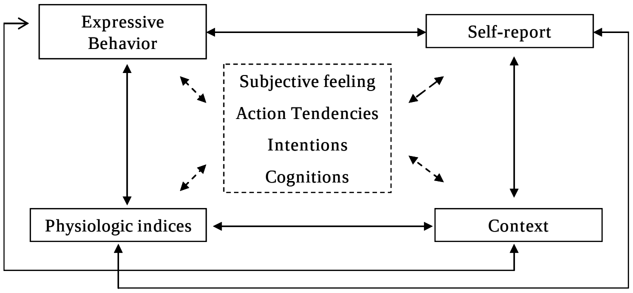


Figure 1. Components and indicators of emotion. Solid boxes represent observables, dashed boxes latent variables. Solid arrows indicate observable correlations among indicators. Large correlations among multiple indicators indicate greater coherence among indicators. Dashed arrows represent inferential paths. Paths between emotion components are omitted.

Here, I present key issues to consider in designing interfaces that approach the naturalness of face-to-face interaction. These include approaches to measurement, types of features, individual differences, dyadic interaction, and inference.

2 Approaches to Measurement

Two major approaches are sign- and message judgment (Cohn, Ambadar, & Ekman, In press). In message judgment, the observer's task is to make *inferences* about something underlying the facial behavior, such as emotion or personality. In measuring sign vehicles, the task is to *describe* the surface of behavior, such as when the face moves a certain way. As an example, upon seeing a smiling face, an observer with a judgment-based approach would make judgments such as "happy," whereas an observer with a sign-based approach would code the face as having an upward, oblique movement of the lip corners. Message judgment implicitly assumes that the face is an emotion "read out." Sign-based measurement is agnostic and leaves inference to higher-order decision making.

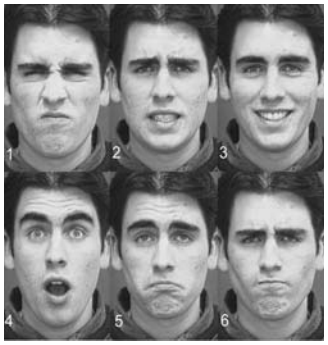


Figure 2. Emotion-specified expressions: disgust, fear, joy, surprise, sadness, and anger.

2.1 Message Judgment

Message judgment approaches define facial expressions in terms of inferred emotion. Of the various descriptors, those of Ekman have been especially influential. Ekman (Ekman, 1992b) proposed six "basic emotions." They are joy, surprise, sadness, disgust, fear, and anger. Each was hypothesized to have universally recognized and displayed signals, universal elicitors, specific patterns of physiology, rapid, unbidden onset, and brief duration, among other attributes. Since then, some additional emotions, such as embarrassment and contempt, have been added. Examples of facial expressions for the initial six basic emotions are shown in Figure 2. Most research in automatic recognition of facial expression (Pantic & Rothkrantz, 2003; Pantic, Sebe, Cohn, & Huang, 2005) and much emotion research in psychology (Keltner & Ekman, 2000) has concentrated on one or more of these six emotions. This list, however, was never intended as exhaustive of human emotion. Rather, it was proposed in terms of conformity with the criteria noted.

An especially important class of expressions is those that include traces of contradictory emotion expression. Masking smiles (Ekman, Friesen, & O'Sullivan, 1988), in which smiling is used to cover up or hide an underlying emotion are the best known. An example is shown in Figure 3. Signs of contempt (AU 14) and sadness (AU 15) can be seen along with the smile (AU 12). Negative emotion is believed to "leak" through the dominant positive expression.

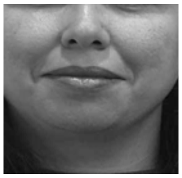


Figure 3. Example of masking smile (AU 12+14+15).

2.2 Sign Measurement

Cohn & Ekman (Cohn & Ekman, 2005) review manual methods for labeling facial actions. Of the various methods, the Facial Action Coding System (FACS) (Ekman & Friesen, 1978; Ekman, Friesen, & Hager, 2002) is the most comprehensive, psychometrically rigorous, and widely used (Cohn, Ambadar, & Ekman, In press; Ekman & Rosenberg, 2005). Using FACS and viewing video-recorded facial behavior at frame rate and slow motion, users can manually label nearly all possible facial expressions, which are decomposed into action units (AUs). Action units, with some qualifications, are the smallest visually discriminable facial movements. By comparison, other systems are less thorough (Malatesta, Culver, Tesman, & Shephard, 1989), fail to differentiate between some anatomically distinct movements (Oster, Hegley, & Nagel, 1992), consider as separable movements that are not anatomically distinct (Oster, Hegley, & Nagel, 1992), and often assume a one-to-one mapping between facial expression and emotion (Cohn, Ambadar, & Ekman, In press; Cohn & Ekman, 2005).

The most recent version of FACS specifies 9 action units in the upper face, 18 in the lower face, 11 for head position and movement, nine for eye position and movement, and additional descriptors for miscellaneous actions, gross body movement, and supplementary codes.

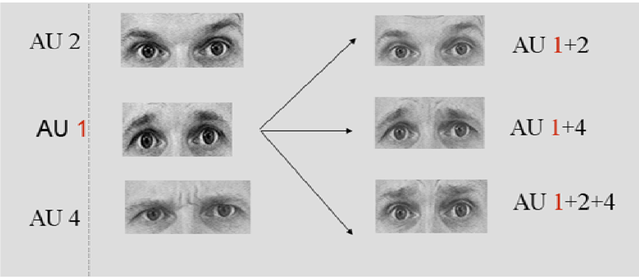


Figure 4. Examples of individual action units and action unit combinations. AU 1+2 is an additive combination. AU 1+4 and AU 1+2+4 are non-additive, comparable to co-articulation effects in speech.

Action units may occur singly or in combinations. Action unit combinations may be additive or non-additive. In additive combinations, the appearance of each action unit is independent; whereas in non-additive combinations they modify each other's appearance. Non-additive combinations are analogous to co-articulation effects in speech, in which one phoneme modifies the sound of ones with which it is contiguous. An example of an additive combination in FACS is AU 1+2, which often occurs in surprise (along with eye widening, AU 5) and in the brow-flash greeting (Eibl-Eibesfeldt, 1989). The combination of these two action units raises the inner (AU 1) and outer (AU 2) corners of the eyebrows and causes horizontal wrinkles to appear across the forehead. The appearance changes associated with AU 1+2 are the product of their joint actions.

An example of a non-additive combination is AU 1+4, which often occurs in sadness (Darwin, 1872/1998) (see Figure 4). When AU 1 occurs alone, the inner eyebrows are pulled upward. When AU 4 occurs alone, they are pulled together and downward. When AU 1 and AU 4 occur together, the downward action of AU 4 is modified. An example is shown in Figure 4. The result is that the inner eyebrows are raised and pulled together. This action typically gives an oblique shape to the brows and causes horizontal wrinkles to appear in the center of the forehead, as well as other changes in appearance. Automatic recognition of non-additive combinations presents similar complexity to that of co-articulation effects in speech. Failure to account for non-additive combination in automatic recognition exploits the correlation among AUs and can lead to inflated estimates of algorithm performance.

2.3 Reliability

The reliability of manually labeled images is a critical concern for machine learning algorithms. If ground truth is contaminated by 20-30% error, which is not uncommon, that is a significant drag on algorithm performance. For both message judgment and sign-based approaches, similar concerns arise. Using AUs as an example, at least four types of reliability (i.e., agreement between observers) are relevant to the interpretation of substantive findings. These are reliability for occurrence/non-occurrence of individual AUs, temporal precision, intensity, and aggregates. Most research in automatic facial expression analysis has focused on occurrence/non-occurrence (Pantic & Rothkrantz, 2000; Tian, Cohn, & Kanade, 2005).

Temporal precision refers to how closely observers agree on the timing of action units, such as when they begin or end. This level of reliability becomes important when examining features such as response latency and turn taking (see Section 5). Action unit intensity becomes important for questions such as whether facial expression is influenced by audience effects (Fridlund et al., 1990). Several groups have found, for instance, that people tend to smile more intensely in social contexts than when they are alone (Cohn & Schmidt, 2004; Fridlund et al., 1990). A related question is whether two measurement systems have concurrent validity for continuous measures of intensity. Our research group recently examined inter-system precision for intensity by comparing Automatic Facial Image Analysis (AFA v.4) with continuous ratings of affective intensity by human observers. Lip-corner displacement in spontaneous smiles was measured from video by AFA. Human observers made continuous ratings of affective intensity using a joy-stick like device. We found high concurrent validity between the two methods (see Figure 5 for an example) (Ibanez, Messinger, Ambadar, & Cohn, 2006; Messinger et al., 2006).

3 Dynamics

Both the configuration of facial features and the timing of facial actions are important in emotion expression and recognition. The configuration of facial actions (whether emotion-specified expressions or individual action units) in relation to emotion, communicative intent, and action tendencies has been a major research topic. Less is known about the timing of facial actions, in part because manual measurement of timing is coarse and labor intensive. We know, however, that people are highly sensitive to the timing of facial actions (Edwards, 1998) in social settings. Slower facial actions, for instance, appear more genuine (Krumhuber & Kappas, 2005), as do those that are more synchronous in their movement (Frank & Ekman, 1997). Especially subtle facial expressions become visible only when motion information is available to the perceiver (Ambadar, Schooler, & Cohn, 2005). Rapid responses to perception of facial expression can be detected within 0.5 seconds using facial EMG (Dimberg, Thunberg, & Grunedal, 2002). Recently, automatic facial image analysis has shown strong concurrent validity with facial EMG (Cohn & Kanade, in press), which suggests that it has similar capability.

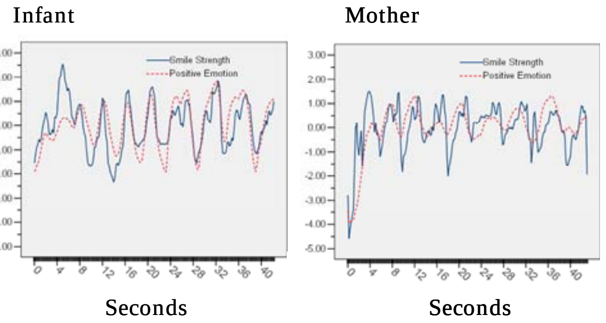


Figure 5. Time series for AFA-measured lip-corner displacement and human-observer based ratings of positive affect in a mother-infant dyad. Data series for human observers are shifted by about ½ second to adjust for human reaction time. (Ibanez, Messinger, Ambadar, & Cohn, 2006)

Dynamics is especially important to inferences about communicative intention. Using automatic facial image analysis to quantify the timing of facial actions, research by the CMU/Pitt group found that dynamic features discriminated between deliberate and spontaneous smiles with 89% accuracy (Cohn & Schmidt, 2004). Adding duration and amplitude to the classifier increased accuracy to 93%. Comparable findings were recently reported by (Valstar, Pantic, Ambadar, & Cohn, 2006). Using similar features, amusement, embarrassment, and polite smiles were discriminated with 83% accuracy (Kanade, Hu, & Cohn, 2005), which is comparable to that of human judges.

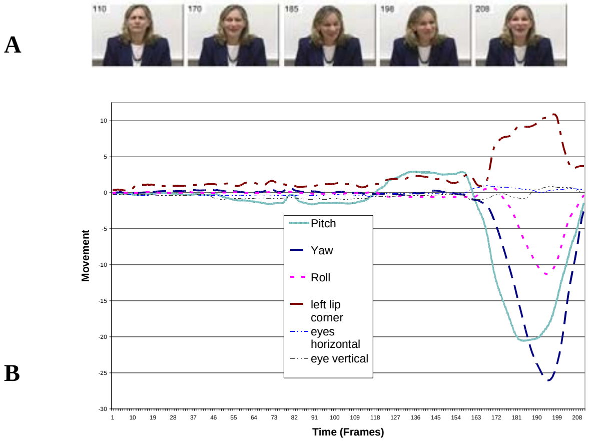


Figure 6. Multimodal coordination of head motion, lip-corner displacement, and gaze in smiles of embarrassment. A: Selected frames from image sequence depicting embarrassment. B: Corresponding time series. Reprinted with permission from (Cohn et al., 2004). (©2004 IEEE)

Recent work suggests that multimodal coordination of facial expression, head motion, and gesture is a defining feature of embarrassment (Keltner, 1995). An example is illustrated in Figure 6. Note that head pitch is closely coordinated with smile intensity. As the head pitches down, smile intensity increases, decreasing again only as the head comes back to frontal. For human-computer interaction, dynamic features are important to empirically based inferences about the meaning of otherwise similar facial actions, such as lip corner raise in smiling.

4 Individual Differences

As noted above, stable individual differences in facial expression emerge early in development and by adulthood represent 25% or more of the variation in emotion expression (Cohn, Schmidt, Gross, & Ekman, 2002; Moore, Cohn, & Campbell, 1997). Individual differences include reaction range for positive and negative affect and specific emotions and the probability of conforming to display rules. Display rules are culturally specific prescriptions for when and how to show emotion in various contexts. Sources of individual differences in emotion expression include temperament, personality, gender, socialization, and cultural background (Camras & Chen, 2006; Matsumoto & Willingham, 2006; Oster et al., 1996). In some cultures, for instance, children learn not to express anger; whereas in others, anger is considered important to self expression. Among traditional Japanese, for instance, anger is less likely to be shown outside the family than in the U.S. (Markus & Kitayama, 1991). As another example, European-American and Chinese-American couples differ in proportion of positive and negative expressions, but not in autonomic reactivity or self-

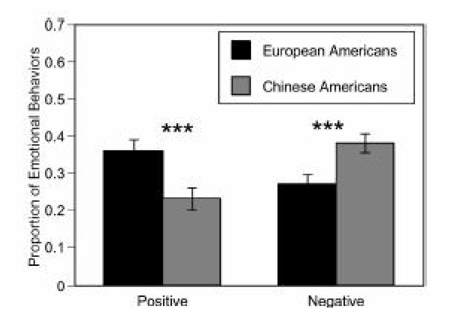


Figure 7. Cultural differences in emotional expression between European-American and Chinese-American couples. Observations were made while they discussed conflicts in their relationship. (Tsai, Levenson, & McCoy, 2006). (©2004 APA)

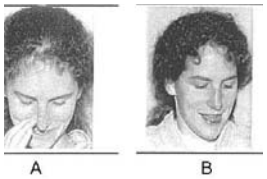


Figure 8. Some expressions appear in all or almost all cultures. Others are culture specific (A and B, respectively). Examples here are for embarrassment. From (Haidt & Keltner, 1999). (©1998 Taylor & Francis)

reported emotion, when discussing conflicts in their relationship (Tsai, Levenson, & McCoy, 2006). Emotion expressions may also be ritualized and culture specific. The tongue-bite display communicates embarrassment/shame in parts of India and south Asia but not the U.S. (see Figure 8). Within a given culture, individual differences in facial expression of all sources are strong enough to serve as a biometric (Cohn, Schmidt, Gross, & Ekman, 2002). An important implication for perceptual computing is that inferences about emotion will become more reliable when individual differences are taken into account.

5 Dyadic Interaction

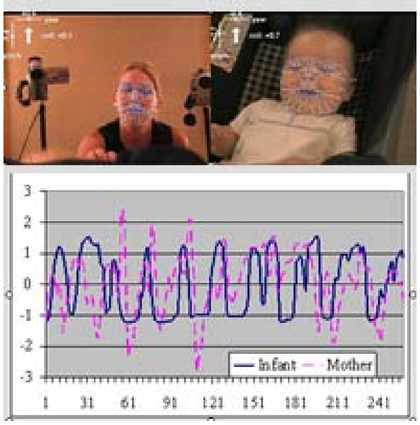


Figure 9. Example of interaction analysis. Synchrony and reciprocity of smiling between mother and infant. Source: (Ibanez, Messinger, Ambadar, & Cohn, 2006; Messinger et al., 2006).

Synchrony or coherence refers to the extent to which individuals are moving together in time with respect to one or more continuous output measures, such as affective valence or level of arousal. Reciprocity refers to the extent to which behavior of one individual is contingent on that of the other. Both synchrony and reciprocity have proven informative in studies of marital interaction, social development, and social psychology. Figure 9 shows an example taken from mother-infant interaction (Ibanez, Messinger, Ambadar, & Cohn, 2006; Messinger et al., 2006). Facial features and head motion were tracked automatically by the CMU/Pitt automated facial image analysis system version 4 (Cohn & Kanade, in press). The time series plot shows displacement of mother and infant lip-corners during smiles. Note that while partners tend to cycle together, there is a pattern of non-stationarity in which mother and infant take turns in leading the dyad into shared smiling, which is indicated by mother and infant time series increasing together. An important advantage of measures derived from interaction analysis is that they are largely outside of people's awareness and are difficult to manipulate intentionally.

Coordinated interpersonal timing (CIT) is the extent to which participants in a social interaction match the duration of interpersonal pauses or floor switches (Jaffe, Beebe, Feldstein, Crown, & Jasnow, 2001). Floor switches are pauses that occur between the time when one person stops speaking and another begins. Coordination of floor switches follows an inverted U-shaped function in relation to affective intensity. Mid-range values are associated with optimal affective involvement and interpersonal attraction. CIT has been studied most often with respect to vocal timing, but applies equally to facial expression and other modalities. CIT is impaired in clinical depression, with switching pauses becoming longer, more variable, and less predictable (Zlochower & Cohn, 1996).

In behavioral science, time- and frequency domain analyses have emphasized issues of quasi-periodicity in the timing of expressive behavior and bidirectional influence with respect to amplitude (Cohn & Tronick, 1988). Lag-sequential and related hidden Markov modeling have been informative with respect to the dynamics of discrete actions and individual and dyadic states (Cohn & Tronick, 1987). Recent work with dampened oscillator models considers regulation of changes in velocity and acceleration (Chow, Ram, Boker, Fujita, & Clore, 2005). Most approaches assume that time series are stationary. This assumption may not always hold for behavioral data. Boker (Boker, Xu, Rotondo, & King, 2002) identified "symmetry breaks," in which the pattern of lead-lag relationships between partners abruptly shifts. Failure to model these breaks may seriously compromise estimates of mutual influence.

6 Conclusion

Emotions are species-typical patterns that evolved because of their value in addressing fundamental life tasks (Ekman, 1992b). They are central to human experience, yet largely beyond the comprehension of contemporary computer interfaces. Human-centered computing seeks to enable computers to unobtrusively perceive, understand, and respond appropriately to human emotion, to do so implicitly, without the need for deliberate human input. To achieve this goal, it is argued that we forgo the notion of "emotion recognition" and adopt an iterative approach found in human-human interaction. In daily life, we continually make inferences about other people's emotions – their intentions, action tendencies, appraisals, other cognitions, and subjective feelings – from their expressive behavior, speech, and context. The success of human-centered computing depends in part on its ability to adopt an iterative approach to inference. Computing systems are needed that can automatically detect and dynamically model a wide range of multimodal behavior from multiple persons, assess context, develop representations of individual differences, and formulate and test tentative hypotheses though the exchange of communicative signals. Part of the challenge is that the computer becomes an active agent, in turn influencing the very process it seeks to understand. Human emotions are moving targets.

Acknowledgments

Portions of this work were supported by NIMH grant R01 MH51435 and NSF grant HSD-0527444.

References

[Ambadar et al., 2005] Ambadar, Z., Schooler, J., & Cohn, J. F. (2005). Deciphering the enigmatic face: The importance of facial dynamics to interpreting subtle facial expressions. *Psychological Science, 16*, 403-410.

[Boker et al., 2002] Boker, S. M., Xu, M., Rotondo, J. L., & King, K. (2002). Windowed cross–correlation and peak picking for the analysis of variability in the association between behavioral time series. *Psychological Methods, 7*(1), 338-355.

[Camras & Chen, 2006] Camras, L. A., & Chen, Y. (2006). Culture, ethnicity, and children's facial expressions: A study of European American, mainland Chinese, Chinese American, and adopted Chinese girls. *6*(1), 103–114.

[Chow et al., 2005] Chow, S. M., Ram, N., Boker, S. M., Fujita, F., & Clore, G. C. (2005). Emotion as a thermostat: representing emotion regulation using a damped oscillator model. *Emotion, 5*(2), 208-225.

[Cohn et al., In press] Cohn, J. F., Ambadar, Z., & Ekman, P. (In press). Observer-based measurement of facial expression with the Facial Action Coding System. In J. A. Coan & J. B. Allen (Eds.), *The handbook of emotion elicitation and assessment. Oxford University Press Series in Affective Science*. New York, NY: Oxford University.

[Cohn & Campbell, 1992] Cohn, J. F., & Campbell, S. B. (1992). Influence of maternal depression on infant affect regulation. In D. Cicchetti & S. L. Toth (Eds.), *Developmental perspectives on depression* (pp. 103-130). Rochester, New York: University of Rochester Press.

[Cohn & Ekman, 2005] Cohn, J. F., & Ekman, P. (2005). Measuring facial action by manual coding, facial EMG, and automatic facial image analysis. In J. A. Harrigan, R. Rosenthal & K. Scherer (Eds.), *Handbook of nonverbal behavior research methods in the affective sciences* (pp. 9-64). New York: Oxford.

[Cohn & Elmore, 1988] Cohn, J. F., & Elmore, M. (1988). Effects of contingent changes in mothers' affective expression on the organization of behavior in 3-month old infants. *Infants Behavior and Development, 11*, 493-505.

[Cohn & Kanade, In press] Cohn, J. F., & Kanade, T. Iin press). Use of automated facial image analysis for measurement of emotion expression. In J. A. Coan & J. B. Allen (Eds.), *The handbook of emotion elicitation and assessment*. New York, NY: Oxford.

[Cohn et al., 2004] Cohn, J. F., Reed, L. I., Moriyama, T., Xiao, J., Schmidt, K. L., & Ambadar, Z. (2004). *Multimodal coordination of facial action, head rotation, and eye motion.* Paper presented at the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, Korea.

[Cohn & Schmidt, 2004] Cohn, J. F., & Schmidt, K. L. (2004). The timing of facial motion in posed and spontaneous smiles. *International Journal of Wavelets, Multiresolution and Information Processing, 2*, 1-12.

[Cohn et al., 2002] Cohn, J. F., Schmidt, K. L., Gross, R., & Ekman, P. (2002). *Individual differences in facial expression: Stability over time, relation to self-reported emotion, and ability to inform person identification.* Paper presented at the International Conference on Multimodal User Interfaces, Pittsburgh, PA.

[Cohn & Tronick, 1987] Cohn, J. F., & Tronick, E. Z. (1987). Mother infant interaction: The sequence of dyadic states at three, six, and nine months. *Developmental Psychology, 23*, 68 77.

[Cohn & Tronick, 1988] Cohn, J. F., & Tronick, E. Z. (1988). Mother-Infant face-to-face interaction: Influence is bidirectional and unrelated to periodic cycles in either partner's behavior. *Developmental Psychology, 34*(3), 386-392.

[Darwin, 1872/1998] Darwin, C. (1872/1998). *The expression of the emotions in man and animals (3rd Edition).* New York, New York: Oxford University.

[Davidson et al., 1990] Davidson, R. J., Ekman, P., Saron, C. D., Senulis, J. A., & Friesen, W. V. (1990). Approach-withdrawal and cerebral asymmetry: Emotional expression and brain physiology I. *Journal of Personality and Social Psychology, 58*(2), 330-341.

[Dimberg et al., 2002] Dimberg, U., Thunberg, M., & Grunedal, S. (2002). Facial reactions to emotional stimuli: Automatically controlled emotional responses. *Cognition and Emotion, 16*(4), 449–471.

[Edwards, 1998] Edwards, K. (1998). The face of time: Temporal cues in facial expressions of emotion. *Psychological Science, 9*(4), 270-276.

[Eibl-Eibesfeldt, 1989] Eibl-Eibesfeldt, I. (1989). *Human ethology*. NY, NY: Aldine de Gruvier.

[Ekman, 1992a] Ekman, P. (1992a). Are there basic emotions? *Psychological Review, 99*(3), 550-553.

[Ekman, 1992b] Ekman, P. (1992b). An argument for basic emotions. *Cognition and Emotion, 6*(3/4), 169-200.

[Ekman, 2003] Ekman, P. (2003). *Emotions revealed.* New York, NY: Times.

[Ekman & Friesen, 1978] Ekman, P., & Friesen, W. V. (1978). *Facial action coding system*. Palo Alto, CA: Consulting Psychologists Press.

[Ekman et al., 2002] Ekman, P., Friesen, W. V., & Hager, J. C. (Eds.). (2002). *Facial action coding system*: Research Nexus, Network Research Information, Salt Lake City, UT.

[Ekman et al., 1988] Ekman, P., Friesen, W. V., & O'Sullivan, M. (1988). Smiles when lying. *Journal of Personality and Social Psychology, 54*(3), 414-420.

[Ekman & Rosenberg, 2005] Ekman, P., & Rosenberg, E. (Eds.). (2005). *What the face reveals* (2nd ed.). New York: Oxford.

[Frank & Ekman, 1997] Frank, M. G., & Ekman, P. (1997). The ability to detect deceit generalizes across different types of high-stakes lies. *Journal of Personality and Social Psychology, 72*(6), 1429-1439.

[Fridlund et al., 1990] Fridlund, A. J., Sabini, J. P., Hedlund, L. E., Schaut, J. A., Shenker, J. J., & Knauer, M. J. (1990). Audience effects on solitary faces during imagery: Displaying to the people in your head. *Journal of Nonverbal Behavior, 14*(2), 113-137.

[Gottman et al., 2001] Gottman, J., Levenson, R., & Woodin, E. (2001). Facial expressions during marital conflict *Journal of Family Communication, 1*(1), 37-57.

[Haidt & Keltner, 1999] Haidt, J., & Keltner, D. (1999). Culture and facial expression: Open-ended methods find more expressions and a gradient of recognition. *Emotion and Cognition, 13*(3), 225-266.

[Ibanez et al., 2006] Ibanez, L., Messinger, D., Ambadar, Z., & Cohn, J. F. (2006). *Automated measurement of infant and mother interactive smiling.* Paper presented at the American Psychological Society.

[Jaffe et al., 2001] Jaffe, J., Beebe, B., Feldstein, S., Crown, C. L., & Jasnow, M. (2001). Rhythms of dialogue in early infancy. *Monographs of the Society for Research in Child Development, 66*(2, Serial No. 264).

[Kanade et al., 2005] Kanade, T., Hu, C., & Cohn, J. F. (2005). *Facial expression analysis.* Paper presented at the IEEE International Workshop on Modeling and Analysis of Faces and Gestures, Beijing, China.

[Keltner, 1995] Keltner, D. (1995). Signs of appeasement: Evidence for the distinct displays of embarrassment, amusement and shame. *Journal of Personality and Social Psychology, 68*(3), 441-454.

[Keltner & Ekman, 2000] Keltner, D., & Ekman, P. (2000). Facial expression of emotion. In M. Lewis & J. M. Haviland (Eds.), *Handbooks of emotions* (second ed., pp. 236-249). New York: Guilford.

[Keltner & Haidt, 1999] Keltner, D., & Haidt, J. (1999). Social functions of emotions at multiple levels of analysis. *Cognition and Emotion, 13*(5), 505-522.

[Krumhuber & Kappas, 2005] Krumhuber, E., & Kappas, A. (2005). Moving smiles: The role of dynamic components for the perception of the genuineness of smiles. *Journal of Nonverbal Behavior, 29*, 3-24.

[Lazarus, 1991] Lazarus, R. S. (1991). *Emotion and adaptation*. NY: Oxford.

[Levenson et al., 1990] Levenson, R. W., Ekman, P., & Friesen, W. V. (1990). Voluntary facial action generates emotion-specific autonomic nervous system activity. *Psychophysiology, 27*(4), 363-384.

[Malatesta et al., 1989] Malatesta, C. Z., Culver, C., Tesman, J. R., & Shephard, B. (1989). The development of emotion expression during the first two years of life. *Monographs of the Society for Research in Child Development, 54*(219).

[Malatesta & Haviland, 1982] Malatesta, C. Z., & Haviland, J. M. (1982). Learning display rules: The socialization of emotion expression in infancy. *Child Development, 53*, 991-1003.

[Markus & Kitayama, 1991] Markus, H. R., & Kitayama, S. (1991). Culture and the self: Implications for cognition, emotion, and motivation. *Psychological Review, 98*, 224-253.

[Matsumoto & Willingham, 2006] Matsumoto, D., & Willingham, B. (2006). The thrill of victory and the agony of defeat: spontaneous expressions of medal winners of the 2004 Athens olympic games. *Journal of Personality & Social Psychology, 91*(5), 568–581.

[Messinger et al., 2006] Messinger, D. S., Chow, S. M., Koterba, S., Hu, C., Haltigan, J. D., Wang, T., et al. (2006). *Continuously measured smile dynamics in infant-mother interaction.*Unpublished manuscript, Miami.

[Moore et al., 1997] Moore, G. A., Cohn, J. F., & Campbell, S. B. (1997). Mothers' affective behavior with infant siblings: Stability and change. *Developmental Psychology., 33*, 856-860.

[Oster et al., 1996] Oster, H., Camras, L. A., Campos, J., Campos, R., Ujiee, T., Zhao-Lan, M., et al. (1996). The patterning of facial expressions in Chinese, Japanese, and American infants in fear- and anger- eliciting situations. Poster presented at the International Conference on Infant Studies, Providence, RI.

[Oster et al., 1992] Oster, H., Hegley, D., & Nagel, L. (1992). Adult judgments and fine-grained analysis of infant facial expressions: Testing the validity of a priori coding formulas. *Developmental Psychology, 28*(6), 1115-1131.

[Pantic & Patras, 2006] Pantic, M., & Patras, I. (2006). Dynamics of facial expressions: Recognition of facial actions and their temporal segments from profile image sequences. *IEEE Tansactions on Systems, Man, and Cybernetics, Part B, 36*(2), 443-449.

[Pantic & Rothkrantz, 2000] Pantic, M., & Rothkrantz, M. (2000). Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*, 1424-1445.

[Pantic & Rothkrantz, 2003] Pantic, M., & Rothkrantz, M. (2003). Toward an affect-sensitive multimodal human-computer interaction. *Proceedings of the IEEE, 91*, 1371-1390.

[Pantic et al., 2005] Pantic, M., Sebe, N., Cohn, J. F., & Huang, T. S. (2005). *Affective multimodal human-computer interaction.* Paper presented at the ACM International Conference on Multimedia.

[Prkachin, 1992] Prkachin, K. M. (1992). The consistency of facial expressions of pain. *Pain, 51*, 297-306.

[Sayette et al., 2003] Sayette, M., Wertz, J., Martin, C. S., Cohn, J. F., Perrott, M., & Hobel, J. (2003). Effects of smoking opportunity on cue-elicited urge: A facial coding analysis. . *Experimental and Clinical Psychopharamacology, 11*, 218-227.

[Schmidt & Cohn, 2001] Schmidt, K. L., & Cohn, J. F. (2001). Human facial expressions as adaptations: Evolutionary perspectives in facial expression research. *Yearbook of Physical Anthropology, 116*, 8-24.

[Tian et al., 2005] Tian, Y., Cohn, J. F., & Kanade, T. (2005). Facial expression analysis. In S. Z. Li & A. K. Jain (Eds.), *Handbook of face recognition* (pp. 247-276). New York, New York: Springer.

[Tsai et al., 2006] Tsai, J. L., Levenson, R. W., & McCoy, K. (2006). Cultural and temperamental variation in emotional response. *Emotion, 6*(3), 484-497.

[Valstar et al., 2006] Valstar, M., Pantic, M., Ambadar, Z., & Cohn, J. F. (2006, November). *Spontaneous vs. posed facial behavior.* Paper presented at the ACM International Conference on Multimodal Interfaces, Banff, Canada.

[Watson & Tellegen, 1985] Watson, D., & Tellegen, A. (1985). Toward a consensual structure of mood. *Psychological Bulletin, 98*(2), 219-235.

[Zlochower & Cohn, 1996] Zlochower, A. J., & Cohn, J. F. (1996). Vocal timing in face-to-face interaction of clinically depressed and nondepressed mothers and their 4-month-old infants. *Infant Behavior and Development, 19*, 373-376.

Machine Understanding of Human Behavior*

Maja Pantic[1,3], Alex Pentland[2], Anton Nijholt[3] and Thomas Huang[4]

[1]Computing Department, Imperial College London, UK
[2]Media Lab, Massachusetts Institute of Technology, USA
[3]Faculty of EEMCS, University of Twente, The Netherlands
[4]Beckman Institute, University of Illinois at Urbana-Champaign, USA
m.pantic@imperial.ac.uk, pentland@media.mit.edu, a.nijholt@ewi.utwente.nl, huang@ifp.uiuc.edu

Abstract

A widely accepted prediction is that computing will move to the background, weaving itself into the fabric of our everyday living spaces and projecting the human user into the foreground. If this prediction is to come true, then next generation computing, which we will call *human computing*, should be about anticipatory user interfaces that should be human-centered, built for humans based on human models. They should transcend the traditional keyboard and mouse to include natural, human-like interactive functions including understanding and emulating certain human behaviors such as affective and social signaling. This article discusses a number of components of human behavior, how they might be integrated into computers, and how far we are from realizing the front end of human computing, that is, how far are we from enabling computers to understand human behavior.

1 Human Computing

Futuristic movies often contain visions of human environments of the future. Fitted out with arrays of intelligent, yet invisible devices, homes, transportation means and working spaces of the future can anticipate every need of their inhabitants (Fig. 1). This vision of the future is often referred to as “ubiquitous computing” [Weiser, 1991] or “ambient intelligence” [Aarts, 2005] . In this vision of the future, humans will be surrounded by intelligent interfaces that are supported by computing and networking technology embedded in all kinds of objects in the environment and that are sensitive and responsive to the presence of different individuals in seamless and unobtrusive way. This assumes a shift in computing – from desktop computers to a multiplicity of smart computing devices diffused into our environment. It assumes that computing will move to the background, weave itself into the fabric of everyday living spaces and disappear from the foreground, projecting the human user into it. However, as computing devices disappear from the scene, become invisible, weaved into our environment, a new set of issues is created concerning the interaction between this technology and humans [Nijholt et al., 2004, 2005, 2006; Streitz and Nixon, 2005; Zhai and Bellotti, 2005]. How can we design the interaction of humans with devices that are invisible? How can we design implicit interaction for sensor-based interfaces? What about users? What does a home dweller, for example, actually want? What are the relevant parameters that can be used by the systems to support us in our activities? If the context is key, how do we arrive at context-aware systems?

One way of tackling these problems is to move away from computer-centered designs toward human-centered designs for human computer interaction (HCI). The former involve usually the conventional interface devices like keyboard, mouse, and visual displays, and assume that the human will be explicit, unambiguous and fully attentive while controlling information and command flow. This kind of interfacing and categorical computing works well for context-independent tasks like making plane reservations and buying and selling stocks. However, it is utterly inappropriate for interacting with each of the (possibly hundreds) computer systems diffused throughout future smart environments and aimed at improving the quality of life by anticipating the users needs. The key to *human computing* and *anticipatory interfaces* is the ease of use, in this case the ability to unobtrusively sense certain behavioral cues of the users and to adapt automatically to his or hers typical behavioral patterns and the context in which he or she acts. Thus, instead of focusing on the computer portion of the HCI context, designs for human computing should focus on the human portion of the HCI context. They should go beyond the traditional keyboard and mouse to include natural, human-like interactive functions including understanding and emulating certain human behaviors like affective and social signaling. The design of these functions will require explorations of *what* is communicated (linguistic message, nonlin-

* This paper was originally published in the Proc. ACM Int'l Conf. Multimodal Interface 2006 (Copyright © ACM Press); see [Pantic et al., 2006]. The work of Maja Pantic and Anton Nijholt was partly supported by the EU 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811).

Fig. 1. Human environments of the future envisioned in movies: (left) hand-gesture-based interface and speech- & iris-id driven car (*Minority Report*, 2002), (right) multimedia diagnostic chart and a smart environment (*The Island*, 2005).

guistic conversational signal, emotion, attitude), *how* the information is passed on (the person's facial expression, head movement, nonlinguistic vocalization, hand and body gesture), *why*, that is, in which context the information is passed on (where the user is, what his or her current task is, are other people involved), and *which* (re)action should be taken to satisfy user needs and requirements.

This article discusses the front end of human computing, that is, what is communicated, how, and why [Pantic et al., 2006]. It focuses on certain human behaviors such as affective and social signaling, how they might be understood by computers, and how far we are from realizing the front end of human computing. For discussions about the back end of human computing, readers are referred to, e.g., [Nijholt et al., 2006; Ruttkay, 2006; Maat and Pantic, 2006].

2 Scientific and Engineering Issues

The scientific and engineering challenges related to the realization of machine sensing and understanding of human behaviors like affective and social signaling can be described as follows.

- ♦ **Which types of messages are communicated by behavioral signals?** This question is related to psychological issues pertaining to the nature of behavioral signals and the best way to interpret them.
- ♦ **Which human communicative cues convey information about a certain type of behavioral signals?** This issue shapes the choice of different modalities to be included into an automatic analyzer of human behavioral signals.
- ♦ **How are various kinds of evidence to be combined to optimize inferences about shown behavioral signals?** This question is related to issues such as how to distinguish between different types of messages, how best to integrate information across modalities, and what to take into account in order to realize context-aware interpretations.

Which types of messages are communicated by behavioral signals? The term behavioral signal is usually used to describe a set of temporal changes in neuromuscular and physiological activity that can last from a few milliseconds (a blink) to minutes (talking) or hours (sitting). Among the types of messages conveyed by behavioral signals are the following [Ekman and Friesen, 1969] (Fig. 2):

- ♦ affective/attitudinal states (e.g. fear, joy, inattention, stress),
- ♦ manipulators (actions used to act on objects in the environment or self-manipulative actions like scratching and lip biting),
- ♦ emblems (culture-specific interactive signals like wink or thumbs up),
- ♦ illustrators (actions accompanying speech such as finger pointing and raised eyebrows),
- ♦ regulators (conversational mediators such as the exchange of a look, palm pointing, head nods and smiles).

While there is agreement across different theories that at least some behavioral signals evolved to communicate information, there is lack of consensus regarding their specificity, extent of their innateness and universality, and whether they convey emotions, social motives, behavioral intentions, or all three [Izard, 1997]. Arguably the most often debated issue is whether affective states are a separate type of messages communicated by behavioral signals (i.e. whether behavioral signals communicate actually felt affect), or is the related behavioral signal (e.g. facial expression) just an illustrator / regulator aimed at controlling "the trajectory of a given social interaction", as suggested by Fridlund [1997]. Explanations of human behavioral signals in terms of internal states such as affective states are typical to psychological stream of thought, in particular to discrete emotion theorists who propose the existence of six or more basic emotions (happiness, anger, sadness, surprise, disgust, and fear) that are universally displayed and recognized from non-verbal behavioral signals (especially facial and vocal expression) [Keltner and Ekman, 2000; Juslin and Scherer, 2005]. Instead of explanations of human behavioral signals in terms of internal states, ethologists focus on consequences of behavioral displays for interpersonal interaction. As an extreme within the ethological line of thought, social constructivists argue that emotions are socially constructed ways of interpreting and responding to particular classes of situations. According to Fridlund, facial expressions should not be labeled in terms of emotions but in terms of Behavioral Ecology interpretations, which explain the influence a certain expression has in a particular context [Fridlund, 1997]. Thus, an "angry" face should not be interpreted as *anger* but as *back-off-or-I-will-attack*. However, as proposed by Izard [1997], one may feel angry without the slightest intention of attacking anyone. In summary, is social communication the sole function of behavioral signals? Do they never represent visible manifestation of emotion / feeling / affective states? Since in some instances (e.g.

arachnophobia, acrophobia, object-elicited disgust, depression), affective states are not social, and their expressions necessarily have aspects other than “social motivation”, we believe that affective states should be included into the list of types of messages communicated by behavioral signals. However, it is not only discrete emotions like surprise or anger that represent the affective states conveyed by human behavioral signals. Behavioral cues identifying attitudinal states like interest and boredom, to those underlying moods, and to those disclosing social signaling like empathy and antipathy are essential components of human behavior. Hence, in contrast to traditional approach, which lists only (basic) emotions as the first type of message conveyed by behavioral signals [Ekman and Friesen, 1969], we treat affective states as being correlated not only to emotions but to other, aforementioned social signals and attitudinal states as well.

Which human communicative cues convey information about a certain type of behavioral signals? Manipulators are usually associated with self-manipulative gestures like scratching or lip biting and involve facial expressions and body gestures human communicative cues. Emblems, illustrators and regulators are typical social signals, spoken and wordless messages like head nods, bow ties, winks, ‘huh’ and ‘yeah’ utterances, which are sent by means of body gestures and postures, facial expressions and gaze, vocal expressions and speech. The most complex messages communicated by behavioral signals are affective and attitudinal states. Affective arousal modulates all human communicative signals. Hence, one could expect that automated analyzers of human behavior should include all human interactive modalities (audio, visual, and tactile) and should analyze all verbal and non-verbal interactive signals (speech, body gestures, facial and vocal expressions, and physiological reactions). However, we would like to make a few comments here. Although spoken language is between 200 thousand and 2 million years old [Gibson and Ingold, 1993], and speech has become the indispensable means for sharing ideas, observations, and feelings, findings in basic research indicate that in contrast to spoken messages [Furnas et al., 1987], nonlinguistic messages are the means to analyze and predict human behavior [Ambady and Rosenthal, 1992]. Anticipating a person’s word choice and the associated intent is very difficult [Furnas et al., 1987]: even in highly constrained situations, different people choose different words to express exactly the same thing. As far as nonverbal cues are concerned, it seems that not all of them are equally important in the human judgment of behavioral signals. People commonly neglect physiological signals, since they cannot sense them at all times. Namely, in order to detect someone’s clamminess or heart rate, the observer should be in a physical contact (touch) with the observed person. Yet, the research in psychophysiology has produced firm evidence that affective arousal has a range of somatic and physiological correlates including pupillary diameter, heart rate, skin clamminess, temperature, respiration velocity [Cacioppo et al., 2000]. This and the recent advent of non-intrusive sensors and wearable computers, which promises less invasive physiological sensing [Starner, 2001], open up possibilities for including tactile modality into automatic analyzers of human behavior [Pentland, 2005]. However, the visual channel carrying facial expressions and body gestures seems to be most important in the human judgment of behavioral cues [Ambady and Rosenthal, 1992]. Human judges seem to be most accurate in their judgment when they are able to observe the face and the body. Ratings that were based on the face and the body were 35% more accurate than the ratings that were based on the face alone. Yet, ratings that were based on the face alone were 30% more accurate than ratings that were based on the body alone and 35% more accurate than ratings that were based on the tone of voice alone [Ambady and Rosenthal, 1992]. These findings indicate that to interpret someone’s behavioral cues, people rely on shown facial expressions and to a lesser degree on shown body gestures and vocal expressions. Note, however, that gestures like (Fig. 2) scratching (manipulator), thumbs up (emblem), finger pointing (illustrator), and head nods (regulator) are typical social signals. Basic research also provides evidence that observers tend to be accurate in decoding some negative basic emotions like anger and sadness from static body postures [Coulson, 2004] and that gestures like head inclination, face touching, and shifting posture often accompany social affective states like shame and embarrassment [Costa et al., 2001]. In addition, although cognitive scientists were unable to identify a set of vocal cues that reliably discriminate among affective and attitudinal states, listeners seem to be rather accurate in decoding some basic emotions from vocal cues like pitch and intensity [Juslin and Scherer, 2005] and some non-basic affective states such as distress, anxiety, boredom, and sexual interest from nonlinguistic vocalizations like laughs, cries, sighs, and yawns [Russell et al., 2003]. Thus, automated human behavior analyzers should at

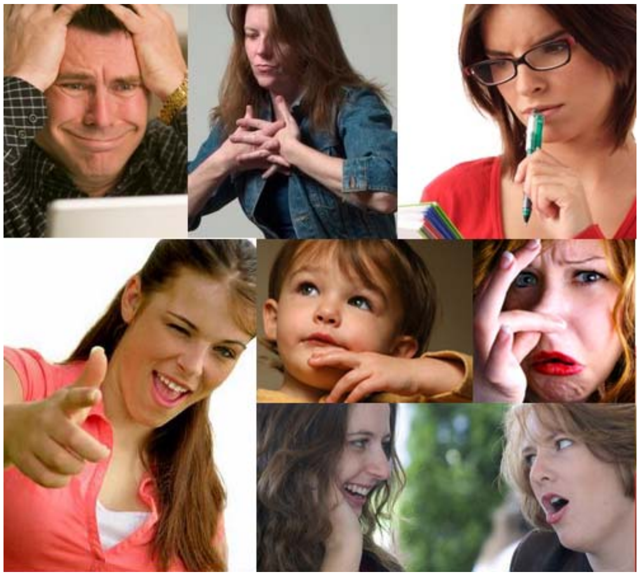


Fig. 2. Types of messages conveyed by behavioural signals: (1st row): affective/attitudinal states, (2nd row, clockwise from left) emblems, manipulators, illustrators, regulators.

least include facial expression and body gestures modalities and preferably they should also include modality for perceiving nonlinguistic vocalizations. Finally, while too much information from different channels seem to be confusing to human judges, resulting in less accurate judgments of shown behavior when three or more observation channels are available (face, body, and speech) [Ambady and Rosenthal, 1992], combining those multiple modalities (including physiology) may prove appropriate for realization of automatic human behavior analysis.

How are various kinds of evidence to be combined to optimize inferences about shown behavioral signals? Behavioral signals do not usually convey exclusively one type of messages but may convey any of the types (e.g. scratching is usually a manipulator but it may be displayed in an expression of confusion). It is crucial to determine to which class of behavioral signals a shown signal belongs since this influences the interpretation of it. For instance, squinted eyes may be interpreted as sensitivity of the eyes to bright light if this action is a reflex (a manipulator), as an expression of disliking if this action has been displayed when seeing someone passing by (affective cue), or as an illustrator of friendly anger on friendly teasing if this action has been posed (in contrast to being unintentionally displayed) during a chat with a friend, to mention just a few possibilities. To determine the class of an observed behavioral cue, one must know the context in which the observed signal has been displayed – where the expresser is (outside, inside, in the car, in the kitchen, etc.), what his or her current task is, are other people involved, and who the expresser is. The latter is of particular importance for recognition of affective and attitudinal states since it is not probable that each of us will express a particular affective state by modulating the same communicative signals in the same way, especially when it comes to affective states other than basic emotions. Since the problem of context-sensing is extremely difficult to solve (if possible at all) for a general case, we advocate that a pragmatic approach (e.g. activity/application- and user-centered approach) must be taken when learning the grammar of human expressive behavior. In addition, because of the impossibility of having users instructing the computers for each possible application, we propose that methods for unsupervised (or semi-supervised) learning must be applied. Moreover, much of human expressive behavior is unintended and unconscious; the expressive nonverbal cues can be so subtle that they are neither encoded nor decoded at an intentional, conscious level of awareness [Ambady and Rosenthal, 1992]. This suggests that the learning methods inspired by human unconscious problem solving processes may prove more suitable for automatic human behavior analysis than the learning methods inspired by human conscious problem solving processes [Valstar and Pantic, 2006a]. Another important issue is that of multimodal fusion. A number of concepts relevant to fusion of sensory neurons in humans may be of interest [Stein and Meredith, 1993]:

- *1+1 >2*: The response of multi-sensory neurons can be stronger for multiple weak input signals than for a single strong signal.
- *Context dependency*: The fusion of sensory signals is modulated depending on the sensed context – for different contexts, different combinations of sensory signals are made.
- *Handling of discordances*: Based on the sensed context, sensory discordances (malfunctioning) are either handled by fusing sensory signals without any regard for individual discordances (e.g. when a fast response is necessary), or by attempting to recalibrate discordant sensors (e.g. by taking a second look), or by suppressing discordant and recombining functioning sensors (e.g. when one observation is contradictory to another).

Thus, humans simultaneously employ the tightly coupled audio, visual, and tactile modalities. As a result, analysis of the perceived information is highly robust and flexible. Hence, one could expect that in an automated analyzer of human behavior input signals should not be considered mutually independent and should not be combined only at the end of the intended analysis, as the majority of current studies do, but that they should be processed in a joint feature space and according to a context-dependent model [Pantic and Rothkrantz, 2003]. However, does this tight coupling persists when the modalities are used for multimodal interfaces as proposed by some researchers (e.g. [Gunes and Piccardi, 2005]), or not, as suggested by others (e.g. [Scanlon and Reilly, 2001])? This remains an open, highly relevant issue.

3 State of the Field

Human sensing: Sensing human behavioral signals including facial expressions, body gestures, nonlinguistic vocalizations, and vocal intonations, which seem to be most important in the human judgment of behavioral cues [Ambady and Rosenthal, 1992], involves a number of tasks.

- Face: face detection and location, head and face tracking, eye-gaze tracking, and facial expression analysis.
- Body: body detection and tracking, hand tracking, recognition of postures, gestures and activity.
- Vocal nonlinguistic signals: estimation of auditory features such as pitch, intensity, and speech rate, and recognition of nonlinguistic vocalizations like laughs, cries, sighs, and coughs.

Because of its practical importance and relevance to face recognition, face detection received the most attention of the tasks mentioned above. Numerous techniques have been developed for face detection, i.e., identification of all regions in the scene that contain a human face [Yang et al., 2002; Li and Jain, 2005]. However, virtually all of them can detect only (near-) upright faces in (near-) frontal view. Most of these methods emphasize statistical learning techniques and use appearance features, including the real-time face detection scheme proposed by Viola and Jones [2004], which is arguably the most commonly employed face de-

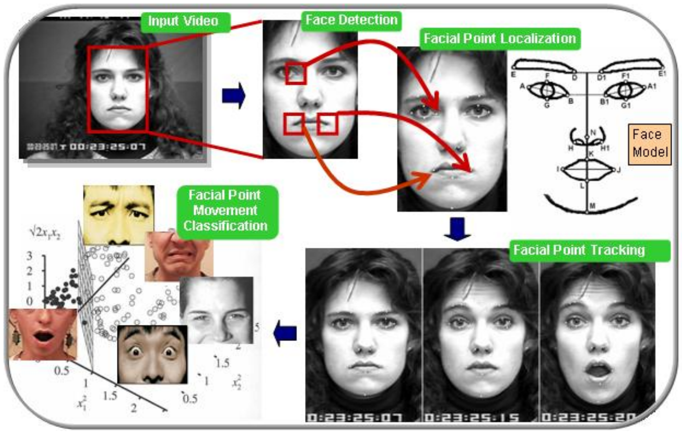


Fig. 3. An AU detection method [Valstar & Pantic, 2006b].

tector in automatic facial expression analysis. Note, however, that one of the few methods that can deal with tilted face images represents a feature-based rather than an appearance-based approach to face detection [Chiang and Huang, 2005].

Tracking is an essential step for human motion analysis since it provides the data for recognition of face/head/body postures and gestures. Optical flow has been widely used for head, face and facial feature tracking [Wang and Singh, 2003]. To omit the limitations inherent in optical flow techniques such as the accumulation of error and the sensitivity to occlusion, clutter, and changes in illumination, researchers in the field started to use sequential state estimation techniques like Kalman and particle filtering schemes [Haykin and Freitas, 2004]. Some of the most advanced approaches to head tracking and head-pose estimation are based on Kalman (e.g. [Huang and Trivedi, 2004]) and particle filtering frameworks (e.g. [Ba and Odobez, 2004]). Similarly, the most advanced approaches to facial feature tracking are based on Kalman (e.g. [Gu and Ji, 2005]) and particle filtering tracking schemes (e.g. [Valstar and Pantic, 2006b]). Although face pose and facial feature tracking technologies have improved significantly in the recent years with sequential state estimation approaches that run in real time, tracking multiple, possibly occluded, expressive faces, their poses, and facial feature positions simultaneously in unconstrained environments is still a difficult problem. The same is true for eye gaze tracking [Duchowski, 2002]. To determine the direction of the gaze, eye tracking systems employ either the so-called red-eye effect, i.e., the difference in reflection between the cornea and the pupil, or computer vision techniques to find the eyes in the input image and then determine the orientation of the irises. Although there are now several companies that sell commercial eye trackers like SMI GmbH, EyeLink, Tobii, Interactive Minds, etc., realizing non-intrusive (non-wearable), fast, robust, and accurate eye tracking remains a difficult problem even in computer-centred HCI scenarios in which the user is expected to remain in front of the computer but is allowed to shift his or her position in any direction for more than 30 cm.

Because of the practical importance of the topic for affective, perceptual, and ambient interfaces of the future and theoretical interest from cognitive scientists [Lisetti and Schiano, 2000; Pantic and Rothkrantz, 2003], automatic analysis of facial expressions attracted the interest of many researchers. Most of the facial expressions analyzers developed so far attempt to recognize a small set of prototypic emotional facial expressions such as happiness or sadness (see also the state of the art in facial affect recognition in the text below) [Pantic and Rothkrantz, 2003]. To facilitate detection of subtle facial signals like a frown or a smile and to make facial expression information available for usage in applications like anticipatory ambient interfaces, several research groups begun research on machine analysis of facial muscle actions (atomic facial cues, action units, AUs, [Ekman et al., 2002]). A number of promising prototype systems have been proposed recently that can recognize 15 to 27 AUs (from a total of 44 AUs) in either (near-) frontal view or profile view face image sequences [Li and Jain, 2005; Pantic and Patras, 2006]. Most of these employ statistical and ensemble learning techniques and are either feature-based (i.e., use geometric features like facial points or shapes of facial components, e.g., see Fig. 3) or appearance-based (i.e., use texture of the facial skin including wrinkles, bulges, and furrows). It has been reported that methods based on appearance features usually outperform those based on geometric features. Recent studies have shown that this claim does not always hold [Pantic and Patras, 2006]. Besides, it seems that using both geometric and appearance features might be the best choice for certain facial cues [Pantic and Patras, 2006]. However, the present systems for facial AU detection typically depend on accurate head, face and facial feature tracking as input and are still very limited in performance and robustness.

Vision-based analysis of hand and body gestures is nowadays one of the most active fields in computer vision. Tremendous amount of work has been done in the field in the recent years [Wang and Singh, 2003; Wang et al., 2003]. Most of the proposed techniques are either model-based (i.e., use geometric primitives like cones and spheres to model head, trunk, limbs and fingers) or appearance-based (i.e., use color or texture information to track the body and its parts). Most of these methods emphasize Gaussian models, probabilistic learning, and particle filtering framework (e.g. [Sand and Teller, 2006; Stenger et al., 2006]. However, body and hands detection and tracking in unconstrained environments where large changes in illumination and cluttered or dynamic background may occur still pose significant research challenges. Also, in casual human behavior, the hands do not have to be always visible (in pockets, under the arms in a crossed arms position, on the back of the neck and under the hair), they may be in a cross fingered position, and one hand may be (partially) occluded by the other. Although some progress has been made to tackle these problems using the knowledge on human kinematics, most of the present methods cannot handle such cases correctly.

In contrast to the linguistic part of a spoken message (*what* has been said) [Furnas et al., 1987], the nonlinguistic part of it (*how* it has been said) carries important information about the speaker's affective state [Juslin and Scherer, 2005] and attitude [Russell et al., 2003]. This finding instigated the research on automatic analysis of vocal nonlinguistic expressions. The vast majority of present work is aimed at discrete emotion recognition from auditory features like pitch, intensity, and speech rate (see the state of the art in vocal affect recognition in the text below) [Oudeyer, 2003; Pantic and Rothkrantz, 2003]. For the purposes of extracting auditory features from input audio signals, freely available signal processing toolkits like Praat[1] are usually used. More recently, few efforts towards automatic recognition of nonlinguistic vocalizations like laughs [Truong and van Leeuwen, 2005], cries [Pal et al., 2006], and coughs [Matos et al., 2006] have been also reported. Since the research in cognitive sciences provided some promising hints that vocal outbursts and nonlinguistic vocalizations like yelling, laughing, and sobbing, may be very important cues for decoding someone's affect/attitude [Russell et al., 2003], we suggest a much broader focus on machine recognition of these nonlinguistic vocal cues.

Context sensing: Context plays a crucial role in understanding of human behavioral signals, since they are easily misinterpreted if the information about the situation in which the shown behavioral cues have been displayed is not taken into account [Pantic and Rothkrantz, 2003]. For computing technology applications, context can be defined as any information that can be used to characterize the situation that is relevant to the interaction between users and the application [Dey et al., 2001]. Six questions summarize the key aspects of the computer's context with respect to nearby humans:

- *Who?* (Who the user is?)
- *Where?* (Where the user is?)
- *What?* (What is the current task of the user?)
- *How?* (How the information is passed on? Which behavioral signals have been displayed?)
- *When?* (What is the timing of displayed behavioral signals with respect to changes in the environment? Are there any co-occurrences of the signals?)
- *Why?* (What may be the user's reasons to display the observed cues? Except of the user's current task, the issues to be considered include the properties of the user's physical environment like lighting and noise level, and the properties of the current social situation like whether the user is alone and what is his or her psychological state.)

Here, we focus on answering context questions relating to the human-part of the computer's context. The questions related exclusively to the user's context and not to the computer's context like what kind of people are the user's communicators and what the overall social situation is, are considered irrelevant for adapting and tailoring the computing technology to its human users and are not discussed in this article.

Because of its relevance for the security, the *who* context question has received the most attention from both funding agencies and commercial enterprises and, in turn, it has seen the most progress. The biometrics market has increased dramatically in recent years, with multiple companies providing face recognition systems like Cognitec and Identix, whose face recognition engines achieved repeatedly top 2D face recognition scores in USA government testing (FRGC, FRVT 2002, FERET 1997). The problem of face recognition has been tackled in various ways in 2D and 3D, using feature-, shape-, and appearance-based approaches as well as the combinations thereof [Zhao et al., 2003; Li and Jain, 2005; Bowyer et al., 2006]. The majority of the present methods employ spectral methods for dimensionality reduction like PCA, LDA, and ICA. Except of the face, biometric systems can be based on other biometric traits like fingerprints, voice, iris, retina, gait, ear, hand geometry, and facial thermogram [Jain and Ross, 2004]. Biometric systems should be deployed in real-world applications and, in turn, should be able to handle a variety of problems including sensor malfunctioning, noise in sensed data, intra-class variations (e.g. facial expression which is treated as noise in face recognition), and spoof attacks (i.e. falsification attempts). Since most of these problems can be overcome by using multiple biometric traits [Jain and Ross, 2004], multimodal biometric systems have recently become a research trend. The most commonly researched multi-biometrics relate to audiovisual speaker recognition. For a survey of commercial systems for alternative biometrics, see [BTT Survey, 2006]. For current research efforts in multi-biometrics, see [MMUA, 2006].

Similarly to the *who* context question, security concerns also drive the research tackling the *where* context-sensing problem, which is typically addressed as a computer-vision problem of surveillance and monitoring. The work in this area is based on one or more unobtrusively mounted cameras used to detect and track people. The process usually involves [Wang et al., 2003]: scene (background) modeling, motion segmentation, object classification, and object tracking. The vast majority of scene modeling approaches can be classified as generative models [Buxton, 2003]. However, generative approaches, which require excessive amount of training data, are not appropriate for complex and incomplete problem domains like dynamic scene modeling. Unsupervised learning techniques are a better choice in that case. Motion segmentation aims at detecting regions in the scene which correspond to moving objects like cars and humans. It is one of the oldest computer vision problems and it has been tackled in various ways including [Wang et al., 2003]: background subtraction, temporal differencing, optical flow, watershed, region growing, scene mosaicing, statistical and Bayesian methods. Since natural scenes may contain multiple moving regions that may correspond to different entities, it is crucial to distinguish those that correspond to humans for the purposes of sensing the human part of the com-

[1] Praat: http://www.praat.org.

puter's context. Note that this step is superfluous where the moving objects are known to be humans. Present methods to moving object classification are usually either shape-based (e.g. human-silhouette-based) or motion-based (i.e. employ the premise that human articulated motion shows a periodic property) [Wang et al., 2003]. When it comes to human tracking for the purposes of answering the *where* context question, typically employed methods emphasize probabilistic methods like Dynamic Bayesian Networks and sequential state estimation techniques like Kalman and particle filtering schemes [Wang and Singh, 2003; Wang et al., 2003]. In summary, since most approaches base their analysis on segmentation and tracking, these present methods are adequate when a priori knowledge is available (e.g. the shape of the object to be tracked), but they are weak for unconstrained environments (e.g. gym, a house party), in which multiple occlusions and clutter may be present. For such cases, methods that perform analysis at the lowest semantic level (i.e. consider only temporal pixel-based behaviour) and use unsupervised learning represent a better solution (e.g. [Bicego et al., 2006]).

In desktop computer applications, the user's task identification (i.e., the *what* context question) is usually tackled by determining the user's current focus of attention by means of gaze tracking, finger pointing, or simply based on the knowledge of current events like keystrokes, mouse movements, and active software (e.g. web browser, e-mail manager). However, as traditional HCI and usability-engineering applications involve relatively well-defined user tasks, many of the methods developed for user task analysis in typical HCI domains are inappropriate for task analysis in the context of human computing and ubiquitous, anticipatory ambient interfaces, where the tasks are often ill-defined due to uncertainty in the sensed environmental and behavioral cues. Analysis of tasks that human may carry out in the context of anticipatory ambient interfaces require adaptation and fusion of existing methods for behavioral cues recognition (e.g. hand/body gesture recognition, focus of attention identification) and those machine learning techniques that can be applicable to solving ill-structured decision-making problems (e.g. Markov decision processes and hidden-state models). However, only a very limited research has been directed to multimodal user's task identification in the context of anticipatory ambient interfaces and the majority of this work is aimed at support of military activities (e.g. airplane cockpit control) and crisis management [Sharma et al., 2003]. Other methods for human activity recognition typically identify the task of the observed person in an implicit manner, by recognizing different tasks as different activities. The main shortcoming of these approaches is the increase of the problem dimensionality – for the same activity, different recognition classes are defined, one for each task (e.g. for the sitting activity, categories like watching TV, dining, and working with desktop computer, may be defined).

The *how* context question is usually addressed as a problem of human sensing (see the state of the art in human sensing in the text above; for a survey on speech recognition see [Deng and Huang, 2004]). When it comes to desktop computer application, additional modalities like writing, keystroke (choice and rate), and mouse gestures (clicks and movements) may be considered as well when determining the information that the user has passed on.

There is now a growing body of psychological research that argues that temporal dynamics of human behavior (i.e., the timing and the duration of behavioral cues) is a critical factor for interpretation of the observed behavior [Russell et al., 2003]. For instance, it has been shown that spontaneous smiles, in contrast to volitional smiles (like in irony), are fast in onset, can have multiple AU12 apexes (i.e., multiple rises of the mouth corners), and are accompanied by other AUs that appear either simultaneously with AU12 or follow AU12 within 1s. In spite of these findings in basic research and except few studies on facial expression analysis [Valstar et al., 2006], present methods for human activity/behavior recognition do not address the *when* context question: the timing of displayed behavioral signals with respect to other behavioral signals is usually not taken into account. When it comes to the timing of shown behavioral signals with respect to changes in the environment, current methods typically approach the *when* question in an implicit way, by recognizing user's reactions to different changes in the environment as different activities.

The *why* context question is arguably the most complex and the most difficult to address context question. It requires not only detection of physical properties of the user's environment like the lighting and noise level (which can be easily determined based on the current illumination intensity and the level of auditory noise) and analysis of whether the user is alone or not (which can be carried out by means of the methods addressing the *where* context question), but understanding of the user's behavior and intentions as well (see the text below for the state of the art in human behavior understanding).

As can be seen from the overview of the current state of the art in so-called W5+ (who, where, what, when, why, how) technology, context questions are usually addressed separately and often in an implicit manner. Yet, the context questions may be more reliably answered if they are answered in groups of two or three using the information extracted from multimodal input streams. Some experimental evidence supports this hypothesis [Nock et al., 2004]. For example, solutions for simultaneous speaker identification (*who*) and location (*where*) combining the information obtained by multiple microphones and surveillance cameras had an improved accuracy in comparison to single-modal and single-aspect approaches to context sensing. A promising approach to realizing multimodal multi-aspect context-sensing has been proposed by Nock et al. [2004]. In this approach, the key is to automatically determine whether observed behavioral cues share a common cause (e.g. whether the mouth movements and audio signals complement to indicate an active known or unknown speaker (how, who, where) and whether his or her focus of attention is another person or a computer (what, why)). The main advantages of such an approach are effective handling of uncertainties due to noise in input data streams and the problem-dimensionality reduction. Therefore, we suggest a much broader focus on spatial and temporal, multimodal multi-aspect context-sensing.

Understanding human behavior: Eventually, automated human behavior analyzers should terminate their execution by translating the sensed human behavioral signals and context descriptors into a description of the shown behavior. The past work in this field can be roughly divided into the methods for understanding human affective / attitudinal states and those for understanding human social signaling (i.e., emblems, regulators, and illustrators).

Understanding Human Affect: As soon as research findings in HCI and usability engineering have suggested that HCI systems which will be capable of sensing and responding properly to human affective states are likely to be perceived as more natural, efficacious, and trustworthy, the interest in human affect machine analysis has surged. The existing body of literature in machine analysis of human affect is immense [Pantic and Rothkrantz, 2003; Oudeyer, 2003; Li and Jain, 2005]. Most of these works attempt to recognize a small set of prototypic expressions of basic emotions like happiness and anger from either face images/video or speech signal. They achieve an accuracy of 64% to 98% when detecting 3-7 emotions deliberately displayed by 5-40 subjects. However, the capabilities of these current approaches to human affect recognition are rather limited.

- Handle only a small set of volitionally displayed prototypic facial or vocal expressions of six basic emotions.
- Do not perform a context-sensitive analysis (either user-, or environment-, or task-dependent analysis) of the sensed signals.
- Do not analyze extracted facial or vocal expression information on different time scales (i.e., short videos or vocal utterances of a single sentence are handled only). Consequently, inferences about the expressed mood and attitude (larger time scales) cannot be made by current human affect analyzers.
- Adopt strong assumptions. For example, facial affect analyzers can typically handle only portraits or nearly-frontal views of faces with no facial hair or glasses, recorded under constant illumination and displaying exaggerated prototypic expressions of emotions. Similarly, vocal affect analyzers assume usually that the recordings are noise free, contain exaggerated vocal expressions of emotions, i.e., sentences that are short, delimited by pauses, and carefully pronounced by non-smoking actors.

Few exceptions from this overall state of the art in the field include a few tentative efforts to detect attitudinal and non-basic affective states such as boredom, fatigue, and pain from face video [e.g., El Kaliouby and Robinson, 2004; Bartlett et al., 2006], a few works on context-sensitive interpretation of behavioral cues like facial expressions [Pantic, 2006], and an attempt to discern spontaneous from volitionally displayed facial behavior [Valstar et al., 2006]. Few works have been also proposed that combine several modalities into a single system for human affect analysis. Although the studies in basic research suggest that the combined face and body are the most informative for the analysis of human expressive behavior [Ambady and Rosenthal, 1992], only 2-3 efforts are reported on automatic human affect analysis from combined face and body gestures [Gunes and Piccardi, 2005]. Existing works combining different modalities into a single system for human affective state analysis investigated mainly the effects of a combined detection of facial and vocal expressions of affective states [Pantic and Rothkrantz, 2003; Song et al., 2004; Zeng et al., 2006]. In general, these works achieve an accuracy of 72% to 85% when detecting one or more basic emotions from clean audiovisual input (e.g., noise-free recordings, closely-placed microphone, non-occluded portraits) from an actor speaking a single word and showing exaggerated facial displays of a basic emotion. Thus, present systems for multimodal human affect analysis have all (and some additional) drawbacks of single-modal analyzers. Hence, many improvements are needed if those systems are to be used for context-sensitive analysis of human behavioral signals where a clean input from a known actor/ announcer cannot be expected and a context-independent processing and interpretation of audiovisual data do not suffice.

An additional important issue is that we cannot conclude that a system attaining a 92% average recognition rate performs “better” than a system achieving a 74% average recognition rate when detecting six basic emotions from audio and/or visual input stream unless both systems are tested on the same dataset. The main problem is that no audiovisual database exists that is shared by all diverse research communities in the field [Pantic and Rothkrantz, 2003]. Although efforts have been recently reported towards development of benchmark databases that can be shared by the entire research community [Pantic et al., 2005; Gunes and Piccardi, 2005], this remains an open, highly relevant issue.

Understanding Human Social Signaling: As we already remarked above, research findings in cognitive sciences tend to agree that at least some (if not the majority) of behavioral cues evolved to facilitate communication between people [Izard, 1997]. Types of messages conveyed by these behavioral cues include emblems, illustrators, and regulators, which can be further interpreted in terms of social signaling like turn taking, mirroring, empathy, antipathy, interest, engagement, agreement, disagreement, etc. Although each one of us understands the importance of social signaling in everyday life situations, and although a firm body of literature in cognitive sciences exists on the topic [Ambady and Rosenthal, 1992; Russell and Fernandez-Dols, 1997; Russell et al., 2003] and in spite of recent advances in sensing and analyzing behavioral cues like blinks, smiles, winks, thumbs up, yawns, laughter, etc. (see the state of the art in human sensing in the text above), the research efforts in machine analysis of human social signaling are few and tentative. An important part of the existing research on understanding human social signaling has been conducted at

MIT Media Lab, under the supervision of Alex Pentland [2005]. Their approach aims to discern social signals like activity level, stress, engagement, and mirroring by analyzing the engaged persons' tone of voice. Other important works in the field include efforts towards analysis of interest, agreement and disagreement from facial and head movements [El Kaliouby and Robinson, 2004] and towards analysis of the level of interest from tone of voice, head and hand movements [Gatica-Perez et al., 2005]. Overall, present approaches to understanding social signaling are multimodal and based on probabilistic reasoning methods like Dynamic Bayesian Networks. However, most of these methods are context insensitive (key context issues are either implicitly addressed, i.e., integrated in the inference process directly, or they are ignored altogether) and incapable of handling unconstrained environments correctly. Thus, although these methods represent promising attempts toward encoding of social variables like status, interest, determination, and cooperation, which may be an invaluable asset in the development of social networks formed of humans and computers (like in the case of virtual worlds), in their current form, they are not appropriate for general anticipatory interfaces.

4 Research Challenges

According to the taxonomy of human movement, activity, and behavioral action proposed by Bobick [1997], movements are low-level semantic primitives, requiring no contextual or temporal knowledge for the detection. Activities are sequences of states and movements, where the only knowledge required to recognize them relates to statistics of the temporal sequence. As can be seen from the overview of the past work done in the field, most of the work on human gesture recognition and human behavior understanding falls in this category. Human behavioral actions, or simply human behavior, are high-level semantic events, which typically include interactions with the environment and causal relationships. An important distinction between these different semantic levels of human behavior representation is the degree to which the context, different modalities, and time must be explicitly represented and manipulated, ranging from simple spatial reasoning to context-constrained reasoning about multimodal events shown in temporal intervals. However, most of the present approaches to machine analysis of human behavior are neither multimodal, nor context-sensitive, nor suitable for handling longer time scales. In our survey of the state of the field, we have tried to explicitly mention most of the existing exceptions from this rule in an attempt to motivate researchers in the field to treat the problem of context-constrained analysis of multimodal behavioral signals shown in temporal intervals as one complex problem rather than a number of detached problems in human sensing, context sensing, and human behavior understanding. Besides this critical issue, there are a number of scientific and technical challenges that we consider essential for advancing the state of the art in the field.

Scientific challenges in human behavior understanding can be summarized as follows.

- ♦ *Modalities:* How many and which behavioral channels like the face, the body, and the tone of the voice, should be combined for realization of robust and accurate human behavior analysis? Too much information from different channels seems to be confusing for human judges. Does this pertain in HCI?
- ♦ *Fusion:* At which abstraction level are these modalities to be fused? Humans simultaneously employ modalities of sight and sound. Does this tight coupling persists when the modalities are used for human behavior analysis, as suggested by some researchers, or not, as suggested by others? Does this depend on the machine learning techniques employed or not?
- ♦ *Fusion & Context:* While it has been shown that the *1+1>2* concept relevant to fusion of sensory neurons in humans pertain in machine context sensing [Nock et al., 2004], does the same hold for the other two concepts relevant to multimodal fusion in humans (i.e. context-dependent fusion and discordance handling)? Note that context-dependent fusion and discordance handling were never attempted.
- ♦ *Dynamics & Context:* Since the dynamics of shown behavioral cues play a crucial role in human behavior understanding, how the grammar (i.e., temporal evolvement) of human behavioral displays can be learned? Since the grammar of human behavior is context-dependent, should this be done in a user-centered manner [Oviatt, 2003] or in an activity/application-centered manner [Norman, 2005]?
- ♦ *Learning vs. Education:* What are the relevant parameters in shown human behavior that an anticipatory interface can use to support humans in their activities? How this should be (re-) learned for novel users and new contexts? Instead of building machine learning systems that will not solve any problem correctly unless they have been trained on similar problems, we should build systems that can be educated, that can improve their knowledge, skills, and plans through experience. Lazy and unsupervised learning can be promising for realizing this goal.

Technical challenges in human behavior understanding can be summarized as follows.

- ♦ *Initialization:* A large number of methods for human sensing, context sensing, and human behavior understanding require an initialization step. Since this is typically a slow, tedious, manual process, fully automated systems are the only acceptable solution when it comes to anticipatory interfaces of the future.
- ♦ *Robustness:* Most methods for human sensing, context sensing, and human behavior under-

standing work only in (often highly) constrained environments. Noise, fast movements, changes in illumination, etc., cause them to fail.

- *Speed:* Many of the methods in the field do not perform fast enough to support interactivity. Researchers usually choose for more sophisticated (but not always smarter) processing rather than for real time processing. A typical excuse is that according to Moore's Law we'll have faster hardware soon enough.
- *Training & Validation Issues:* United efforts of different research communities working in the field should be made to develop a comprehensive, readily accessible database of annotated, multimodal displays of human expressive behavior recorded under various environmental conditions, which could be used as a basis for benchmarks for efforts in the field. The related research questions include the following. How one can elicit spontaneous expressive behavior including genuine emotional responses and attitudinal states? How does one facilitate efficient, fast, and secure retrieval and inclusion of objects constituting this database? How could the performance of a tested automated system be included into the database? How should the relationship between the performance and the database objects used in the evaluation be defined?

5 Conclusions

Human behavior understanding is a complex and very difficult problem, which is still far from being solved in a way suitable for anticipatory interfaces and human computing application domain. In the past two decades, there has been significant progress in some parts of the field like face recognition and video surveillance (mostly driven by security applications), while in the other parts of the field like in non-basic affective states recognition and multimodal multi-aspect context-sensing at least the first tentative attempts have been proposed. Although the research in these different parts of the field is still detached, and although there remain significant scientific and technical issues to be addressed, we are optimistic about the future progress in the field. The main reason is that anticipatory interfaces and their applications are likely to become the single most widespread research topic of AI and HCI research communities. Even nowadays, there are a large and steadily growing number of research projects concerned with the interpretation of human behavior at a deeper level.

References

[Aarts, 2005] E. Aarts. Ambient intelligence drives open innovation. *ACM Interactions,* 12(4): 66-68, July-Aug. 2005.

[Ambady and Rosenthal, 1992] N. Ambady and R. Rosenthal. Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological Bulletin*, 111(2): 256-274, Feb. 1992.

[Ba and Odobez, 2004] S.O. Ba and J.M. Odobez. A probabilistic framework for joint head tracking and pose estimation. In *Proc. Conf. Pattern Recognition, vol. 4*, pp. 264-267, 2004.

[Bartlett et al., 2006] M.S. Bartlett, G. Littlewort, M.G. Frank, C. Lainscsek, I. Fasel and J. Movellan. Fully automatic facial action recognition in spontaneous behavior. In *Proc. Conf. Face & Gesture Recognition*, pp. 223-230, 2006.

[Bicego et al., 2006] M. Bicego, M. Cristani and V. Murino. Unsupervised scene analysis: A hidden Markov model approach. *Computer Vision & Image Understanding*, 102(1): 22-41, Apr. 2006.

[Bobick, 1997] A.F. Bobick. Movement, activity and action: The role of knowledge in the perception of motion. *Philosophical Trans. Roy. Soc. London B,* 352(1358): 1257-1265, Aug. 1997.

[Bowyer et al., 2006] K.W. Bowyer, K. Chang and P. Flynn. A survey of approaches and challenges in 3D and multimodal 3D+2D face recognition. *Computer Vision & Image Understanding,* 101(1): 1-15, Jan. 2006.

[Buxton, 2003] H. Buxton. Learning and understanding dynamic scene activity: a review. *Image & Vision Computing,* 21(1): 125-136, Jan. 2003.

[Cacioppo et al., 2000] J.T. Cacioppo, G.G. Berntson, J.T. Larsen, K.M. Poehlmann and T.A. Ito. The psychophysiology of emotion. In *Handbook of Emotions*. M. Lewis and J.M. Haviland-Jones, Eds. Guilford Press, New York, 2000, pp. 173-191.

[Chiang and Huang, 2005] C.C. Chiang and C.J. Huang. A robust method for detecting arbitrarily tilted human faces in color images. *Pattern Recognition Letters*, 26(16): 2518-2536, Dec. 2005.

[Costa et al., 2001] M. Costa, W. Dinsbach, A.S.R. Manstead and P.E.R. Bitti. Social presence, embarrassment, and nonverbal behavior. *Journal of Nonverbal Behavior,* 25(4): 225-240, Dec. 2001.

[Coulson, 2004] M. Coulson. Attributing emotion to static body postures: Recognition accuracy, confusions, & viewpoint dependence. *J. Nonverbal Behavior,* 28(2): 117-139, Jun. 2004.

[Deng and Huang, 2004] B.L. Deng and X. Huang. Challenges in adopting speech recognition. *Communications of the ACM,* 47(1): 69-75, Jan. 2004.

[Dey et al., 2001] A.K. Dey, G.D. Abowd and D.A. Salber. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *J. Human-Computer Interaction,* 16(2-4): 97-166, Dec. 2001.

[Duchowski, 2002] A.T. Duchowski. A breadth-first survey of eye-tracking applications. *Behavior Research Methods, Instruments and Computing,* 34(4): 455-470, Nov. 2002.

[Ekman and Friesen, 1969] P. Ekman and W.F. Friesen. The repertoire of nonverbal behavioral categories – origins, usage, and coding. *Semiotica*, 1: 49-98, 1969.

[Ekman et al., 2002] P. Ekman, W.V. Friesen and J.C. Hager. *Facial Action Coding System*. A Human Face, Salt Lake City, 2002.

[El Kaliouby and Robinson, 2004] R. El Kaliouby and P. Robinson. Real-Time Inference of Complex Mental States from Facial Expressions and Head Gestures. In *Proc. Int'l Conf. Computer Vision & Pattern Recognition, vol. 3*, p. 154, 2004.

[Fridlund, 1997] A.J. Fridlund. The new ethology of human facial expression. *The psychology of facial expression*. J.A. Russell and J.M. Fernandez-Dols, Eds. Cambridge University Press, Cambridge, UK, 1997, pp. 103-129.

[Furnas et al., 1987] G. Furnas, T. Landauer, L. Gomes and S. Dumais. The vocabulary problem in human-system communication, *Communications of the ACM,* 30(11): 964-972, Nov. 1987.

[Gatica-Perez et al., 2005] D. Gatica-Perez, I. McCowan, D. Zhang and S. Bengio. Detecting group interest level in meetings. In *Proc. Int'l Conf. Acoustics, Speech & Signal Processing, vol. 1*, pp. 489-492, 2005.

[Gibson and Ingold, 1993] K.R. Gibson and T. Ingold, Eds. *Tools, Language and Cognition in Human Evolution*. Cambridge University Press, Cambridge, UK, 1993.

[Gu and Ji, 2005] H. Gu and Q. Ji. Information extraction from image sequences of real-world facial expressions. *Machine Vision and Applications*, 16(2): 105-115, Feb. 2005.

[Gunes and Piccardi, 2005] H. Gunes and M. Piccardi. Affect Recognition from Face and Body: Early Fusion vs. Late Fusion, In *Proc. Int'l Conf. Systems, Man and Cybernetics*, pp. 3437- 3443, 2005.

[Haykin and de Freitas, 2004] S. Haykin and N. de Freitas, Eds. Special Issue on Sequential State Estimation. *Proceedings of the IEEE*, 92(3): 399-574, Mar. 2004.

[Huang and Trivedi, 2004] K.S. Huang and M.M. Trivedi. Robust real-time detection, tracking, and pose estimation of faces in video. In *Proc. Conf. Pattern Recognition, vol. 3*, pp. 965-968, 2004.

[Izard, 1997] C.E. Izard. Emotions and facial expressions: A perspective from Differential Emotions Theory. In *The psychology of facial expression*. J.A. Russell and J.M. Fernandez-Dols, Eds. Cambridge University Press, Cambridge, UK, 1997, pp. 103-129.

[Jain and Ross, 2004] A.K. Jain and A. Ross. Multibiometric systems. *Communications of the ACM,* 47(1): 34-40, Jan. 2004.

[Juslin and Scherer, 2005] P.N. Juslin and K.R. Scherer. Vocal expression of affect. In *The New Handbook of Methods in Nonverbal Behavior Research*. J. Harrigan, R. Rosenthal and K.R. Scherer, Eds. Oxford University Press, Oxford, UK, 2005.

[Keltner and Ekman, 2000] D. Keltner and P. Ekman. Facial expression of emotion. In *Handbook of Emotions*, M. Lewis and J.M. Haviland-Jones, Eds. The Guilford Press, New York, 2000, pp. 236-249.

[Li and Jain, 2005] S.Z. Li and A.K. Jain, Eds. *Handbook of Face Recognition*. Springer, New York, 2005.

[Lisetti and Schiano, 2000] C.L. Lisetti and D.J. Schiano. Automatic facial expression interpretation: Where human-computer interaction, AI and cognitive science intersect. *Pragmatics and Cognition*, 8(1): 185-235, Jan. 2000.

[Maat and Pantic, 2006] L. Maat and M. Pantic. Gaze-X: Adaptive affective multimodal interface for single-user office scenarios. In *Proc. Int'l Conf. Multimodal Interfaces*, pp. 171-178, 2006.

[Matos et al., 2006] S. Matos, S.S. Birring, I.D. Pavord and D.H. Evans. Detection of cough signals in continuous audio recordings using HMM. *IEEE Trans. Biomedical Engineering,* 53(6): 1078-1083, June 2006.

[Nijholt et al., 2004] A. Nijholt, T. Rist and K. Tuinenbreijer. Lost in ambient intelligence. In *Proc. Int'l Conf. Computer Human Interaction*, pp. 1725-1726, 2004.

[Nijholt et al., 2006] A. Nijholt, B. de Ruyter, B., D. Heylen, and S. Privender. Social Interfaces for Ambient Intelligence Environments. Chapter 14 in: *True Visions: The Emergence of Ambient Intelligence*. E. Aarts and J. Encarnaçao, Eds. Springer, New York, 2006, pp. 275-289.

[Nijholt and Traum, 2005] A. Nijholt and D. Traum. The Virtuality Continuum Revisited. In *Proc. Int'l Conf. Computer Human Interaction*, pp. 2132-2133, 2005.

[Nock et al., 2004] H.J. Nock, G. Iyengar and C. Neti. Multimodal processing by finding common cause. *Communications of the ACM,* 47(1): 51-56, Jan. 2004.

[Norman, 2005] D.A. Norman. Human-centered design considered harmful, *ACM Interactions,* 12(4): 14-19, July-Aug. 2005.

[Oudeyer, 2003] P.Y. Oudeyer. The production and recognition of emotions in speech: features and algorithms. *Int'l J. Human-Computer Studies,* 59(1-2): 157-183, July 2003.

[Oviatt, 2003] S. Oviatt. User-centered modeling and evaluation of multimodal interfaces. *Proceedings of the IEEE,* 91(9): 1457-1468, Sep. 2003.

[Pal et al., 2006] P. Pal, A.N. Iyer and R.E. Yantorno. Emotion detection from infant facial expressions and cries. In *Proc. Int'l Conf. Acoustics, Speech & Signal Processing, 2*, pp. 721-724, 2006.

[Pantic, 2006] M. Pantic. Face for Ambient Interface. *Lecture Notes in Artificial Intelligence*, 3864: 35-66, 2006.

[Pantic and Patras, 2006] M. Pantic and I. Patras. Dynamics of Facial Expressions – Recognition of Facial Actions and their Temporal Segments from Face Profile Image

Sequences. *IEEE Trans. Systems, Man, and Cybernetics, Part B*, 36(2): 433-449, Apr. 2006.

[Pantic and Rothkrantz, 2003] M. Pantic and L.J.M. Rothkrantz. Toward an Affect-Sensitive Multimodal Human-Computer Interaction. *Proceedings of the IEEE*, 91(9): 1370-1390, Sep. 2003.

[Pantic et al., 2005] M. Pantic, M.F. Valstar, R. Rademaker and L. Maat. Web-based database for facial expression analysis. In *Proc. Int'l Conf. Multimedia and Expo*, pp. 317-321, 2005. (www.mmifacedb.com)

[Pantic et al., 2006] M. Pantic, A. Pentland, A. Nijholt and T. Huang. Human Computing and Machine Understanding of Human Behavior: A Survey. In *Proc. Int'l Conf. Multimodal Interfaces*, pp. 239-248, 2006.

[Pentland, 2005] A. Pentland. Socially aware computation and communication. *IEEE Computer*, 38(3): 33-40, Mar. 2005.

[Russell and Fernandez-Dols, 1997] J.A. Russell and J.M. Fernandez-Dols, Eds. *The psychology of facial expression*. Cambridge University Press, Cambridge, UK, 1997.

[Russell et al., 2003] J.A. Russell, J.A. Bachorowski and J.M. Fernandez-Dols. Facial and Vocal Expressions of Emotion. *Annual Review of Psychology*, 54: 329-349, 2003.

[Ruttkay et al., 2006] Z.M. Ruttkay, D. Reidsma and A. Nijholt. Human computing, virtual humans and artificial imperfection. In *Proc. Int'l Conf. Multimodal Interfaces*, pp. 179-184, 2006.

[Sand and Teller, 2006] P. Sand and S. Teller. Particle Video: Long-Range Motion Estimation using Point Trajectories. In *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pp. 2195-2202, 2006.

[Scanlon and Reilly, 2001] P. Scanlon and R.B. Reilly. Feature analysis for automatic speech reading. In *Proc. Int'l Workshop Multimedia Signal Processing*, pp. 625-630, 2001.

[Sharma et al., 2003] R. Sharma, M. Yeasin, N. Krahnstoever, I. Rauschert, G. Cai, A.M. Maceachren and K. Sengupta. Speech-gesture driven multimodal interfaces for crisis management. *Proceedings of the IEEE*, 91(9): 1327-1354, Sep. 2003.

[Song et al., 2004] M. Song, J. Bu, C. Chen and N. Li. Audio-visual based emotion recognition – A new approach. In *Proc. Int'l Conf. Computer Vision and Pattern Recognition*, pp. 1020-1025, 2004.

[Starner, 2001] T. Starner. The Challenges of Wearable Computing. *IEEE Micro*, 21(4): 44-67, July-Aug. 2001.

[Stein and Meredith, 1993] B. Stein and M.A. Meredith. *The Merging of Senses*. MIT Press, Cambridge, USA, 1993.

[Stenger et al., 2006] B. Stenger, P.H.S. Torr and R. Cipolla. Model-based hand tracking using a hierarchical Bayesian filter. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 28(9): 1372-1384, Sep. 2006.

[Streitz and Nixon, 2005] N. Streitz and P. Nixon. The Disappearing Computer. *ACM Communications*, 48(3): 33-35, Mar. 2005.

[Truong and van Leeuwen, 2005] K.P. Truong and D.A. van Leeuwen. Automatic detection of laughter. In Proc. *Interspeech Euro. Conf.*, pp. 485-488, 2005.

[Valstar and Pantic, 2006a] M.F. Valstar and M. Pantic. Biologically vs. logic inspired encoding of facial actions and emotions in video. In *Proc. Int'l Conf. on Multimedia and Expo*, 2006.

[Valstar and Pantic, 2006b] M.F. Valstar and M. Pantic. Fully automatic facial action unit detection and temporal analysis. In *Proc. Int'l Conf. Computer Vision and Pattern Recognition, vol. 3*, p. 149, 2006.

[Valstar et al., 2006] M.F. Valstar, M. Pantic, Z. Ambdar and J.F. Cohn. Spontaneous vs. Posed Facial Behavior: Automatic Analysis of Brow Actions. In *Proc. Int'l Conf. Multimodal Interfaces*, pp. 162-170, 2006.

[Viola and Jones, 2004] P. Viola and M.J. Jones. Robust real-time face detection. *Int'l J. Computer Vision*, 57(2): 137-154, May 2004.

[Wang and Singh, 2003] J.J. Wang and S. Singh. Video analysis of human dynamics – a survey. *Real Time Imaging*, 9(5): 321-346, Oct. 2003.

[Wang et al., 2003] L. Wang, W. Hu and T. Tan. Recent developments in human motion analysis. *Pattern Recognition*, 36(3): 585-601, Mar. 2003.

[Weiser, 1991] M. Weiser. The Computer for the Twenty-First Century. *Scientific American*, 265(3): 94-104, Sep. 1991.

[Yang et al., 2002] M.H. Yang, D.J. Kriegman and N. Ahuja. Detecting faces in images: A survey. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(1): 34-58, Jan. 2002.

[Zhai and Bellotti, 2005] S. Zhai and V. Bellotti. Sensing-Based Interaction. *ACM Trans. Computer-Human Interaction*, 12(1): 1-2, Jan. 2005.

[Zhao et al., 2003] W. Zhao, R. Chellappa, P.J. Phillips and A. Rosenfeld. Face recognition: A literature survey. *ACM Computing Surveys*, 35(4): 399-458, Dec. 2003.

[Zeng et al., 2006] Z. Zeng, Y. Hu, G.I. Roisman, Y. Fu and T.S. Huang. Audio-visual Emotion Recognition in Adult Attachment Interview. In *Proc. Int'l Conf. Multimodal Interfaces*, pp. 139-145, 2006.

[BTT Survey, 2006] BTT Survey on Alternative Biometrics. *Biometric Technology Today*, 14(3): 9-11, Mar. 2006.

[MMUA, 2006] MMUA: http://mmua.cs.ucsb.edu/

Towards Embeddable Vision Architectures for Human Computing

Marten den Uyl
CEO VicarVision and president SMRgroep
Amsterdam, The Netherlands
denuyl@vicarvision.nl

Abstract

Human Computing is about perceptive, anticipatory interfaces that support natural and intuitive human-computer interaction by understanding human behavior, emotions and social signaling. Yet, machine understanding of human behavior and emotion is limited and fragmented until this day. It is proposed that in order to manage the complexities of multimodal, multilevel and contextual machine perception of human behavior, an embedded systems approach towards the design of vision architectures for human computing seems advisable. The problems encountered in such an approach are illustrated from past and present development projects on vision systems for watching humans.

1 Introduction

Sentient Machine Research was founded in 1990 as an R&D company in AI with the aim to contribute to developing machines into sentient interaction partners. What makes a machine 'sentient' from the third person point of view, in the eye of a human observing the system, is first of all whether the system seems to understand us, seems 'human aware'. Many fascinating AI systems have been developed since e.g. Weizenbaum's Eliza back in 1966, that create an illusion of user awareness. But in fact, and in spite of much progress in recent years, machine understanding of human behavior and emotion is limited and fragmented until this day [Pantic et al 2006]. Machines struggle with perceiving the utterances, grimaces and gestures and often fail to understand the intentions and feelings of the humans they interact with. Whether or when any such machine should moreover be considered sentient in the first person view, i.e. whether 'it is something to be that machine', is still a challenging philosophical question, that in the end may well be decided by empirical means. Man and other animals are the only truly sentient machines we have seen so far, thus the alternative aim of Sentient Machine Research is to achieve a better understanding of human nature and experience by synthetic research methods, that is by building AI artifacts. VicarVision, a subsidiary of the SMRgroep, was founded in 2001 with the mission to develop computer vision systems for perceiving humans in video streams. The long term aim is to develop general purpose vision systems that allow robots to classify and label objects and events in mundane environments in human understandable terms.

A problem for Human Computing, the automatic sensing and understanding of human behavior in an *ambient intelligent* environment, is that on the one hand computer vision system architectures tend to expand over time in complexity, with more components, connections, subroutines and dependencies and an ever increasing appetite for faster computers. Yet, on the other hand, there are good reasons to try to make vision systems embeddable, efficient and small. First of all, it would of course be nice if competent vision systems could be run on hardware sufficiently small, cheap and energy efficient to be embedded in affordable robots or mobile devices. Even if price, size and energy consumption are not major concerns in the design of a vision system, it might be worthwhile to strive for a vision architecture that is embeddable in principle. The big challenge is that vision, particularly when watching humans, really is a very complex computational problem, requiring many specialist processes, extensive knowledge and massive raw computational power to perform real world tasks in real time. Thus, the basic requirements for embeddable systems -cheap, small and energy efficient- are directly challenged by the large and highly variable computational loads that come with vision tasks, even with the smartest possible algorithms. Further requirements for embedded systems follow from their embeddedness –by definition- within a host system with more functions and components to care about. Embedded systems generally need to be dependable, predictable and collaborative. Dependable because the performance of the system as a whole may critically depend on the proper functioning of the embedded system, there might not be a fall back when it fails. Predictability and collaborativeness of embedded systems are particularly important in multitasking systems with concurrent processing. Embedded systems that are part of such architectures are likely to share, and thus compete for, inherently limited resources such as access to communication channels, or instruction sets for action, or central memory and cpu cycles. Predictability of performance allows for control and in collaborative multitasking each process performing a task is optimally transparent –for predictability- and interruptible –for co-cooperativeness.

Embedded systems preferably are self-supporting with minimal need for external maintenance, support or upgrades. It is at least inconvenient if the host system must be serviced or repaired because some embedded system shows some malfunctioning. Obviously, being dependable, predictable, collaborative and self-supporting are often desirable features for any information system that humans interact with, even if there is no need yet to realize the system by embedded systems technology.

2 Does Human Computing really need embeddable system architectures?

Ambient Intelligence is about disembodiment, the computer has disappeared out of sight, the actual processing occurs somewhere 'in the back end'. Embeddableness of perceptual processing may then seem just an engineering issue for the efficient technical realization of Human Computing systems. However, some defining aspects of Human Computing imply that embeddability is highly relevant for theoretical issues. Human Computing is multimodal, multilevel and contextual and it must proceed in real time. Various sensory modalities may contribute to understanding behavior besides vision; audio, tactile even olfactory modalities can be used to sense a human. Actually, Human Computing may also use senses underdeveloped in humans such as perception of electromagnetic fields, infrared radiation, ultrasound, etc. Within a single modality such as vision a number of analysis channels may be distinguished, as for example, one may choose to watch the face, or the body, or the eyes, or the gesticulating hands of a person. And sometimes multiple instantiations of analytic processes are required, e.g. when observing two people interacting. Thus, Human Computing requires concurrent processing architectures, and in such architectures some compromise must be made between dedicated and shared resources per modality -or rather per processing channel. If all perceptual processes have fully dedicated resources, this gives maximal robustness against interference by other processes, at the cost of extreme redundancy of computational resources that will remain idle much of the time, while the processes they support are not triggered by current inputs.

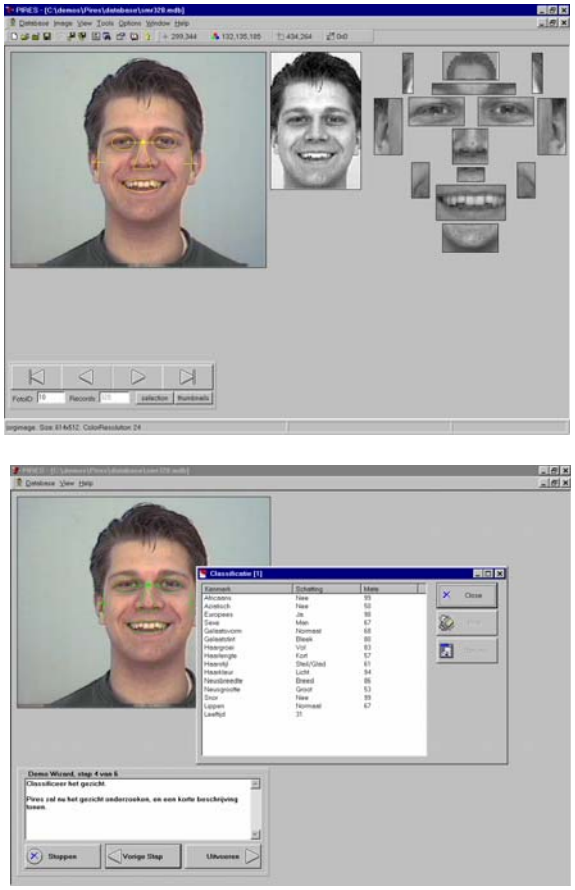
Human Computing is multi-level in the sense that the full path must be covered from registration by sensors, pixels from cameras or soundwaves from microphones, through various stages of processing, until arriving at some understanding of the intentions, actions and experiences of the humans observed. Typically, the lower levels have dedicated resources while the highest levels share resources. The balance that a given architecture strikes between dedicated and shared resources is related to the classical issue of early versus late fusion in multimodal sensory integration. Perhaps somewhat counter intuitively, it is much more difficult to perform early fusion in an architecture where low level processes must share resources. [Nock et al 2004] provide an interesting analysis of a classical Gestalt principle of perception, the detection of common cause by temporal contiguity, on the lowest possible level of multimodal analysis, temporal correlation between pixel and sound wave intensities. In shared resource models, computation of these correlations is very costly, because it requires explicit computation and memory buffering of large temporal index structures. In a dedicated resource model where image and sound are processed in parallel by transparently embedded systems, the correlation patterns for detecting common cause can be obtained at little extra cost by temporal correlation over small sets of system processing load parameters. The aim of Human Computing is high level analysis, a proper response requires an understanding of the meaning of the human behavior observed. A framework for high level analysis of action, expression and experience can be found in emotion theory [Frijda 1986], where it is proposed that event appraisal results from the matching of situation aspects to active concerns. From such a formulation it directly follows that some knowledge of context is required to be able to infer what concerns may be active in the person observed. Is this person showing a sad face because she has just received unpleasant news, or because she was asked to pose a sad face?

Context in fact is all important in Human Computing. Not just because at the highest level affect and behavior can only be understood to the extent one knows, i.e. has at least some general or default model of 'where a person comes from' and what moves the person in the situation. The extensive psychological literature on priming, the role of expectation and context effects in perception indicates that for humans, perception is context dependent on all levels. Perception is not a one-way processing stream from input to meaning but rather a cyclical process steered by top down anticipation as well as the bottom up input data stream [Neisser1976]. First time students of neuroanatomy are often surprised to learn that the neural circuitry for the vision system does not consist of just an upwards series of processing stages or projection fields, but that almost as much circuitry is dedicated to the downward modulation of these processing stages. The basic reason is that only through anticipation the complexities in real time visual –or auditory- processing can be mastered. Just as efficient tracking of a moving object requires some form of prediction in x,y coordinates, so does understanding of say the current face expression of the subject requires some form of prediction of direction in affective coordinates –dimensions or categories. And sensitive evaluation of the expression on a particular kind of face –young or old, Asian or European- requires a momentary specialization, an anticipation based on similar expressions seen on similar faces before. It has proven difficult to produce computational vision systems that come anywhere near the ability of natural vision for anticipatory and adaptive perceptual processing. The basic hunch is that one should first be able to realize vision components as predictable and self-supporting embedded systems, before the added complexities that come with adaptivity and context dependency can be successfully dealt with.

The conflicts that arise when vision architectures increase in complexity and size, while embedded systems require a small footprint and strong encapsulation, will be illustrated by a short description of some vision systems that have been developed at SMR and VicarVision over the years.

3 Pires (1997)

PIRES (PIcture REtrieval System) is a forensic face recognition system that accepts pictures, ‘mug shots’ and constructs an extensive face index representation for each individual portrait. This index structure allows for search by image in a person database. When presented with a portrait as query, similar portraits will be found. PIRES also produces a detailed description of the characteristics of the face and is able to fill most of the Dutch police standard person description, *signalement,* reporting form with about 40 facial feature categories –from big/small nose to hairstyles and ethnic origin.


PIRES - [C:\demos\Pires\database\smr328.mdb]
Database Image View Tools Options Window Help
FotoID 10
Records
selection
thumbnails
orgimage Size 614x512 ColorResolution 24
Database View Help
Classificatie [1]
Close

Kenmerk	Schatting	Mate
Africaans	Nee	93
Aziatisch	Nee	53
Europees	Ja	98
Sexe	Man	67
Gelaatsvorm	Normaal	68
Gelaatstint	Bleek	80
Haargroei	Vol	83
Haarlengte	Kort	57
Haarstijl	Steil/Glad	61
Haarkleur	Licht	94
Neusbreedte	Breed	86
Neusgrootte	Groot	53
Snor	Nee	99
Lippen	Normaal	67
Leeftijd	31	

Demo Wizard, step 4 van 6

Classificeer het gezicht.

Pires zal nu het gezicht onderzoeken, en een korte beschrijving tonen.

Figure 1: PIRES face analysis

The PIRES system performs the face analysis task in a three step perceptual processing architecture: *Find, Frame, Featurize.*

1) Face finding, the detection and localization of the face in the image is performed by standard template search. A template is constructed by averaging the grey pixel values of a set of aligned faces. One or more templates are evaluated in discrete steps over the image at a number of scales and the best match position is selected.
2) A visual object is ‘framed’ by establishing a correspondence between localised features of the image and an object topology model. PIRES employs a small set of local feature templates to find the position of eyes, nose and mouth in the portrait. From these positions the full face topology can be estimated.
3) The features of the framed face are derived by a set of neural network classifiers trained on various sets of annotated ‘clips’ taken from the face.

For each portrait a long representation vector is constructed, consisting of neural network hidden node activation values, explicit classifier labels and face topology coordinates. An associative search engine is used to find best matches for a newly analyzed portrait in a list of previously indexed portraits. PIRES performs quite reasonable as a forensic face analysis system under a limited range of conditions. The system can only handle high quality frontal images, but achieves respectable recognition rates / retrieval within best n matches, on police image databases. PIRES in the 1997 implementation may not be a good candidate for an embedded system because the template based face framing routine is limited to frontal faces and even with frontal faces sometimes framing errors are made. However, there are no principled reasons why PIRES could not be implemented as an embedded system, say for automatic indexing of portraits in a high end digital camera, if performance is improved within the same architecture. The three basic processing steps are performed by modules that each can be encapsulated and used in a predictable sequence of operations.

4 Vicar (2001)

The aim of the VICAR Video Explorer system -developed in the VICAR (Video Indexing, Classification, Annotation and Retrieval) HPCN/IST EU project- is to provide for semantic indexing and content based search for large amounts of video footage. The field of operation is the professional video archive market (broadcasting, movie productions and agencies, security). Types of contents indexed by VICAR include:

- shot detection and camera motion;
- moving object detection and segmentation;

- setting classification, evaluate the general setting and background of images [Israël et al 2004];
- object recognition; VICAR contains object recognizers for faces, cars and horses [Noorman et al 2002];
- recognition of individuals, VIPFinder recognizes faces from a short list of famous persons;
- classification of behaviors, e.g. walking, running, limping.

While VICAR demonstrated the feasibility in principle of indexing contents at many levels, performance at some levels was not yet up to operational use in 2001. For present purposes it is of interest to look at a sketch of -part of- the VICAR system architecture.

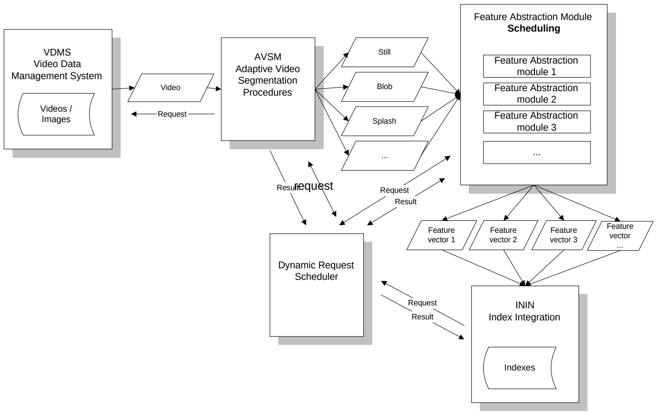
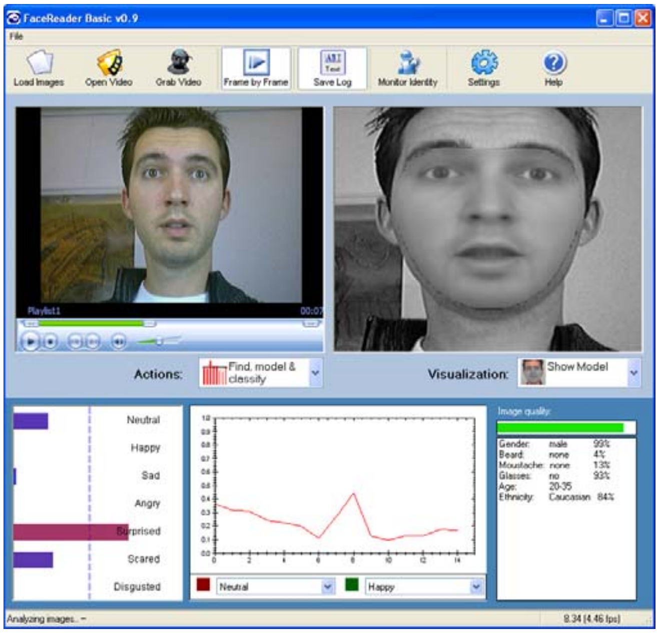


Figure 2: VICAR architecture, flow of control.

Even a superficial inspection of figure 2 suggests that the VICAR architecture looks like a worst case for an embeddable system, with many modules and datatypes and the control of processing going in all directions. Actually, VICAR was not designed to run as an embedded system, but on almost the opposite kind of platform, a multi cpu supercomputer. To review some of the problems, first again comes the need for computational power, e.g. some of the object recognition procedures at that time required many seconds of cpu time for processing a single image. Next, specific perceptual processes require specific subsets of the x,y,t pixel volume that makes up a videostream, e.g. when analyzing behavior of a walking person, the person must be tracked and the relevant pixels must be segmented from the stream. This requires either complex dynamic memory management, or huge amounts of random access memory. Both options are not popular with embedded system engineers. Then, since vision is data-driven, a new scene may spawn many perceptual processes, that will compete for available resources, while many interdependencies may exist between different levels of content analysis. Together, these complications make the processing that occurs within the architecture rather unpredictable. And since time-constrained operation –even if not real time- makes it unsure whether the relevant processes will all have managed to run to completion, the results of perceptual analysis may not be very dependable.

5 FaceReader (2005)

FaceReader is a commercially available product for real time analysis of facial expression [Den Uyl and Van Kuilenburg 2005]. FaceReader fits a face in a video stream with a mask computed by an active appearance model [Cootes and Taylor 2000] and derives persistent –gender, age, ethnicity- and changing features, particularly the emotional expression of the face. Expressions are classified in 7 emotion categories, 6 basic emotions -happy, sad, angry, surprised, scared and disgusted- and neutral.



Actions: Find, model & classify

Visualization: Show Model

🟥 Neutral 🟩 Happy

Gender:	male	99%
Beard:	none	4%
Moustache:	none	13%
Glasses:	no	93%
Age:	20-35	
Ethnicity:	Caucasian	84%

Analyzing images... ~

8.34 (4.46 fps)

Figure 3: FaceReader interface

FaceReader employs the same three step –find, frame, featurize- architecture as PIRES, though some of the steps have changed considerably. Where PIRES uses static templates for finding faces, FaceReader uses one or more ‘Active Templates’ [Song and Poggio 1998] for finding a face in the image. The Active Template Method moves a set of deformable face templates over an image, returning the most likely face position. To frame the face, FaceReader uses an Active Appearance Model [Cootes and Taylor 2000], able to produce good fits over a wide range of variation in persons and lighting, orientation and expression. Face features are derived by neural networks, trained on the appearance vector –the list of about 100 appearance parameters found for the best fit mask [Van Kuilenburg et al 2005].

The FaceReader architecture (Figure 4) shows a principled distinction between the online or execution model and the offline training environment. This contributes to the embeddableness of the online system, since different configurations, optimized for different tasks, can be obtained by replacing modules. FaceReader is an online system, expecting a face in a video stream, whereas PIRES is an event-driven system, triggered by the presentation of a still image. This is reflected by the inbound arrows in the FaceReader architecture. For face finding, once a face is found, a tracking subroutine will start tracking the face for further speed optimization, currently the FaceReader can do the full find, fit and classify cycle at around 20 frames per second on a quick PC. The inbound arrow on the classification box indicates temporal integration; identity, current expression and other feature estimates are based –by default- on temporal integration over a series of images. Although the active appearance modeling approach is particularly computationally intensive, even after extensive optimization, FaceReader could be implemented as an embedded system about as well as PIRES.

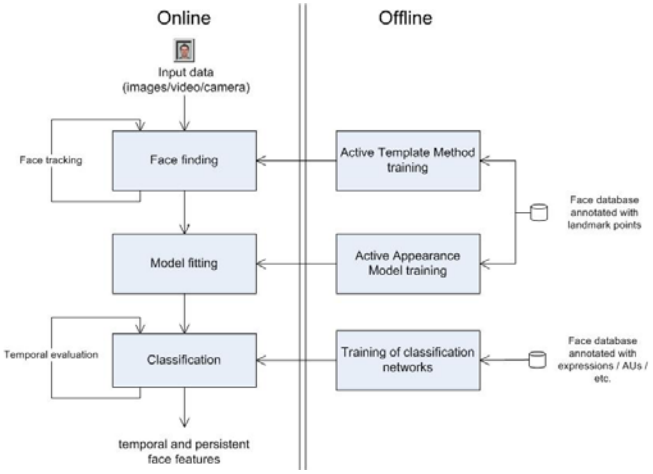
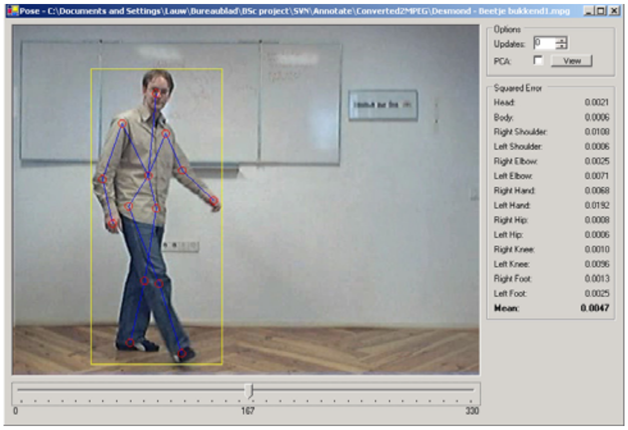

Online
Offline
Input data
(images/video/camera)
Face tracking
Face finding
Active Template Method training
Face database annotated with landmark points
Model fitting
Active Appearance Model training
Temporal evaluation
Classification
Training of classification networks
Face database annotated with expressions / AUs / etc.
temporal and persistent face features

Figure 4: FaceReader architecture.

5 BodyReader (2006)

A system currently under development at VicarVision is the BodyReader, which aims to give an estimate of body pose –i.e. the localization of body, head and limbs- of a moving person in real time [Van der Meer and Metz 2006]. The global architecture is again a three step -find, frame, featurize- perceptual process, with a strict separation between online and offline facilities, just as for FaceReader. Almost all the analysis processes are however entirely different. For finding the moving body, a standard motion differential method puts a box around a suitably sized moving blob in the image. Framing the body within the box is a three step process in itself. First a neural network makes a rough guess at the location of 14 anchor points on the body (see figure 5). Then a PCA model trained on a body topology reference database is used to move the anchor points to more plausible positions. Lastly, for each anchor point placement a local refinement is attempted by an active search method. Static pose features –arms up or down- can be derived trivially from the framing model, the more interesting class is that of dynamic pose features, that is, body movements.

Note that the BodyReader can only track the movements of one person at a time, but it might well do that as an embedded system.


Pose - C:\Documents and Settings\Lauw\Bureaublad\BSc project\SVN\Annotate\Converted2MPEG\Desmond - Beetje bukkend1.mpg

Options
Updates: 0
PCA: □ View

Squared Error

Head:	0.0021
Body:	0.0006
Right Shoulder:	0.0100
Left Shoulder:	0.0006
Right Elbow:	0.0025
Left Elbow:	0.0071
Right Hand:	0.0068
Left Hand:	0.0192
Right Hip:	0.0008
Left Hip:	0.0006
Right Knee:	0.0010
Left Knee:	0.0096
Right Foot:	0.0013
Left Foot:	0.0025
Mean:	**0.0047**

Figure 5: BodyReader: automatic annotation of 14 body anchor points.

6 AIVOS (2006)

AIVOS –Architecture for Intelligent Video Observation Systems- is a project-under-development at VicarVision aimed at developing vision systems that can fulfill a range of security and surveillance tasks. This is about as broad a task as the VICAR Video Explorer task of indexing any

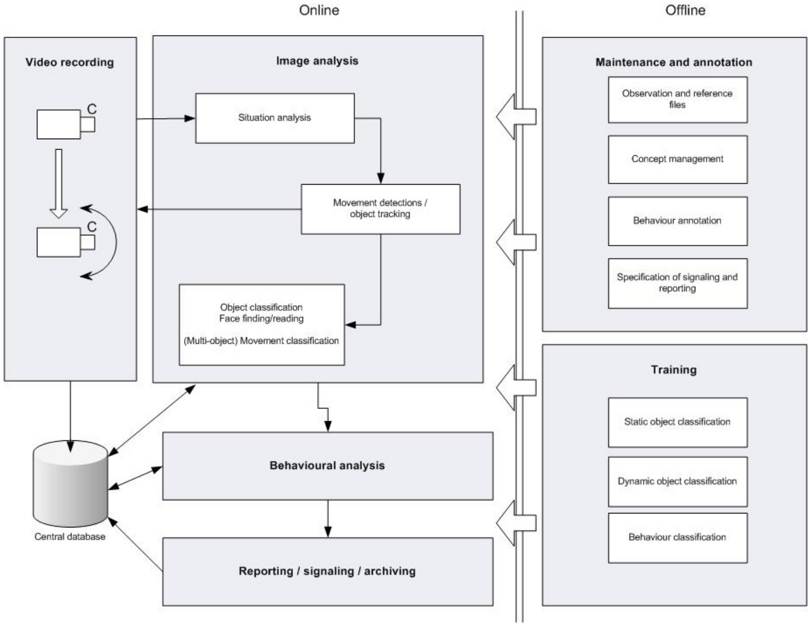

Online
Offline
Video recording
C
C
Image analysis
Situation analysis
Movement detections / object tracking
Object classification
Face finding/reading
(Multi-object) Movement classification
Central database
Behavioural analysis
Reporting / signaling / archiving
Maintenance and annotation
Observation and reference files
Concept management
Behaviour annotation
Specification of signaling and reporting
Training
Static object classification
Dynamic object classification
Behaviour classification

Figure 6: AIVOS main components

kind of video footage for archive. Even though most surveillance tapes show little of interest happening at all, pretty much anything worth reporting could happen at any moment. In fact it is mostly mundane human behavior that is to be observed, signaled and reported. Thus the main 'find' routine in AIVOS is motion based object detection, as in the BodyReader, except that surveillance requires that multiple moving objects can be detected and tracked, possibly with an active camera. Framing the body of moving persons is performed by the BodyReader, just as their faces are modeled and classified by FaceReader, when they are detected to be within scope of the face fitting appearance models.

AIVOS then is the framework for setting up virtual vision engines: object controlled perceptual analysis channels, that direct what pixels they want from the input and apply perceptual processes to this pixel stream, relevant to the object in the image and the purposes of the observation system. When the visual input is rich, a number of people moving around, managing a set of virtual vision systems that share physical resources becomes an intricate problem that easily leads to overload even for high capacity systems. It is not necessarily the case that a system is embeddable, even if all of its components are embeddable. But an AIVOS type system might well be made embeddable, by strictly limiting its span of visual attention, the number of virtual vision engines it can run concurrently.

7 Conclusion

Human Computing is all about managing the complexities of multimodal and multilevel perceptual processing, adaptive and context sensitive and in real time. Aiming for systems that are embeddable in principle is a basic 'divide and conquer' strategy. Only out of well-behaved, transparent and self-supporting components can we hope to be able to build such complex systems. A straightforward conclusion appears to be that a vision architecture is embeddable if it performs a single chain of tasks on a single class of visual objects and if its basic resources –cpu cycles and memory or knowledge and data access- can be managed 'on board'. This tends to limit candidate vision architectures for embedding to 'one trick ponies', which in turn tend to have limited use as an embedded vision system for general purpose hosts like robots or mobile devices; their users would like to see them do many tricks. A way out of this dilemma might be to develop vision architectures that support multiple virtual vision engines. This seems not so much a matter of developing new vision algorithms, rather of better compositions with existing algorithms, embedded or not .

It has been mentioned repeatedly that embedded systems should preferably be self-supporting. That implies in fact, in the case of vision, that systems should be able to teach themselves to see new things. This seems the aspect where current vision architectures are still the furthest away from target. It may be a long time before one can install the off-line training environment for a vision system on board and feel reasonably confident that the host will know what to do with it.

References

[Cootes and Taylor 2000] Tim J. Cootes,. and C. J. Taylor, (2000). Statistical models of appearance for computer vision. Technical report, University of Manchester.

[Frijda 1986] Nico H. Frijda, (1986) "The Emotions". Cambridge University Press: Studies in Emotion and Social Interaction series.

[Israël et al 2004] Menno Israël, Egon L. van den Broek, Peter van der Putten, and Marten J. den Uyl, (2004). Automating the construction of scene classifiers for content-based video retrieval. In L. Khan and V.A. Petrushin (Eds.), *Proceeding of the Fifth International Workshop on Multimedia Data Mining (MDM/KDD'04)*, p. 38-47. August 22, Seattle, WA - USA.

[Van Kuilenburg et al 2005] Hans van Kuilenburg, Marco Wiering and Marten J. den Uyl. A model based method for automatic facial expression recognition. Machine Learning: ECML 2005: *16th European Conference on Machine Learning*, Porto, Portugal, October 3-7, 2005. Proceedings pp. 194 - 205 Springer-Verlag GmbH.

[Van der Meer and Metz 2006] Desmond van der Meer and Lauwerens Metz (2006). AIVOS and Human Pose Estimation, internal report VicarVision/Delft University of Technology.

[Nock et al 2004] Harriet J. Nock, Giridharan Iyengar, Chalapathy Neti. Multimodal Processing by Finding Common Cause. Communications of the ACM, January 2004, Vol 47, 51-56.

[Neisser 1976] Ulric Neisser. Cognition and Reality. (1976) W.H. Freeman and Company, San Francisco.

[Noorman et al 2002] Merel Noorman, Kai Otto, Marten J. den Uyl, Rein van den Boomgaard. Horse Recognition: A General Approach to Object Recognition. Proceedings of the 12th Portuguese Conference on Pattern Recognition, 2002

[Pantic et al 2006] Maja Pantic, Alex Pentland, Anton Nijholt, Thomas Huang. Human Computing and Machine Understanding of Human Behavior: A Survey. International Conference on Multimodal Interfaces 2006.

[Sung and Poggio 1998] K.K. Sung and Tomaso Poggio. Example-based learning for view-based human face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1):39–51, 1998.

[Den Uyl and Van Kuilenburg 2005] Marten J. den Uyl, Hans van Kuilenburg, (2005). The FaceReader: Online facial expression recognition. Proceedings of Measuring Behaviour 2005, 5th *International Conference on Methods and Techniques in Behavioural Research*, 589-590.

Trajectory-based Representation of Human Actions

A. Oikonomopoulos[1], I.Patras[2], M.Pantic[3], N.Paragios[4]
[1,3]Imperial College London, 180 Queensgate, SW7 2AZ London, UK
[2]University of York, Heslington, York, YO 10 5DD, UK
[4] Ecole Centrale de Paris, Grande Voie des Vignes, 92 295 Chatenay-Malabry, FRANCE
aoikonom@doc.ic.ac.uk, I.Patras@cs.york.ac.uk, m.pantic@imperial.ac.uk, nikos.paragios@ecp.fr

Abstract

This work addresses the problem of human action recognition by introducing a representation of a human action as a collection of short trajectories that are extracted in areas of the scene with significant amount of visual activity. The trajectories are extracted by an auxiliary particle filtering tracking scheme that is initialized at points that are considered salient both in space and time. The spatiotemporal salient points are detected by measuring the variations in the information content of pixel neighborhoods in space and time. We implement an online background estimation algorithm in order to deal with inadequate localization of the salient points on the moving parts in the scene, and to improve the overall performance of the particle filter tracking scheme. We use a variant of the Longest Common Subsequence algorithm (LCSS) in order to compare different sets of trajectories corresponding to different actions. We use Relevance Vector Machines (RVM) in order to address the classification problem. We propose new kernels for use by the RVM, which are specifically tailored to the proposed representation of short trajectories. The basis of these kernels is the modified LCSS distance of the previous step. We present results on real image sequences from a small database depicting people performing 12 aerobic exercises.

1 Introduction

The key to ambient intelligence, anticipatory interfaces, and human computing is the ease of use - the ability to unobtrusively sense certain behavioral cues of the users and to adapt automatically to their typical behavioral patterns and the context in which they act [Pantic *et al.*, 2006]. This paper concerns sensing and interpretation of human behavioral cues expressed by means of body actions. Because of its practical importance and relevance for the security (video surveillance and monitoring) as well as for natural multimodal interfaces, vision-based analysis of hand and body gestures is nowadays one of the most active field of computer vision. Tremendous amount of work has been done in the field in recent years [Wang and Singh, 2003],[Wang *et al.*, 2003]

In order to obtain a semantic description of the content of a scene, we do not need to use all the available information. What is happening in a scene can be determined by monitoring the temporal transitions of the scene's non-static elements. As far as humans in the scene are concerned, this would translate in tracking the motion of the hands, head or even the entire body. Recently, tracking approaches based on particle filtering have been successfully used in order to track the state of a temporal event given a set of noisy observations [Isard and Blake, 1998]. In [Pitt and Shephard, 1999], an auxiliary particle filter is proposed as an extension to the classical particle filter, in order to deal with outlier problems. In [Pantic and Patras, 2006] the auxiliary particle filtering scheme proposed in [Pitt and Shephard, 1999] is used, in order to track 15 facial points in an input face-profile sequence. Another interesting approach that uses trajectories for activity recognition is presented in [Rao *et al.*, 2002],[Rao *et al.*, 2003]. The spatiotemporal curvatures of the trajectories of moving objects, such as the hands of the subject are used in order to represent human actions. The local maxima of these curvatures are view-invariant and are used for image sequence alignment and matching of the actions. In [Blank *et al.*, 2005] human actions are treated as three-dimensional shapes in space-time volume. The method utilizes properties of the solution to the Poisson equation to extract space-time features such as local space-time saliency, action dynamics, shape structure and orientation, while spectral clustering is used in order to group similar actions. In [Zelnik-Manor and Irani, 2001], long video sequences are segmented in the time domain by detecting single events in them. The detection is done without prior knowledge of the types of events, their models, or their temporal extent. The method can be used for event-based indexing even when only one short example-clip is available. In [Stauffer, 1999] , an adaptive background estimation algorithm is presented in order to distinguish moving objects from their background. Persisting objects in the scene are considered to be part of the background and are modelled using a Gaussian mixture, whose parameters are updated for every incoming frame in the sequence. In [Vlachos *et al.*, 2002], [Buzan *et al.*, 2004] a Longest Common Subsequence (LCSS) algorithm is introduced in order to obtain a measure of similarity between individual trajectories. The algorithm is implemented via a dynamic programming approach and works by detecting common sub-trajectories, maintaining the ordering of the consisting points of the original trajectories.

A wide variety of hand and body tracking methods use some form of markers in order to assist the initialization and the overall operation of the tracking process [Figueroa *et al.*, 1998],[Moeslund and Nrgaard, 2003]. In order to avoid the use of markers, an interesting alternative could be the use of interesting points for tracker initialization. According to Haralick and Shapiro [Haralick and Shapiro, 1993] an interesting point is a) distinguishable from its neighbors and b) its position is invariant with respect to the expected geometric transformation and to radiometric distortions. Gilles introduces the notion of saliency in terms of local signal complexity or unpredictability in [Gilles, 1998]. Kadir and Brady [Kadir and Brady, 2000] extend the original Gilles algorithm and estimate the information content of pixels in circular neighborhoods at different scales in terms of the entropy. Local extremes of changes in the entropy across scales are detected and the saliency of each point at a certain scale is defined in terms of the entropy and its rate of change at the scale in question.

In this work, we detect spatiotemporal features in given image sequences by extending in the temporal direction the salient feature detector developed in [Kadir and Brady, 2000]. The detected salient points correspond to peaks in activity variation such as the edges of a moving object. Like in [Kadir and Brady, 2000], we automatically detect the scales at which the entropy achieves local maxima and form spatiotemporal salient regions by clustering spatiotemporal points with similar location and scale. We derive a suitable distance measure between sets of salient regions, which is based on the Chamfer distance, and we optimize this measure with respect to a number of temporal and scaling parameters. In this way we achieve invariance against scaling and we eliminate the temporal differences between the representations. We extend our previous work on salient points presented at [Oikonomopoulos *et al.*, 2005] by using the detected salient regions in order to initialize a tracking scheme based on the auxiliary particle filter, proposed in [Pitt and Shephard, 1999]. Each image sequence is then represented as a set of short trajectories. The spatiotemporal coordinates of the points that consist the extracted trajectories are appropriately transformed according to the parameters that were estimated in the Chamfer distance optimization step. We use the adaptive background estimation algorithm presented in [Stauffer, 1999] in order to model the background in the available sequences and to improve the overall quality of the implemented tracking scheme. We use a variant of the Longest Common Subsequence algorithm (LCSS) that was proposed in [Vlachos *et al.*, 2002],[Buzan *et al.*, 2004] in order to compare different sets of trajectories. We use Relevance Vector Machines in order to address the classification problem. We propose new kernels for use by the RVM, which are specifically tailored to the proposed short trajectory representation. The basis of these kernels is the modified LCSS distance of the previous step.

We test the proposed method using real image sequences of subjects performing several aerobic exercises. Possible applications lie in the area of e-health, where the development of non-stationary, non-intrusive, non-invasive monitoring inside and outside the clinical environment is essential, due to demanding patients, aging population and rising costs. The method can be realized as an adaptive system that will be able to monitor and assess the correctness of the performed exercise, and will provide an appropriate alternative (senior) fitness plan, assisting in this way nurses,physical therapists and family members. The system can also be configured for use at home, to accommodate elderly but otherwise healthy patients or patients suffering from conditions like rheumatism and chronic pain.

The remainder of the paper is organized as follows: In section 2, the spatiotemporal feature detector used is described, along with the proposed space-time warping technique. In section 3, the auxiliary particle filter that was used is briefly analyzed along with the background estimation model that was utilized. In section 4 the proposed kernel-based recognition method is described. In section 5, we present our experimental results, and in section 6, final conclusions are drawn.

2 Spatiotemporal Salient Points

2.1 Spatiotemporal Saliency

Let us denote by $N_c(s, \vec{v})$ the set of pixels in an image I that belong to a circular neighborhood of radius s, centered at pixel $\vec{v} = (x, y)$. In [Kadir and Brady, 2000], in order to detect salient points in static images, Kadir and Brady define a saliency measure $y_D(s, \vec{v})$ based on measuring changes in the information content of N_c for a set of different circular radiuses (i.e. scales). In order to detect spatiotemporal salient points at peaks of activity variation we extend the Kadir's detector by considering cylindrical spatiotemporal neighborhoods at different spatial radiuses s and temporal depths d. More specifically, let us denote by $N_{cl}(\vec{s}, \vec{v})$ the set of pixels in a cylindrical neighborhood of scale $\vec{s} = (s, d)$ centered at the spatiotemporal point $\vec{v} = (x, y, t)$ in the given image sequence. At each point $\vec{v}$ and for each scale $\vec{s}$ we will define the spatiotemporal saliency $y_D(\vec{s}, \vec{v})$ by measuring the changes in the information content within $N_{cl}(\vec{s}, \vec{v})$. Since we are interested in activity within an image sequence, we consider as input signal the convolution of the intensity information with a first-order Gaussian derivative filter. Formally, given an image sequence $I_0(x, y, t)$ and a filter G_t, the input signal that we use is defined as:

$$I(x, y, t) = G_t * I_0(x, y, t)\,. \qquad (1)$$

For each point $\vec{v}$ in the image sequence, we calculate the Shannon entropy of the signal histogram in a cylindrical neighborhood $N_s(\vec{s}, \vec{v})$ around it. That is,

$$H_D(s, d, \vec{v}) = -\sum_{q \in D} p(q, s, d, \vec{v}) \log p(q, s, d, \vec{v})\,, \qquad (2)$$

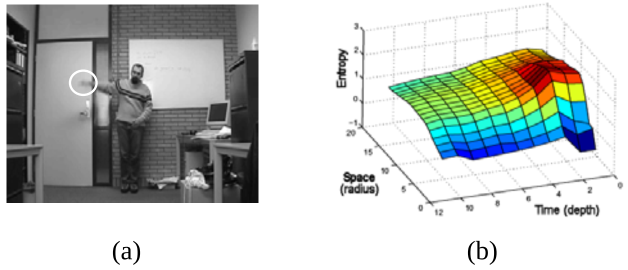
The set of scales at which the entropy is peaked is given by:

$$\begin{aligned}\hat{S}_p = \{(s, d) : H_D(s-1, d, \vec{v}) < H_D(s, d, \vec{v}) > H_D(s+1, d, \vec{v})\\ \wedge H_D(s, d-1, \vec{v}) < H_D(s, d, \vec{v}) > H_D(s, d+1, \vec{v})\}\end{aligned} \qquad (3)$$

The saliency measure at the candidate scales is given by:

$$y_D(s, d, \vec{v}) = H_D(s, d, \vec{v}) W_D(s, d, \vec{v}), \quad \forall\, (s, d) \in \hat{S}_p\,, \qquad (4)$$

The first term of eq. 4 is a measure of the variation in the information content of the signal. The weighting function



(a) (b)

Figure 1: (a) Single frame from a sample image sequence where the subject is raising its right hand and (b) the corresponding entropy plot as a function of the spatial radius and temporal depth of all the applied cylindrical neighborhoods. The origin of all the applied cylindrical neighborhoods is the center of the white circle in (a).

$W_D(s, \vec{v})$ is a measure of how prominent the local maximum is at s, and is given by:

$$W_D(s, d, \vec{v}) = \frac{s^2}{2s-1} \sum_{q \in D} |p(q, s, d, \vec{v}) - p(q, s-1, d, \vec{v})|$$
$$+ d \sum_{q \in D} |p(q, s, d, \vec{v}) - p(q, s, d-1, \vec{v})|, \forall\, (s, d) \in \hat{S}_p \,, \quad (5)$$

where the values in front of each summation in the right part of eq. 5 are normalization factors. When a peak in the entropy for a specific scale is distinct, then the corresponding pixel probability density functions at the neighboring scales will differ substantially, giving a large value to the summations of eq. 5 and thus, to the corresponding weight value assigned. On the contrary, when the peak is smoother, then the summations in eq. 5 will have a smaller value. Let us note that we considered cylindrical neighborhoods for simplicity reasons. However, more complicated shapes, such as elliptical neighborhoods at different orientations and with different axes ratios could be considered.

In Fig. 1(a), a single frame from a sample image sequence is presented, where the subject is raising its right hand. By selecting as origin the center pixel of the drawn white circle, we apply a number of cylindrical neighborhoods of various scales in the sequence and we calculate the corresponding entropy values. The result is shown in Fig. 1(b), where the various entropy values are plotted with respect to the radiuses and depths of the corresponding cylindrical neighborhoods. The scale which corresponds to the distinct peak of the plot is considered candidate salient scale, and is assigned a saliency value, according to eq. 4.

2.2 Salient Regions

The analysis of the previous section leads to a set of candidate spatiotemporal salient points $S = \{(\vec{s}_i, \vec{v}_i, y_{D,i})\}$, where $\vec{v}_i = (x, y, t)$, $\vec{s}_i$ and $y_{D,i}$ are respectively, the position vector, the scale and the saliency value of the feature point with index i. In order to achieve robustness against noise we follow a similar approach as that in [Kadir and Brady, 2000] and develop a clustering algorithm, which we apply to the detected salient points. By this we define salient regions instead of salient points, the location of which should be more stable than the individual salient points, since noise is unlikely to affect all of the points within the region in the same way. The proposed algorithm removes salient points with low saliency and creates clusters that are a) well localized in space, time and scale, b) sufficiently salient and c) sufficiently distant from each other. The steps of the proposed algorithm can be summarized as follows:

1. Derive a new set S_T from S by applying a global threshold T to the saliency of the points that consist S. Thresholding removes salient points with low saliency, that is,
$$S_T = \{(\vec{s}_i, \vec{v}_i, y_{D,i}) : y_{D,i} > T\} \,. \quad (6)$$
2. Select the point i in S_T with the highest saliency value and use it as a seed to initialize a salient region R_k. Add nearby points j to the region R_k as long as the intra-cluster variance does not exceed a threshold T_V. That is, as long as
$$\frac{1}{|R_k|} \sum_{j \in R_k} c_j^2 < T_V \,, \quad (7)$$
where R_k is the set of the points in the current region k and c_j is the Euclidean distance of the jth point from the seed point i.
3. If the overall saliency of the region R_k is lower than a saliency threshold T_S,
$$\sum_{j \in R_k} y_{D,j} \leq T_S \,, \quad (8)$$
discard the points in the region back to the initial set of points and continue from step 2 with the next highest salient point. Otherwise, calculate the Euclidean distance of the center of region R_k from the center of salient regions already defined, that is, from salient regions $R_{k'}, k' < k$.
4. If the distance is lower than the average scale of R_k, discard the points in R_k back to the initial set of points, and continue with the next highest salient point. Otherwise, accept R_k as a new cluster and store it as the mean scale and spatial location of the points in it.
5. Form a new set S_T consisting of the remaining salient points, increase the cluster index k and continue from step 2 with the next highest salient point.

By setting the threshold T_V in step 2, we define clusters that have local support and are well localized in space and time. In addition, we want to take the saliency of the points into consideration such that the overall saliency of the region is sufficient. We do this in step 3, by setting a saliency threshold, T_S. Finally, the purpose of step 4 is to accept clusters that are sufficiently distant from each other. To summarize, a new cluster is accepted only if it has sufficient local support, its overall saliency value is above the saliency threshold, and it is sufficiently distant in terms of Euclidean distance from already existing clusters.

2.3 Space-Time Warping

There is a large amount of variability between feature sets due to differences in the execution speed of the corresponding actions. Furthermore, we need to compensate for possible shifting of the representations forward or backward in time, caused by imprecise segmentation of the corresponding actions. To cope with both these issues, we propose a linear space-time warping technique with which we model variations in time using a time-scaling parameter a and a time-shifting parameter b. In addition, in order to achieve invariance against scaling, we introduce a scaling parameter c in the proposed warping technique. To accommodate this procedure, we propose the Chamfer distance as an appropriate distance measure, in order to cope with unequal number of features between different sets of salient points. More specifically, for two feature sets $F = \{(x_i, y_i, t_i), 1 \leq i \leq M\}$ and $F' = \{(x'_j, y'_j, t'_j), 1 \leq j \leq M'\}$ consisting of an M and M' number of features, respectively, the Chamfer distance of the set F from the set F' is defined as follows:

$$D(F, F') = \frac{1}{M} \sum_{i=1}^{M} \min_{j=1}^{M'} \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2 + (t'_j - t_i)^2} \,. \tag{9}$$

From eq. 9 it is obvious that the selected distance measure is not symmetrical, as $D(F, F') \neq D(F', F)$. For recognition purposes, it is desirable to select a distance measure that is symmetrical. A measure that satisfies this requirement is the average of $D(F, F')$ and $D(F', F)$, that is,

$$D_c(F, F') = \frac{1}{2}(D(F, F') + D(F', F)) \,. \tag{10}$$

Let us denote by $F_w = \{(cx_i, cy_i, at_i - b), 1 \leq i \leq M\}$ the feature set F with respect to feature set F'. Then, the distance between F' and F_w is given by eq. 9 as:

$$D(F_w, F') = \frac{1}{M} \sum_{i=1}^{M} \min_{j=1}^{M'} \sqrt{(x'_j - cx_i)^2 + (y'_j - cy_i)^2 + (t'_j - at_i + b)^2} \,. \tag{11}$$

Similarly, the feature set F' with respect to feature set F can be represented as $F'_w = \{(\frac{1}{c}x'_j, \frac{1}{c}y'_j, \frac{1}{a}t'_j + b), 1 \leq j \leq M'\}$ and their distance as:

$$D(F'_w, F) = \frac{1}{M'} \sum_{j=1}^{M'} \min_{i=1}^{M} \sqrt{(x_i - \frac{1}{c}x'_j)^2 + (y_i - \frac{1}{c}y'_j)^2 + (t_i - \frac{1}{a}t'_j - b)^2} \,. \tag{12}$$

The distance to be optimized follows from the substitution of eq. 11 and eq. 12 to eq. 10. We follow an iterative gradient descent approach for the adjustment of the a, b and c parameters. The update rules are given by:

$$a^{n+1} = a^n - \lambda_1 \frac{\partial D_c}{\partial a^n}, \tag{13}$$

$$b^{n+1} = b^n - \lambda_2 \frac{\partial D_c}{\partial b^n}, \tag{14}$$

$$c^{n+1} = c^n - \lambda_3 \frac{\partial D_c}{\partial c^n}\,, \tag{15}$$

where $\lambda_1, \lambda_2, \lambda_3$ are the learning rates and n is the iteration index. The algorithm iteratively adjusts the values of a, b and c towards the minimization of the Chamfer distance between the two feature sets, given by eq. 10. The iterative procedure stops when the values of a, b and c do not change significantly or after a fixed number of iterations.

3 Tracking

3.1 Auxiliary Particle Filtering

Recently, particle filtering tracking schemes [Isard and Blake, 1998], [Pitt and Shephard, 1999], have been successfully used [Su *et al.*, 2004], [Pantic and Patras, 2006], [Patras and Pantic, 2005] in order to track the state of a temporal event given a set of noisy observations. Its ability to maintain simultaneously multiple solutions, called particles, makes it particularly attractive when the noise in the observations is not Gaussian and makes it robust to missing or inaccurate data.

The particle filtering tracking scheme described in this section is initialized at the spatiotemporal salient points that are detected using the procedure of section 2. Let c denote the template that contains the color information in a rectangular window centered at each detected salient point and α denote the unknown location of the facial feature at the current time instant. Furthermore, let us denote by $Y = \{y^1, \ldots, y^-, y\}$ the observations up to the current time instant. The main idea of the particle filtering is to maintain a particle based representation of the a posteriori probability $p(\alpha|Y)$ of the state α given all the observations Y up to the current time instance. The distribution $p(\alpha|Y)$ is represented by a set of pairs (s_k, π_k) such that if s_k is chosen with probability equal to π_k, then it is as if s_k was drawn from $p(\alpha|Y)$. Our knowledge about the a posteriori probability is updated in a recursive way. Suppose that we have a particle based representation of the density $p(\alpha^-|Y^-)$, that is we have a collection of K particles and their corresponding weights (i.e. (s^-_k, π^-_k)). Then, the Auxiliary Particle Filtering can be summarized as follows:

1. Propagate all particles s^-_k via the transition probability $p(\alpha|\alpha^-)$ in order to arrive at a collection of K particles μ_k.
2. Evaluate the likelihood associated with each particle μ_k, that is let $\lambda_k = p(y|\mu_k; c)$.

   For the definition of $p(y|\mu_k; c)$ we use, in this paper, the observation model described in [Patras and Pantic, 2005].
3. Draw K particles s^-_k from the probability density that is represented by the collection $(s^-_k, \lambda_k\pi^-_k)$. In this way, the auxiliary particle filter favors particles with high λ_k, that is particles which, when propagated with the transition density, end up at areas with high likelihood.
4. Propagate each particle s^-_k with the transition probability $p(\alpha|\alpha^-)$ in order to arrive at a collection of K particles $s_k{}'$.
5. Assign a weight $\pi_k{}'$ to each particle as follows,

$$w_k{}' = \frac{p(y|s_k{}'; c)}{\lambda_k}, \quad \pi_k{}' = \frac{w_k{}'}{\sum_j w_j} \tag{16}$$

   This results in a collection of K particles and their corresponding weights (i.e. $\{(s_k{}', \pi_k{}')\}$) which is an approximation of the density $p(\alpha|Y)$.

3.2 Online Background Estimation

The particle filtering tracking scheme described in the previous section is initialized at the spatiotemporal salient points that are detected using the procedure described in section 2. As indicated from eq. 1, the input signal that is used is the convolution of the original image sequence with a Gaussian derivative filter along the temporal dimension. The result of this is that the detected salient points are localized on the edges of the moving objects existing in the scene, rather than on the objects themselves. This fact may deteriorate the output of the tracker used, since the patches of the sequence that are being tracked also include a considerable portion of the scene's background. For this reason, we implement the adaptive background estimation algorithm described in [Stauffer, 1999], in order to determine which pixels belong to the foreground and which ones to the background. According to this algorithm, the values of a particular pixel over time are considered as a temporal process. At each time t, what is known about a particular pixel (x_0, y_0) is its history:

$$\{X_1, \ldots, X_t\} = \{I(x_0, y_0, i) : 1 \leq i \leq t\}, \tag{17}$$

where I is the image sequence. The recent history of each pixel is modeled by a mixture of K Gaussian distributions. The probability of observing the current pixel value is given by:

$$P(X_t) = \sum_{i=1}^{K} w_{i,t} \cdot \eta(X_t, \mu_{i,t}, \Sigma_{i,t}), \tag{18}$$

where K is the number of distributions, $w_{i,t}$ is an estimate of the weight of the i_{th} Gaussian in the mixture at time t, $\mu_{i,t}$ is the mean value of the i_{th} Gaussian in the mixture at time t, $\Sigma_{i,t}$ is the covariance matrix of the i_{th} Gaussian in the mixture at time t, and η is a Gaussian probability density function. K was set to 3, and the covariance matrix Σ is assumed to be diagonal, meaning that the RGB values of the pixels are assumed to be uncorrelated.

The parameters of each Gaussian mixture were initially estimated using the Expectation-Maximization (EM) algorithm and by using a small portion of the available data (i.e. the first few frames of the image sequence). Subsequently at each new frame t we follow an update procedure similar to the one of [Stauffer, 1999]. Every new pixel value X_t is checked against the existing K distributions until a match is found. A match is defined if the current pixel is within 3 standard deviations of a distribution. In case a match is found the parameters of the Gaussians are updated. If none of the K distributions match the current pixel value, the least probable distribution is replaced with a distribution with the current value as its mean value, an initially high variance, and low prior weight.

At each iteration of the particle filtering tracking scheme of section 3.1, every new particle is evaluated based on an invariant colour distance between the initial template (centered at the initializing spatiotemporal salient point) and the block that corresponds to the particle that is being evaluated. In order to take the estimated background model into account, we add an additional cost in the evaluation process of each new particle. The additional cost for every pixel is equal to the probability that the pixel belongs to the current background model, that is,

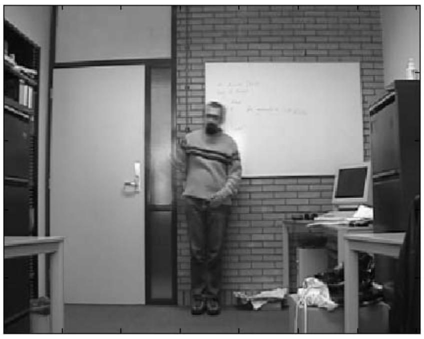


Figure 2: Initial estimation of the background for an action where the subject is just raising its right hand

$$C_{i,j,t} = \sum_{i=1}^{K} w_{i,j} \eta(X_{j,t}, \mu_{i,j,t}, \Sigma_{i,j,t}), \tag{19}$$

where K is the number of distributions, $w_{i,j,t}$ is an estimate of the weight of the i_{th} Gaussian in the mixture for the pixel j at time t, $\mu_{i,j,t}$ is the mean value of the i_{th} Gaussian in the mixture for the pixel j at time t and $\Sigma_{i,j,t}$ is the covariance matrix of the i_{th} Gaussian in the mixture for pixel j at time t.

If a pixel in the block belongs to the background, then eq. 19 will assign a large cost to that pixel, since the resulting probability will be high. If most pixels in the block belong to the background, then the additional cost to that block will also be large and consequently, a smaller weight will be assigned to it by the particle filter. In this way, the tracking scheme favors blocks that contain larger number of foreground pixels and assigns larger weights to the corresponding particles.

In Fig. 2 the initial background model that was estimated for an action where the subject is raising its right hand is presented. As can be seen from the figure, parts of the body that do not present significant motion are also considered part of the background. On the other hand, fast moving parts (e.g. right hand) are considered to belong to the foreground and are not included in the estimation.

4 Recognition

4.1 Longest Common Subsequence (LCSS) Algorithm

Using the analysis of the previous sections, we represent a given image sequence by a set of short trajectories, where each trajectory is initialized at a point which is considered salient both in space and time. Formally, an image sequence is represented by a set of trajectories $\{A_i\}, i = 1 \ldots K$, where K is the number of trajectories that consist the set. Each trajectory is defined as $A_i = ((t_{i,n}, x_{i,n}, y_{i,n}), \ldots)$, $n = 1 \ldots N$, where $t_{i,n}, x_{i,n}, y_{i,n}$ are spatiotemporal coordinates and N is the number of samples that consist A_i. Let us define another trajectory set $\{B_j\}$, $j = 1 \ldots L$ representing a different image sequence. Similar to $\{A_i\}$, the trajectories in $\{B_j\}$ are defined as $B_j = ((t_{j,m}, x_{j,m}, y_{j,m}), \ldots)$,

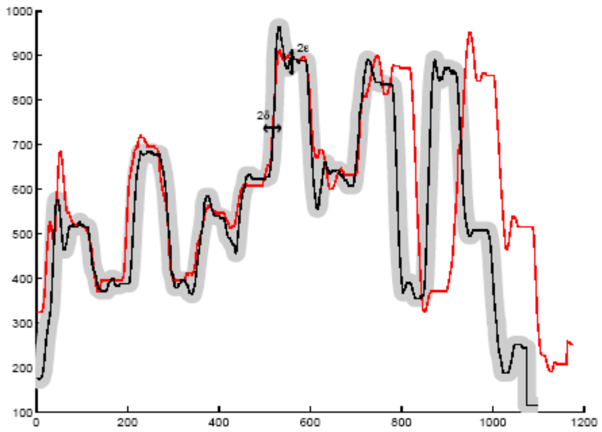


Figure 3: The notion of the LCSS matching within a region of δ and ϵ of a trajectory.

$m = 1 \ldots M$, where M is the number of individual trajectories that consist $\{B_j\}$. We use a variant of the LCSS algorithm presented at [Vlachos *et al.*, 2002], [Buzan *et al.*, 2004] in order to compare the two sets. Before we proceed with the comparison, we align the two sets in space and time using the a, b and c parameters that were computed using the procedure of section 2.3. Let us define the function $Head(A_i) = ((t_{i,n}, x_{i,n}, y_{i,n})), n = 1 \ldots N-1$, that is, the individual trajectory A_i reduced by one sample. Then, according to the LCSS algorithm, the distance between individual trajectories A_i and B_j is given by:

$$d_L(A_i, B_j) = \begin{cases} 0, \quad if\ A_i\ or\ B_j\ is\ empty \\ \\ d_e((t_{i,n}, x_{i,n}, y_{i,n}), (t_{j,m}, x_{j,m}, y_{j,m})) \\ +d_L(Head(A_i), Head(B_j)), \\ if\ \ |t_{i,n} - t_{j,m}| < \delta\ and\ \ |x_{i,n} - x_{j,m}| < \varepsilon \\ and\ \ |y_{i,n} - y_{j,m}| < \varepsilon \\ \\ \max(d_L(Head(A_i), B_j), d_L(A_i, Head(B_j))) + p, \\ otherwise \end{cases}, \tag{20}$$

where d_e is the Euclidean distance, δ controls how far in time we can go in order to match a given point from one trajectory to a point in another trajectory, ϵ is the matching threshold and p is a penalty cost in case of mismatch. The notion of the LCSS distance of eq. 20 is depicted in Fig. 3.

Subsequently, the distance between sets $\{A_i\}$ and $\{B_j\}$is defined as follows:

$$D_L(\{A_i\}, \{B_j\}) = \frac{1}{K} \sum_i \min_j d_L(A_i, B_j) + \frac{1}{L} \sum_j \min_i d_L(B_j, A_i), \tag{21}$$

that is, the average over the set of the minimum distances, as they have been defined in eq. 20, between the K trajectories of set $\{A_i\}$ and the L trajectories of set $\{B_j\}$.

4.2 Relevance Vector Machine Classifier

We propose a classification scheme based on Relevance Vector Machines [Tipping, 1999] in order to classify given examples of human actions. A Relevance Vector Machine (RVM) is a probabilistic sparse kernel model identical in functional form to the Support Vector Machines (SVM). In their simplest form, Relevance Vector Machines attempt to find a hyperplane defined as a weighted combination of a few Relevance Vectors that separate samples of two different classes. In contrast to SVM, predictions in RVM are probabilistic. Given a dataset of N input-target pairs $\{(F_n, l_n), 1 \leq n \leq N\}$, an RVM learns functional mappings of the form:

$$y(F) = \sum_{n=1}^{N} w_n K(F, F_n) + w_0 \,, \tag{22}$$

where $\{w_n\}$ are the model weights and $K(.,.)$ is a Kernel function. Gaussian or Radial Basis Functions have been extensively used as kernels in RVM. In our case, we use as a kernel a Gaussian Radial Basis Function defined by the distance measure of eq. 21. That is,

$$K(F, F_n) = e^{-\frac{D_L(F, F_n)^2}{2\eta}} \,, \tag{23}$$

where η is the Kernel width. RVM performs classification by predicting the posterior probability of class membership given the input F. The posterior is given by wrapping eq. 22 in a sigmoid function, that is:

$$p(l|F) = \frac{1}{1 + e^{-y(F)}} \tag{24}$$

In the two class problem, a sample F is classified to the class $l \in [0, 1]$, that maximizes the conditional probability $p(l|F)$. For L different classes, L different classifiers are trained and a given example F is classified to the class for which the conditional distribution $p_i(l|F)$,$1 \leq i \leq L$ is maximized, that is:

$$Class(F) = \arg\max_i (p_i(l|F)) \,. \tag{25}$$

5 Experimental Results

For the evaluation of the proposed method, we use aerobic exercises as a test domain. Our dataset consists of 12 different aerobic exercises, performed by amateurs, that have seen a video with an instructor performing the same set of exercises. Each exercise is performed twice by four different subjects, leading to a set of 96 corresponding feature sets.

In order to illustrate the ability of the proposed method to extract the kind of motion performed, we present in Fig. 4 the trajectories that were extracted from two different actions along with a snapshot of the corresponding actions. The salient points that are visible in the upper part of the figure were used in order to extract some of the trajectories presented in the lower part of the same Figure. Furthermore, the extracted trajectory set seems to correctly capture the pattern of the motion performed. This can easily be observed from the arch-like trajectories of the lower part of the figure, which correspond to the motion of the subjects' hands.

In order to classify a test example using the Relevance Vector Machines, we constructed 12 different classifiers, one for each class, and we calculated for each test example F the conditional probability $p_i(l|F), 1 \leq i \leq 12$. Each example was assigned to the class for which the corresponding classifier

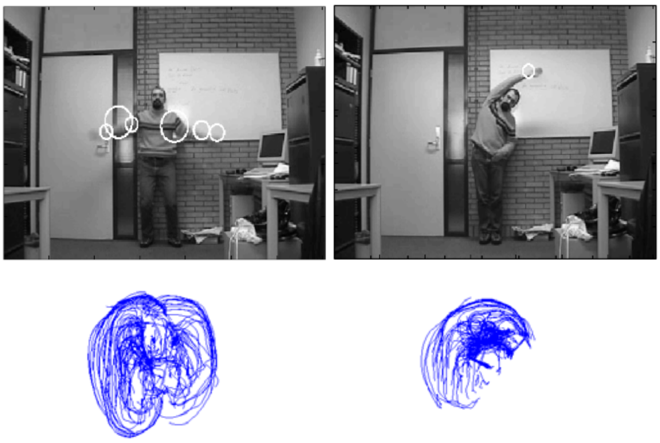


Figure 4: Extracted trajectories for two different actions.

Class Labels	1	2	3	4	5	6
RVM Recall	1	1	1	1	0.5	0.5
RVM Precision	1	1	1	1	0.44	0.4
Class Labels	**7**	**8**	**9**	**10**	**11**	**12**
RVM Recall	1	0.88	0.63	0.63	0.88	1
RVM Precision	1	1	0.63	0.83	0.88	1

Table 1: Recall and Precision rates for the kNN and RVM classifiers

provided the maximum conditional probability, as depicted in eq. 25. Note that for estimating each of the $p_i(l|F)$, an RVM is trained by leaving out the example F as well as all other instances of the same exercise that were performed by the subject from F. The corresponding recall and precision rates, calculated as an average of all test trials, are given in Table 1. The total recognition rate is equal to 80.61%, which is a relatively good performance, given the small number of examples with respect to the number of classes, and the fact that the subjects were not trained. In Table 2 the confusion matrix generated by the RVM classifier is also given.

Class labels	1	2	3	4	5	6	7	8	9	10	11	12	Total
1	**8**	0	0	0	0	0	0	0	0	0	0	0	8
2	0	**8**	0	0	0	0	0	0	0	0	0	0	8
3	0	0	**8**	0	0	0	0	0	0	0	0	0	8
4	0	0	0	**8**	0	0	0	0	0	0	0	0	8
5	0	0	0	0	**4**	5	0	0	0	0	0	0	9
6	0	0	0	0	4	**3**	0	0	3	0	0	0	10
7	0	0	0	0	0	0	**8**	0	0	0	0	0	8
8	0	0	0	0	0	0	0	**7**	0	0	0	0	7
9	0	0	0	0	0	0	0	1	**5**	2	0	0	8
10	0	0	0	0	0	0	0	0	0	**5**	1	0	6
11	0	0	0	0	0	0	0	0	0	1	**7**	0	8
12	0	0	0	0	0	0	0	0	0	0	0	**8**	8
Total	8	8	8	8	8	8	8	8	8	8	8	8	8

Table 2: RVM Confusion Matrix

The confusion matrix in Table 2 conceals the fact that for some of the misclassified examples the probability assigned by the RVM classifier to the correct matching move might be very close to the probability assigned to the move actually selected by the classifier. We used the average ranking percentile in order to extract this kind of information and to measure the overall matching quality of our proposed algorithm. Let us denote with r^{F_n} the position of the correct match for the test example $F_n, n = 1 \ldots N_2$, in the ordered list of N_1 match values. Rank r^{F_n} ranges from $r = 1$ for a perfect match to $r = N_1$ for the worst possible match. Then, the average ranking percentile is calculated as follows:

$$\overline{r} = \left(\frac{1}{N_2} \sum_{n=1}^{N_2} \frac{N_1 - r^{F_n}}{N_1 - 1} \right) 100\%. \qquad (26)$$

Since our dataset consists of 96 test image sequences divided in 12 separate classes, it follows that $N_1 = 12$ and $N_2 = 96$. Each of the 12 match values are provided for each example by the 12 trained RVM classifiers. The average ranking percentile for the RVM classifier is 94.5%. Its high value shows that for the majority of the missclassified examples, the correct matches are located in the first positions in the ordered list of match values.

6 Conclusions

In this work, previous work on spatiotemporal saliency was enhanced in order to extract a number of short trajectories from given image sequences. Each detected spatiotemporal point was used in order to initialize a tracker based on auxiliary particle filtering. A background estimation model was also implemented and incorporated into the particle evaluation process, in order to deal with inadequate localization of the initialization points and to improve, thus, the performance of the tracker. A variant of the LCSS algorithm was used in order to compare different sets of trajectories. The derived LCSS distance was used in order to define a kernel for the RVM classifier that was used for recognition. We have illustrated the efficiency of our representation in recognizing human actions using as a test domain aerobic exercises. Finally, we presented results on real image sequences that illustrate the consistency in the spatiotemporal localization and scale selection of the proposed method.

Acknowledgements

This work has been partially supported by the Dutch-French Van Gogh program (project VGP-62-604) and the work of A. Oikonomopoulos has been supported by the Greek State Scholarship Foundation (IKY). The data set was collected while I. Patras was with the ISIS group at the University of Amsterdam.

References

[Blank *et al.*, 2005] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri. Actions as space-time shapes. *Proc. IEEE Int. Conf. Computer Vision*, 2:1395 – 1402, 2005.

[Buzan *et al.*, 2004] D. Buzan, S. Sclaroff, and G. Kollios. Extraction and clustering of motion trajectories in video. *Proceedings, International Conference on Pattern Recognition*, 2:521 – 524, 2004.

[Figueroa *et al.*, 1998] P.J. Figueroa, N.J. Leitey, R.L. Barros, and R. Brenzikofer. Tracking markers for human motion analysis. *Proc. of IX European Signal Processing Conf., Rhodes, Greece*, pages 941 – 944, 1998.

[Gilles, 1998] S. Gilles. *Robust Description and Matching of Images*. PhD thesis, University of Oxford, 1998.

[Haralick and Shapiro, 1993] R. Haralick and L. Shapiro. *Computer and Robot Vision II*. Addison-Wesley, 1993. Reading, MA.

[Isard and Blake, 1998] M. Isard and A. Blake. Condensation conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5 – 28, 1998.

[Kadir and Brady, 2000] T. Kadir and M. Brady. Scale saliency: a novel approach to salient feature and scale selection. *International Conference on Visual Information Engineering*, pages 25 – 28, 2000.

[Moeslund and Nrgaard, 2003] T. Moeslund and L. Nrgaard. A brief overview of hand gestures used in wearable human computer interfaces. *Technical Report CVMT 03-02, ISSN 1601-3646*, 2003.

[Oikonomopoulos *et al.*, 2005] A. Oikonomopoulos, I. Patras, and M. Pantic. Spatiotemporal Salient Points for Visual Recognition of Human Actions. *IEEE Trans. Systems, Man and Cybernetics Part B*, 36(3):710 – 719, 2005.

[Pantic and Patras, 2006] M. Pantic and I. Patras. Dynamics of Facial Expressions-Recognition of Facial Actions and their Temporal Segments from Face Profile Image Sequences. *IEEE Trans. Systems, Man and Cybernetics Part B*, 36(2):433 – 449, 2006.

[Pantic *et al.*, 2006] M. Pantic, A. Pentland, A. Nijholt, and T. Huang. Human computing and machine understanding of human behavior: A survey. *International Conference on Multimodal Interfaces*, 2006.

[Patras and Pantic, 2005] I. Patras and M. Pantic. Tracking deformable motion. *IEEE International Conference on Systems, Man and Cybernetics*, pages 1066 – 1071, 2005.

[Pitt and Shephard, 1999] M.K. Pitt and N. Shephard. Filtering via simulation: auxiliary particle filtering. *J. American Statistical Association*, 94:590 –, 1999.

[Rao *et al.*, 2002] C. Rao, A. Yilmaz, and M. Shah. View-invariant representation and recognition of actions. *International Journal of Computer Vision*, 50(2):203–226, 2002.

[Rao *et al.*, 2003] C. Rao, A. Gritai, M. Shah, and T. Syeda-Mahmood. View-invariant alignment and matching of video sequences. *Proc. IEEE Int. Conf. Computer Vision*, 2:939–945, 2003.

[Stauffer, 1999] C. Stauffer. Adaptive background mixture models for real-time tracking. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, pages 246 – 252, 1999.

[Su *et al.*, 2004] C. Su, Y. Zhuang, L. Huang, and F. Wu. A two-step approach to multiple facial feature tracking: Temporal particle filter and spatial belief propagation. *Proc. IEEE Intl Conf. on Automatic Face and Gesture Recognition*, pages 433 – 438, 2004.

[Tipping, 1999] M.E. Tipping. The Relevance Vector Machine. *Advances in Neural Information Processing Systems*, pages 652 – 658, 1999.

[Vlachos *et al.*, 2002] M. Vlachos, G. Kollios, and D. Gunopulos. Discovering similar multidimensional trajectories. *Proc. International Conference on Data Engineering*, pages 673 – 684, 2002.

[Wang and Singh, 2003] J. J. Wang and S. Singh. Video analysis of human dynamics - A survey. *Real Time Imaging*, 9(5):321 – 346, 2003.

[Wang *et al.*, 2003] L. Wang, W. Hu, and T. Tan. Recent Developments in Human Motion Analysis. *Pattern Recognition*, 36(3):585 – 601, 2003.

[Zelnik-Manor and Irani, 2001] L. Zelnik-Manor and M. Irani. Event-based analysis of video. *Proceedings, IEEE Conference on Computer Vision and Pattern Recognition*, 2:123 – 130, 2001.

Human Gaze Differentiation for Man-Machine Interaction using a Hierarchical Solution for Eye Detection and Tracking

E. Regentova, V. Muthukumar, J. Zheng, A. Ponzio, T. Wu, Z. Devlin

University of Nevada Las Vegas,
Dept. of Electrical and Computer Engineering,
{venkim,regent}@ee.unlv.edu

Abstract

Eye tracking and gaze differentiation has been an important area of research for more than three decades, distinguished mainly by the field of application. The problem of identifying the gaze direction is complicated due to limitations of image modality, such as resolution, lighting conditions, variety of poses. In this paper, we present a system which implements eye detection, tracking and gaze differentiation. The basic approach is the hierarchical solution with successive refinement. Even though a number of steps is involved, the system is efficient due to its successive nature and the simplicity of each step, including a simple eye model. Face/eyes detection is done using artificial neural networks. The gaze differentiation method employs a two dimensional geometric mapping approach, where the position of the eye pupil is mapped onto the region of view on the screen. The process of eye tracking is solved by employing the mean-shift algorithm. The gaze differentiation accuracy of the proposed method was evaluated for nine regions on the computer screen, which corresponds to an application of Human-Computer Interaction for gaming machines. Experimental evaluation shows that the proposed eye tracking and gaze differentiation algorithm has an accuracy of 94.75% and is suitable for real-time application with a best case execution time of 0.238 sec. The algorithm also performs well for user head rotation and head tilt of 20 degrees. The method thus has a wide applicability and further refinement is expected to yield even higher accuracies and greater speed.

Keywords: HCI, Face Detection, Eye Detection, Eye Tracking, Gaze Differentiation, Neural Network, Mean-shift.

1. Introduction

Robust eye detection and tracking is an essential aspect of vision based man-machine interaction technology. This paper proposes an efficient method for the problem of eye detection, tracking and gaze differentiation for human computer interaction application. The application targeted in this work was human gaze as a virtual input for gaming machines. All parameters like the players distance from the camera and screen, illumination, gaze points, etc. where specific to a player in front of the gaming machine. The problem is divided into three sub-problems: 1) eye detection and eye modeling, 2) eye gaze differentiation and 3) eye tracking. Eye detection and tracking research is entering its fourth era, distinguished by the emergence of interactive applications [Duchowski, 2002]. Research in earlier eras includes: basic eye movement, perceptual span, behavioral movements in psychology and recording eye movements with increased accuracy. A typical eye tracking system can be classified as *interactive* or *diagnostic*. The interactive system is further classified as *selective* (point of gaze is analogous to user selection) or *gaze contingent* (knowledge of user gaze is used to facilitate rapid rendering of complex displays). The above classification can be further classified as *screen based* or *model based*. Eye tracking and gaze differentiation has applications in fields such as Computer Science [Ji and Yang, 2001, Ji and Zhu, 2002], NeuroScience [Robinson, 1968], Psychology [Rayner, 1998], Natural Tasks [Allopenna et. al., 1998], Industrial Engineering [Anders, 2001], Usability Evaluation [Vertegaal, 1999], etc.

For eye detection and tracking devices to be widely accepted in common environments such as homes and offices, its implementation needs to be non-intrusive. The existing methods of non-intrusive eye detection are mainly camera based and employ image processing techniques. Image processing based methods can be classified into two categories: traditional image based passive approaches and the active IR based approaches. The former approach processes the eye image based on the unique intensity distribution or shape of the eyes. Since eyes appear different from the rest of the face in intensity, texture and shape, and by exploiting these differences, eyes can be detected and tracked. The active IR based approach detects and tracks the pupil based on the pupil's unique intensity distribution under IR illumination, it is called the bright/dark pupil effect. Our main research interests are focused on the traditional image based passive approaches.

The traditional image based passive approaches can be broadly classified into two categories: appearance based methods [Pentland et. al., 1994, Huang and Wechsler, 1999] and feature based methods [Kawato and Ohya, 2000, Sirohey and Rosenfeld, 2001]. The appearance based methods detect eyes based on their photometric appearance. For this method to work, a large amount of training data needs to be collected. The training dataset needs to be as comprehensive as possible, and has to include different instances of eyes of different subjects, under different face orientations, and under different illumination conditions. This dataset is used to train a classifier such as a neural network or the support vector machine, and detection is achieved by classification. The general drawback of the appearance based method is that it is computational complex and requires a great amount of time to collect and refine the training data for the classifier. However, the classification process is relatively quick and although the general shape of the face can be detected and classified with relative ease, it is difficult to classify and analyze the details of the face.

Feature based methods mainly explore the intensity characteristics of the eyes and use those characteristics to identify the eye object. One example is the abrupt change of intensity at the boundary between the eye and the surrounding skin; distinct intensity levels of the white sclera region and dark iris region are also good indicators of the eye object. Experiments show that this method will fail if the eyes are closed or partially occluded by hair or face orientation. Moreover feature based techniques require manual initialization of the eye model in the first picture frame for better results. Generally, this method still requires a high contrast image to detect and track eye corners and to obtain a good edge image.

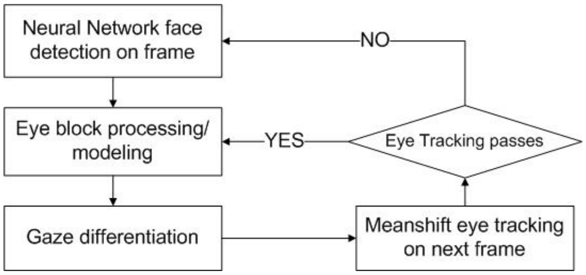

Neural Network face detection on frame
NO
Eye block processing/ modeling
YES
Eye Tracking passes
Gaze differentiation
Meanshift eye tracking on next frame

Figure 1: Basic flow the eye tracking and gaze differentiation system

The proposed eye tracking and gaze differentiation system in which the appearance based method (neural network) is first used to obtain the general location of the face and eyes. Next the feature based method is employed around the initially estimated eye region to accurately model the eyes. Based on the relative position of the pupil with respect to the contour of the eye generated by the eye model the precise gaze direction can be determined. Further tracking of the eyes is performed by the mean-shift algorithm. If in any case the mean-shift method fails to track the eye movement, neural network is invoked again to re-estimate the general location of the face and eye regions. It should be noted that in this scheme, even one eye detection can yield a sufficient accuracy for gaze differentiation. Figure 1, illustrates the basic flow of our system.

2. Face detection using neural network

Our method adopts the neural network face and eye detection framework proposed by Rowley et. al [Rowley et. al., 1998]. An initial step in the system is rotation of the blocks of the input image with a discrete step and then feeding them to the NN. This is done with the purpose of vertical alignment for which the NN is trained. The essential features of the NN training are eyes, mouth and the nose. This way the system is able to deliver coordinates of the detected face region and the approximate estimate of the eye position. However, this estimate is not sufficient for gaze differentiation. Figure 2, below illustrates example outputs of neural network face detection and eye centers estimation. The rectangle box indicates the detected face region, and the two crosses indicate the initial estimate of the eye centers.

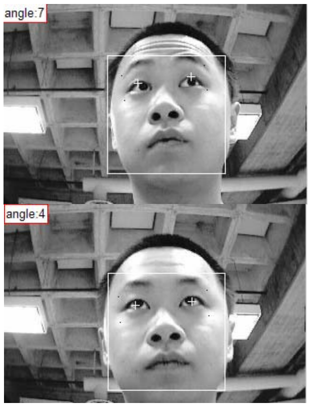


Figure 2: Results of the neural network output of face detection and eye block estimation.

3. Eye block analysis

3.1 Obtaining the Eye Block

The Neural Network algorithm outputs a rough estimate of the coordinates of two eyes. The next step of our system development is to extract the eye block from the face image. The distance '**d**' between the pupils (inter-pupil distance) and the angle of orientation of the pupil axis '**α**' is determined. Figure 3, demonstrates the parameters of **d** and **α**.

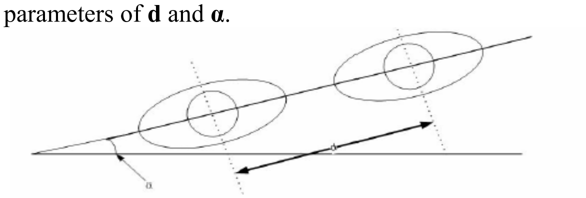


Figure 3: Inter-pupil distance d and angle of orientation α.

Once the inter-pupil distance and orientation are determined, an eye block is constructed and extracted for future processing. The eye block of length ‘L’ and height ‘H’ are proportional to the inter-pupil distance ‘**d’**, and tilted by an angle ‘**α’**. The optimal length ‘L’ and height ‘H’ to obtain the eye block for our application under consideration are experimentally determined to be L = 1.6 × **d** and H = 0.4 × **d** (when the eye is at a distance approx. 33~36 cm from the camera). Figure 4, shows the obtained eye block from the parameters ‘**d’** and ‘**α’**.

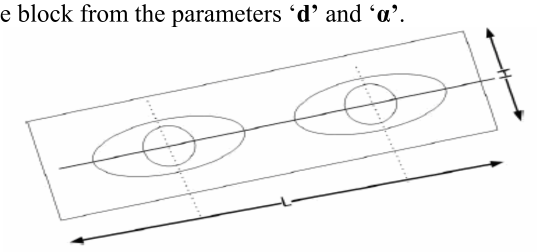


Figure 4: Eye block obtained from the inter-pupil distance d and angle of orientation α.

3.2 Eye detection: Initial Study

After obtaining the eye-block, further processing of the block is performed for obtaining the eye. Particularly, locations of the eye corners, eyelids and iris boundaries, as well as the pupil position are determined. Using this information, the eye is modeled as an ellipse, the iris as a circle, and the pupil as a single point within the iris region. Once the eyes are modeled, the gaze direction can be determined by evaluating the position of the pupil with respective to the eye modeled as an ellipse.

Thresholding is a way to separate the foreground object from the background. This is done by converting a gray-level image into a binary one, where image pixels whose intensity values exceed the specified threshold value are assigned to one category (object), and those pixels whose intensity values are below the threshold value as the background. The authors have experimented with thirteen different thresholding methods [Niemistö, 2004] and have identified two best thresholding methods 1) Entropy based [Kapur et. al., 1985] and 2) Intermeans [Otsu, 1979]) based on experiments. They are the most accurate in choosing the correct threshold value to separate the eyes from the background.

In the entropy based algorithm, the histogram of the image is divided into two probability distributions, one representing the objects (foreground) and one representing the background. It chooses the threshold value such that the sum of the entropies of the probability distributions is maximized.

In the Intermeans method, a threshold is selected such that the inter-class variance is maximized and the intra-class variance is minimized. The algorithm positions the threshold midway between the means of the two classes.

Integral projection [Feng and Yuen, 1998] functions are useful in locating important eye landmarks. Suppose I(x, y) is the intensity of a pixel at location (x, y) of an image, the horizontal integral projection function, HIPF of I(x, y) is defined as an integral of intensities along the vertical axis (cumulative vertical intensity at a given x value in image I), and the vertical integral projection function, VIPF is defined as an integral of intensities along the horizontal axis (cumulative horizontal intensity at a given y value), i.e.,

$$\textbf{HIPF(y)} = \int_{x1}^{x2} I(x,y)dx \quad (1)$$

$$\textbf{VIPF(x)} = \int_{y1}^{y2} I(x,y)dy \quad (2)$$

The peaks and dips in the observed integral profile functions indicate the positions of blobs of different intensities. In addition, the derivatives of the integral projection functions can be used to detect boundaries between different objects.

The eye block obtained is divided into two regions, one containing the left eye and one containing the right eye. The most intuitive way is to divide the block by the nose. As shown in Figure 5, looking at the profile graph generated from the horizontal integral projection function, it is clear that the nose is the region with the largest cumulative intensities, since it is outstanding and is illuminated more thoroughly than any other part of the face. Therefore the nose is used to divide the eye block into two separate eye objects and process them individually.

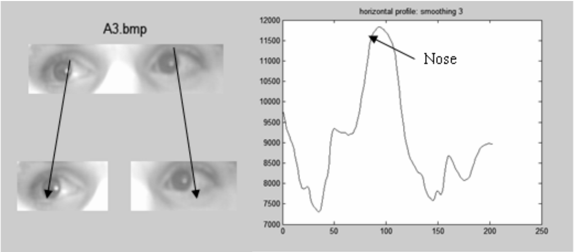


Figure 5: The process of dividing the eye block into two eye objects.

Next, the coordinates of the eye corners are determined by applying the entropy thresholding/Intermeans to the divided eye block. The results of this process are shown in Figure 6.

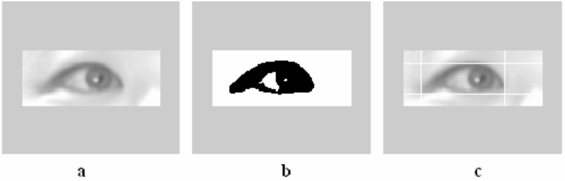


Figure 6: (a) The right eye image before thresholding, (b) after thresholding and (c) detected eye corners/lids.

The coordinates of the leftmost, rightmost, topmost and bottommost black pixels are determined and marked as the left eye corner, right eye corner, upper eye lid, and lower eye lid respectively as shown in Figure 6(c). To obtain the parameters of an ellipse for eye modeling, we calculate the center of this rectangle, the semi-major axis (from the center to the left/right eye corner), and the semi-minor axis (from the center to the upper/lower eye lid) of the ellipse. An ellipse with center **(a,b)**, semi-major axis of length **c/2**, and semi-minor of length **d/2**, has parametric equations:

$$x = a + c \bullet \cos(t) \qquad (3)$$

$$y = b + d \bullet \sin(t) \qquad (4)$$

Figure 7, shows the eye modeled as an ellipse.

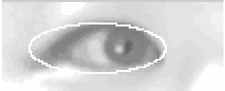


Figure 7: Eye modeled as an ellipse.

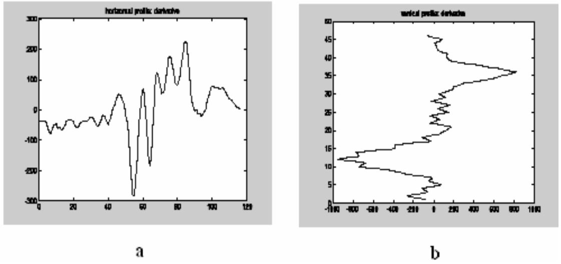


Figure 8: (a) The derivative of the horizontal projection profile generated from Figure 7. (b) The derivative of the vertical projection profile generated from Figure 7.

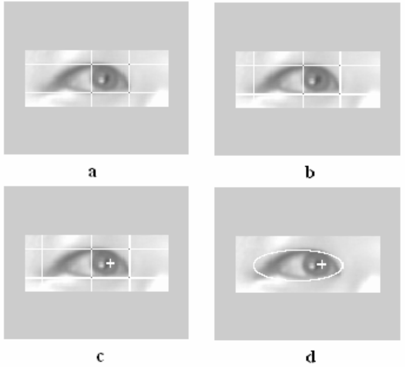


Figure 9: (a) original image with the marked edges of iris, (b) original image with marked edges of iris and eye corners/lids, (c) the pupil is defined as a center of iris rectangle, (d) combined ellipse and pupil image.

The next step in eye modeling is to obtain a circle of the iris. To model the iris, we determine the coordinates of the iris boundaries. This can be done by observing the derivative of the Integral Projection Function. The local minima shown in both, derivative of the horizontal and vertical projection profiles, represent transitions from a bright to a dark region in the image, and the local maxima represent transition from a dark to a bright region in the image. The smallest local minimum indicates the transition from the sclera to the iris region, and the largest local maximum indicates the transition from the iris to the sclera transition. As shown in Figure 8, the local minimum and maximum of the horizontal projection function derivative in the x-axis and the local minimum and maximum of the vertical projection function derivative in the y-axis are used to determine the boundary of the iris.

The maxima and minima locations define the iris boundaries. Straight lines drawn through these points form an iris rectangle (Figure 9a). Position of the pupil is defined as a center of the iris rectangle (Figure 9c).

The iris is modeled as a circle with center (**a, b)** which is a center of the iris rectangle, and radius **r** which is half of the average of height and width of the iris rectangle. The equation of the model is

$$(x-a)^2 + (y-b)^2 = r^2 \qquad (5)$$

The pupil is defined as a center of the circle. Figure 10, shows the combined pupil, iris, and eye model.

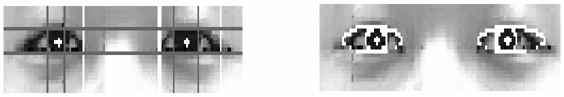


Figure 10: Eye model.

3.3 Analysis of eye modeling for gaze differentiation

For the gaze differentiation experiment, two test datasets, 1) with individuals looking at nine different positions on the computer screen (shown in Figure 11) and 2) with individual looking into and away from the screen were collected.

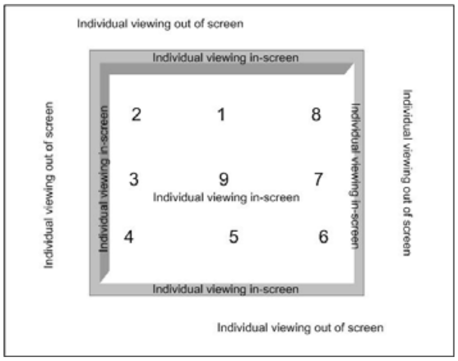

Individual viewing out of screen
Individual viewing in-screen
Individual viewing out of screen
Individual viewing in-screen
Individual viewing in-screen
Individual viewing out of screen
2
1
8
3
9
7
Individual viewing in-screen
4
5
6
Individual viewing in-screen
Individual viewing out of screen

Figure 11: Nine directions of gaze from the test sets.

Some results using images taken by the webcam of an individual looking at the nine different screen points/regions are shown in Figure 12.

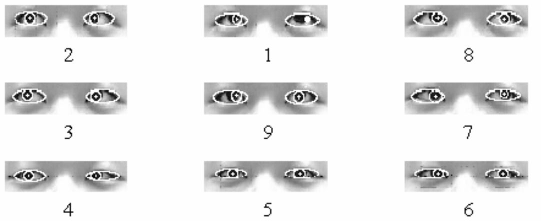


Figure 12: Eye modeling: for gaze directions mapped onto Figure 11.

This eye modeling algorithm is able to model the eye, iris and the pupil with high accuracy. However, it has some difficulties in detecting the iris boundaries, especially for the frontal gaze cases, in some cases the algorithm detects the two iris boundaries being too close to each other. This is due to the inherent noise in the image. Therefore, the authors propose several refinements to the eye modeling algorithm discussed above.

4. Eye modeling: Refined algorithm

4.1 Bounded Profile Refinement

A bounded profile calculation allows for correcting the boundary of the iris by recalculating the horizontal projection function. In this modified algorithm, the analysis of the horizontal profile is constrained by upper, lower, left and right bounds of the eye instead of the whole eye block. With the new localized profile, the analysis method is the same as in the previous algorithm. This modified algorithm fixes cases of frontal, that is direct gazes but marks out the left/right gaze cases.

4.2 Smoothening Refinement

Next, smoothening of the projection function is performed as follows: Right eye: profile function is smoothened by means of averaging it in a window sliding to the left of the pupil until there is only one maximum left. Identically smoothening of the right eye inner portion of the sclera is performed by sliding the window to the right of the pupil until there is only one minimum left. Alternatively, smoothing can be performed by binning/rough quantization of profile function

The idea is to smoothen out the noise until there is only one peak (maximum) or valley (minimum) exists. Then, while traversing the pupil from the inner eye corner (adjacent to the nose), one can find two extremes corresponding to iris boundaries providing a refined radius of the circle. Finally, the pupil location is readjusted to be the central point between two extrema. One can see that only adjacent to nose portions are smoothened. This is due to possible shadows cast by the nose.

In the original algorithm, the center of the iris is set as the pupil coordinate. Further refinement based on the above two techniques is the refinement of the pupil position by setting it at the minimum on the horizontal profile bounded by the iris boundaries.

The results of applying the all above improvements are shown in Figure 13.

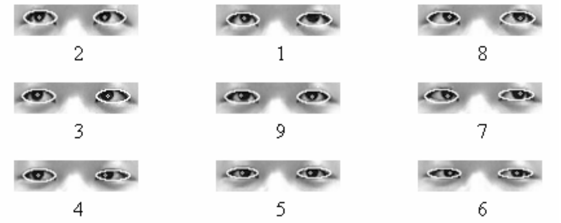


Figure 13: Refinement based on Bounded Profile, Smoothening and Pupil Refinement.

5. Gaze Differentiation

The problem of gaze differentiation is to map the position of the pupil (point) within the eye (ellipse) to the region of view on the screen. In general, for exact mapping of the pupil position to the region of view on the screen, one needs a perfect estimate of the face orientation in 3D. Due to limitations of computer vision techniques, imperfection of images, illumination and shadows, this deems to be a complex task. The proposed gaze differentiation method employs a two dimensional geometric mapping approach. Our approach also assumes that the user's face is parallel to the viewing screen plane (i.e. rotation of the user's head is insignificant) and the user's head tilt is also insignificant. It also assumes that the user is in viewing distance from the screen (≈33-36 cm).

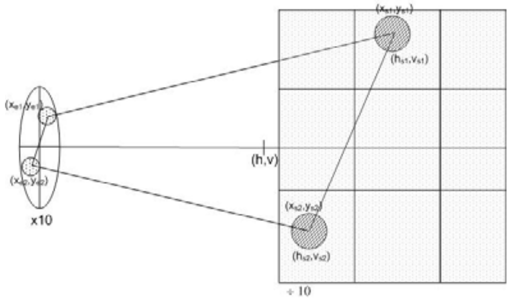


Figure 14: Mapping of eye movement corresponding to the displacement of point of gaze.

The proposed gaze differentiation process is divided into two phases 1) *Calibration phase* and 2) *Evaluation phase*. The calibration phase determines the displacement of the pupil (point) when the center of region of view is shifted from position U to position V, with respect to a reference point on the screen (h,v). Figure 14, shows the relation between the pupil displacement and the displacement in the region of view. The coordinates (x_{e1}, y_{e1}) and (x_{e2}, y_{e2}) correspond to the distance as seen by the camera and coordinates (x_{s1}, y_{s1}) and (x_{s2}, y_{s2}) correspond to physical screen distances. Experimental results show that when the eyes are displaced 16 pixels (as seen by the camera) is equivalent to physical eye displacement of 2.5 cms in the X-axis and 14.7 pixels (as seen by the camera) is equivalent to physical eye displacement of 1.7 cms in the Y-axis. These results are obtained when the camera is placed 33 cms from the user and the resolution of the captured image is 640 x 480.

The evaluation phase determines the relative position of the pupil with respect to reference point (h,v) and calculates the point of gaze (x and y coordinates) on the screen based on the data collected during the calibration phase. The gaze differentiation algorithm accuracy is determined by evaluating the algorithm's ability to identify the gaze direction among the nine regions of view.

6. Mean Shift Based Eye Tracking

Neural network is a time consuming process of comparing the current face image with a large database of images. Therefore instead of invoking neural network on every image frame to re-estimate the position of the eye centers, it is faster to track the eyes using some other methods once the initial eye locations are found. Mean shift tracking is an appearance-based object tracking method [Comaniciu and Meer, 2000]. It employs the analysis of the means of histogram distributions in a shifted position of a block/object under the search. A most similar appearance to the target model according to an adopted distance metrics will provide a target candidate region. In our case, an image block containing one eye is initialized to be our target model, areas around this target model is searched in the next image frame to determine where the eyes have moved to.

In order to compare distributions accurately, a window size and the search scheme is very important. The window must contain the most important features of the eye without including excessive information (i.e. eye brows and nose areas). The search area selection is also important, as it is used to calculate window distributions effectively. Experimental results suggest that window sizes of 20 x 30 and 30 x 30 have the best performance and are adopted in our eye tracking method.

The algorithm performs search in (*8xN + 1*) regions for the 320x240 resolution images, where *N* is the iteration number. This is done to search as many close regions to the original distribution as possible without searching too exhaustively. For example, if the number of iterations is set to 1, 9 regions will be searched as shown in Figure 15.

This decision is made by selecting one block within the searched area that has the most similar intensity distribution as the target model. Figure 16, shows an example of the mean shift tracking.

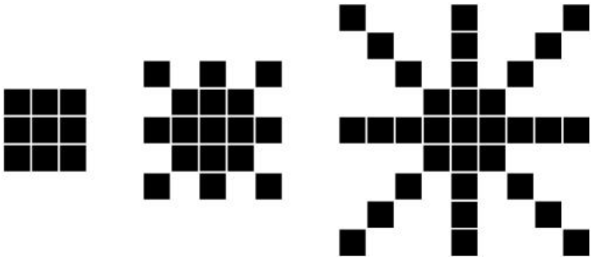


Figure 15: 1-iteration, 2-iteration and 4-iteration windowing scheme

This decision is made by selecting one block within the searched area that has the most similar intensity distribution as the target model, where the similarity is estimated based on the Bhattacharya distance metric [Fukunaga, 1990]. The optimum accuracy is obtained by running the window through four iterations. We base the accuracy on presence and absence of the eye in the current search window. If the eye is completely in the search window and roughly centered, we say it accurately found its target. Anything else is considered inaccurate. The window size selection has a crucial effect on the accuracy and the speed. In our experiments a window of 20x30 pixels yielded highest accuracies in 4 iterations.

Figure 16, shows an example of the mean shift tracking in which one can see that after 87 frames tracking a person with quick move left-right-up and down the algorithm continues perform quite accurate.

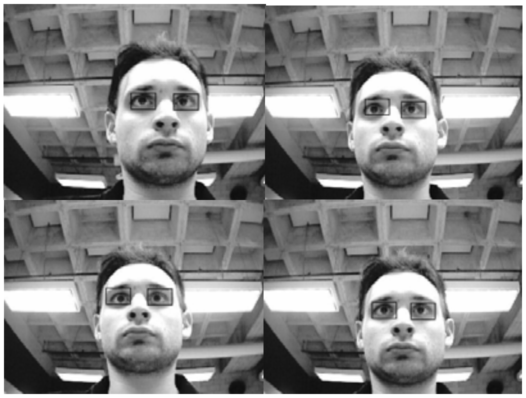


Figure 16: left → right→up→down moves of head and their tracking (box) through frames 15, 35, 68 and 87.

7. Results and Conclusion

In this paper, a robust non-intrusive eye detection and tracking system for gaze differentiation is presented. Our method combines the appearance based (neural network) and feature based (eye block processing) techniques. It is a three step process, in which first neural network is employed to determine the initial position of the face and the eyes. In the next step, the eye block identified by neural network is refined and the eye, iris, and pupil are detected and modeled as an ellipse, circle and dot respectively. In step 2, the region of view (gaze direction) is determined based on the relative position of the pupil whose model has been mapped onto the computer screen. In step 3, eye tracking for subsequent frames are tracked using mean shift algorithm. If the mean-shift eye tracking algorithm fails, Neural Network is invoked again to re-estimate the position of the face and the eyes. Some examples of our system outputs are shown in Figure 17 and Figure 18.

The developed Eye tracking and Gaze differentiation system consists of the following components: a) Progressive Web Camera (320 lines resolution), b) Video or Image acquisition board and c) Eye detection, tracking and gaze differentiation software. In the experimental setup, the camera is placed at the bottom-center of the computer/game screen and is pointed towards the user, such that the center of image contains the user's face. The experimental setup also assumes that the user is at 33~36 cms from the computer/game screen, which is the normal viewing distance for a computer/game screen and the camera is at 33 cms from the user. All experiments were conducted under standard office illumination conditions and variations in illumination were imposed by augmenting additional light sources. This augmentation of light source was required to evaluate the performance of our method for various applications that may illuminate the user's face depending on the content of the screen. The eye tracking performance of the proposed method was evaluated under various illuminations, various window sizes and iterations during eye tracking using the mean shift tracking algorithm. The gaze direction performance of the proposed method was evaluated by dividing the region of view on the computer screen into nine regions and evaluating the gaze direction to one of the nine regions of view. Experiments were conducted with users having different skin colors, eye shapes, inter-eye distance, etc. The total number of datasets collected was of 5 individuals looking at 9 different positions and individuals looking into and away from the screen. For the purpose of calibration and evaluation of accuracy of gaze direction, a screen consisting of 5cm x 5cm grids was placed behind the user during the calibration process (Figure 19).

Eye modeling has been tested on seven sets of grayscale CCD images, two sets of infrared images, and two sets of webcam (Panasonic Progressive Scan) grayscale images. Each set has nine images of the same person looking at different directions. Due to the considerable noise inherent in the CCD images, the modeling could not achieve the desired accuracy. The infrared imagery shows better modeling accuracy. The reason is that the IR illumination makes the projection function profile much smoother, hence the probability of taking erroneous local minima and maxima is reduced. However, infrared images are of lower contrast, that makes further analysis based on thresholding for eye outline detection weak. Therefore, a progressive scan web camera which has a lower noise than the CCD camera but sharper contrast than the IR camera was determined as the most suitable camera for the system.

The performance of our method was evaluated for the following conditions:

1. Various cameras (CCD, IR CCD and progressive scan webcam) with resolutions ranging from 280 lines to 480 lines.
2. Two and nine gaze differentiations of the screen as shown in Figure 11.
3. Time, cost of the system, accuracy and resolution of our approach was compared to other gaze differentiation methods found in literature.

Two and Nine Gaze differentiations:

The proposed eye tracking system determines a two region and nine region gaze differentiation. This scenario was selected to apply our approach for Human-Computer Interactive application for gaming machines.. In the two region gaze differentiation, the method is able to differentiate if the user is looking into the screen or away from the screen (Figure 17). The accuracy obtained of two gaze differentiation is 100%. In the nine gaze differentiation, the system detects the center of region of view as one among the nine regions in the screen (Figure 11) as shown in Figure 18. The system evaluates the position of the eye pupil and the position of center of region of view. Calculation from experimental results also shows that our method can differentiate gaze in the range of 1.5 mm.

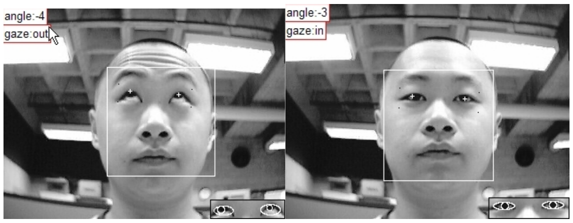


Figure 17: Two gaze differentiation

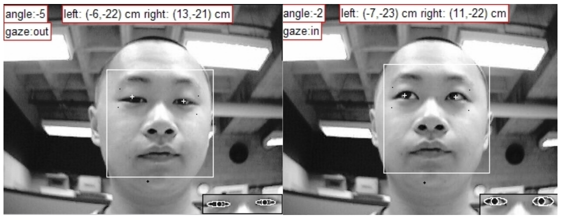

angle:-5
left: (-6,-22) cm right: (13,-21) cm
gaze:out
angle:-2
left: (-7,-23) cm right: (11,-22) cm
gaze:in

Figure 18: Nine gaze differentiation

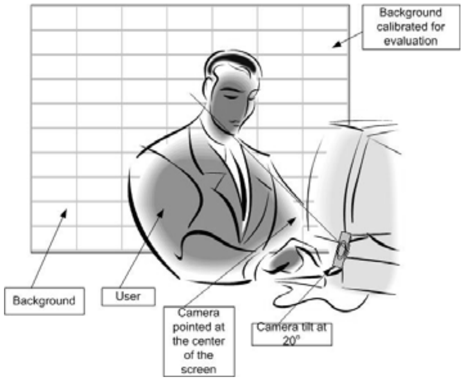

Background calibrated for evaluation
Background
User
Camera pointed at the center of the screen
Camera tilt at 20°

Figure 19: Experimental setup for evaluating accuracy of the eye tracking system

Execution Time:

Table 2, shows the execution times of each involved.step The total worst case execution time for detecting, modeling and tracking eyes is 0.5 sec (using NN eye tracking, eye modeling and gaze differentiation). However, the time taken by subsequent image frames is 0.238 Sec (using mean-shift eye tracking, modeling and gaze differentiation).

Table 2: Execution time of eye tracking and gaze differentiation

ALGORITHM	TIME/FRAME (ms)
Neural Network for face detection	400
Pupil and Eye detection, Eye modeling and Gaze Differentiation	100
Eye tracking using Mean Shift Method	138

The software implementation of our proposed algorithm was developed using Matlab. The overall accuracy of gaze differentiation is 94.75% for the collected dataset; assuming normal office lighting condition and frontal with respect to the camera face position, The face rotation and tilt angle is restrained within 20 degrees. The accuracy of gaze detection could be improved with higher resolution images, but at the expense of longer processing times.

Table 3: Comparison of the proposed system

	Our Method	Pupil Tracking*	ANN*
Face Access	Good	Good	Good
Subject Contact	No	No	No
Accuracy	H=2.71°, V=2.09°	0.003°	1.5°
Resolution	Good	Good	Good
Range	±40°	±20-40°	±5°
Sampling Rate	30 Hz	50 - 250 Hz	15 Hz
Real-time	Yes	Yes	No
Rotation	X/Y	X/Y	X/Y
Price	$1000	$10K ~ $45K	-

* [Duchowski, 2002]

Table 3, compares the proposed gaze differentiation approach with other approaches in the literature. The performance characteristics taken for comparison include face access, type of subject contact, accuracy (percentage of the error in gaze differentiation), resolution (ability to detect the smallest shift in eye gaze), angular range the system is able to differentiate the gaze, ability of operating in real-time, and work under head rotation; and the cost of the system. The main advantage of the proposed approach is that a complete system is developed that involves both eye detection, tracking and gaze differentiation.

Reference

[Ji and Yang, 2001] Q. Ji, X. Yang, Real time visual cues extraction for monitoring driver vigilance, in: Proc. of International Workshop on Computer Vision Systems, Vancouver, Canada, 2001.

[Ji and Zhu, 2002] Q. Ji, Z. Zhu, Eye and gaze tracking for interactive graphic display, in: 2nd International Symposium on Smart Graphics, Hawthorne, NY, USA, 2002.

[Pentland et. al., 1994] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, in: Proc. of IEEE Conf. on Computer Vision and Pattern Recognition (CVPR'94), Seattle, WA, 1994.

[Huang and Wechsler, 1999] J. Huang, H. Wechsler, Eye detection using optimal wavelet packets and radial basis functions (RBFS), International Journal of Pattern recognition and Artificial Intelligence 13 (7), 1009-1025, 1999.

[Kawato and Ohya, 2000] S. Kawato, J. Ohya, Two-step approach for real-time eye tracking with a new filtering technique, in: Proc. Int. Conf. on System, Man & Cybernetics, pp.1366-1371, 2000.

[Sirohey and Rosenfeld, 2001] S. A. Sirohey, A. Rosenfeld, Eye detection in a face image using linear and nonlinear filters, Pattern recognition (341), pp. 367-1391, 2001.

[Rowley et. al., 1998] H. A. Rowley, S. Baluja, and T. Kanade, Neural network-based face detection. IEEE Transactions on Pattern Analysis and Machine Intelligence, 20(1), January 1998.

[Niemistö, 2004] Antti Niemistö (2004) HistThresh toolbox for MATLAB.
<http://www.cs.tut.fi/~ant/histthresh/ThreshComp.pdf>

[Feng and Yuen, 1998] G. C. Feng, P. C. Yuen, Variance projection function and its application to eye detection for human face recognition, International Journal of Computer Vision Vol. 19, pp. 899-906, 1998.

[Kapur et. al. 1985] J. N. Kapur, P. K. Sahoo, and A. K. C. Wong, A new method for gray-level picture thresholding using the entropy of the histogram, Computer Vision Graphics Image Process., Vol. 29, pp. 273-285, 1985.

[Fukunaga, 1990] K. Fukunaga, Introduction to Statistical Pattern Recognition, second Ed. Academic Press, 1990.

[Otsu, 1979] N. Otsu, A threshold selection method from gray-level histogram, IEEE Trans. Systems Man Cybernet., vol. 9, pp. 62-66, 1979.

[Comaniciu and Meer, 2000] D. Comaniciu, D. and P. Meer, Real-Time Tracking of Non-Rigid Objects Using Mean Shift, Proc. IEEE Conf. Computer Vision and pattern Recognition, Vol., pp. 142-149, June 2000.

[Duchowski, 2002] A.T. Duchowski, A breadth-first survey of eye-tracking applications. Behavior Research Methods, Instruments, & Computers, Volume 34, Number 4, pp. 455-470(16), 2002.

[Anders, 2001] G. Anders, Pilot's Attention Allocation During Approach and Landing–Eye- and Head-Tracking Research in an A330 Full Flight Simulator. In International Symposium on Aviation Psychology (ISAP). Columbus, OH, 2001.

[Robinson, 1968] D.A. Robinson, The Oculomotor Control System: A Review. Proceedings of the IEEE, 56(6), pp. 1032-1049, 1968.

[Rayner, 1998] K. Rayner, Eye Movements in Reading and Information Processing: 20 Years of Research. Psychological Bulletin, 124(3), pp. 372-422, 1998.

[Allopenna et. al., 1998] .D. Allopenna, J.S. Magnuson, M.K. Tanenhaus, Tracking the Time Course of Spoken Word Recognition Using Eye Movements: Evidence for Continuous Mapping Models. Journal of Memory and Language, 38(4), pp. 419-439, 1998.

[Vertegaal, 1999] R. Vertegaal, The GAZE Groupware System: Mediating Joint Attention in Mutiparty Communication and Collaboration. In Human Factors in Computing Systems: CHI '99 Conference Proceedings pp. 294-301, 1999.

Emotion & Reinforcement: Affective Facial Expressions Facilitate Robot Learning

Joost Broekens[1], Pascal Haazebroek[2]
[1]Leiden Institute of Advanced Computer Science (LIACS), Leiden University, Leiden, The Netherlands.
[2]Cognitive Psychology Unit, Leiden University, Leiden, The Netherlands
broekens@liacs.nl, phaazebroek@fsw.leidenuniv.nl

Abstract

Computer models can be used to investigate the role of emotion in learning. Here we present *EARL*, our framework for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning (RL). EARL enables the study of, among other things, communicated affect as reinforcement to the robot; the focus of this paper. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. Emotional expression can therefore be a reinforcement signal to a child. We hypothesize that affective facial expressions facilitate robot learning, and compare a *social* setting with a *non-social* one to test this. The non-social setting consists of a simulated robot that learns to solve a typical RL task in a continuous grid-world environment. The social setting additionally consists of a human (parent) observing the simulated robot (child). The human's emotional expressions are analyzed in real time and converted to an additional reinforcement signal used by the robot; positive expressions result in reward, negative expressions in punishment. We quantitatively show that the "social robot" indeed learns to solve its task significantly faster than its "non-social sibling". We conclude that this presents strong evidence for the potential benefit of affective communication with humans in the reinforcement learning loop.

1 Introduction

In humans, emotion influences thought and behavior in many ways (Damasio, 1994; Rolls, 1999; Custers & Aarts, 2005; Dreisbach & Goschke, 2004). For example, emotion influences how humans process information by controlling the broadness versus the narrowness of attention. Also, emotion functions as a social signal that communicates reinforcement of behavior in, e.g., parent-child relations. Computational modeling (including robot modeling) has proven to be a viable method of investigating the relation between emotion and learning (Broekens, Kosters & Verbeek, 2007) Gandanho, 2003), emotion and problem solving (Belavkin, 2004; Bothello & Coehlo, 1998), emotion and social robots (Breazeal, 2001; for review see Fong, Nourbakhsh & Dautenhahn, 2003), and emotion, motivation and behavior selection (Avila-Garcia & Cañamero, 2004; Blanchard and Cañamero, 2006; Cos-Aguilera et al., 2005; Velasquez, 1998). Although many approaches exist and much work has been done on computational modeling of emotional influences on thought and behavior, none explicitly targets the study of the relation between emotion and learning using a complete end-to-end framework in a reinforcement learning context[1]. By this we mean a framework that enables systematic *quantitative* study of the relation between affect and RL in a large variety of ways, including (a) affect as reinforcement to the robot (both internally generated as well as socially communicated), (b) affect as perceptual feature to the robot (again internally generated and social), (c) affect resulting from reinforced robot behavior, and (d) affect as meta-parameters for the robot's learning mechanism. In this paper we present such a framework. We call our framework *EARL*, short for the systematic study of the relation between *e*motion, *a*daptation and *r*einforcement *l*earning.

In this paper we specifically focus on the influence of socially communicated emotion on learning in a reinforcement learning context. We show, using our framework *EARL*, that human emotional expressions can be used as additional reinforcement signal used by a simulated robot.

The robot's task is to optimize food-finding behavior while navigating through a continuous grid world environment. The grid world is not discrete, nor is an attempt made to define discrete states based on the continuous input. The gridworld contains walls, path and food patches. The robot perceives its direct surroundings as they are. We have developed an action-based learning mechanisms that learns to predict values of actions based on the current perception of the agent (note that in this paper we use the terms agent and robot interchangeably). Every action has its own Multi-Layer Percepton network (see also, Lin, 1993) that learns to predict a modified version of the *Q*-value (Sutton & Barto, 1998). We have used this setup such that observed robot

[1] Although the work by Gandanho (2003) is a partial exception as it explicitly addresses emotion in the context of RL. However, this work does not address social human input and social robot output.

behavior can be extrapolated to the real world; building the actual robot with appropriate sensors and actuators would, in theory, suffice to replicate the results. We explain our modeling method in more detail in Section 5.

As mentioned above, we study the effect of a human's emotional expression on the learning behavior of the robot. In humans, emotions are crucial to learning. For example, a parent—observing a child—uses emotional expression to encourage or discourage specific behaviors. In this case, the emotional expression is used to setup an *affective communication channel* (Picard, 1997) and is used to communicate a reinforcement signal to a child. In this paper we take *affect* to mean the positiveness versus the negativeness of a situation, object, etc. (see Rolls, 1999; Russell, 2003; and Broekens, Kosters & Verbeek, 2007 for a more detailed argumentation of this point of view). The human observes the simulated robot while it learns to find food, and affect in the human's facial expression is recognized by the robot in real time. As such a smile is interpreted as communicating positive affect and therefore converted to a small additional reward (additional to the reinforcement the robot receives from its simulated environment). The expression of fear is interpreted as communicating negative affect and therefore converted to a small additional punishment. We call this the *social* setting. The *non-social* setting is the same, emotional expression generating additional reinforcement apart. That is, the non-social setting is a standard experimental reinforcement learning setup.

We hypothesized that robot learning (in a RL context as described above) is facilitated by additional social reinforcement. Our experimental results support this hypothesis. We compared the learning performance of our simulated robot in the social and non-social settings, by analyzing averages of learning curves. The main contribution of this research is that it presents *quantitative* evidence of the fact that a human-in-the-loop can boost learning performance in real-time, in a non-trivial learning environment. We belief this is an important result. It provides a solid base for further study of human mediated robot-learning in the context of real-world applicable reinforcement learning, using the communication protocol nature has provide for that purpose, i.e., emotional expression and recognition. As such, our results suggest that robots can be trained and their behaviors optimized using natural social cues. This facilitates human-robot interaction.

The rest of this paper is structured as follows. In Section 2 we explain in some more detail our view of affect, emotion and how affect influences learning in humans. In Section 3 we briefly introduce *EARL*, our complete framework. In Section 4 we describe how communicated affect is linked to a social reinforcement signal. In Section 5, we explain our method of study (e.g., the grid-world, the learning mechanism). Section 6 discusses the results and Section 7 discusses these in a broader context and presents concluding remarks and future work.

2 Affect Influences Learning

In this paper we specifically focus on the influence of socially communicated affect on learning. Affect and emotion are concepts that lack a single concise definition, instead there are many (Picard et al., 2004). Therefore we first explain our meaning to these concepts. In general, the term emotion refers to a set of—in social animals—naturally occurring phenomena including facial expression, motivation, emotional actions such as fight or flight behavior, a tendency to act, and—at least in humans—feelings and cognitive appraisal (see, e.g., Scherer, 2001). An emotional state is the combined activation of instances of a subset of these phenomena, e.g., angry involves a tendency to fight, a typical facial expression, a typical negative feeling, etc. Time is another important aspect in this context. A short term (intense, object directed) emotional state is often called an *emotion*; while a longer term (less intense, non-object directed) emotional state is referred to as *mood*. The direction of the emotional state, either positive or negative, is referred to as *affect* (e.g., Russell, 2003). Affect is often differentiated into two orthogonal (independent) variables: *valence*, a.k.a. pleasure, and *arousal* (Dreisback & Goschke, 2004; Russell, 2003). Valence refers to the positive versus negative aspect of an emotional state. Arousal refers to the activity of the organism during that state, i.e., physical readiness. For example, a car that passes you in a dangerous manner on the freeway, immediately (*time*) elicits a strongly negative and highly arousing (*affect*) emotional state that includes the expression of anger and fear, feelings of anger and fear, and intense cognitive appraisal about what could have gone wrong. On the contrary, learning that one has missed the opportunity to meet an old friend involves cognitive appraisal that can negatively influence (*affect*) a person's mood for a whole day (*time*), even though the associated emotion is not necessarily arousing (*affect*). Eating a piece of pie is a more positive and biochemical example. This is a bodily, emotion-eliciting event resulting in mid-term moderately-positive affect. Eating pie can make a person happy by, e.g., triggering fatty-substance and sugar-receptor cells in the mouth. The resulting positive feeling typically is not of particularly strong intensity and certainly does not involve particularly high or low arousal, but might last for several hours.

Emotion influences thought and behavior in many ways. For example, at the neurological level, malfunction of certain brain areas not only destroys or diminishes the capacity to have (or express) certain emotions, but also has a similar effect on the capacity to make sound decisions (Damasio, 1994) as well as on the capacity to learn new behavior (Berridge, 2003). Behavioral evidence suggests that the ability to have sensations of pleasure and pain is strongly connected to basic mechanisms of learning and decision-making (Berridge, 2003; Cohen & Blum, 2002). These findings indicate that brain areas important for emotions are also important for "classical" cognition and instrumental learning.

At the level of cognition, a person's belief about something is updated according to the emotion: the current emotion is used as information about the perceived object (Clore

& Gasper, 2000; Forgas, 2000), and emotion is used to make the belief resistant to change (Frijda & Mesquita, 2000). Ergo, emotions are “at the heart of what beliefs are about” (Frijda et al., 2000).

Emotions play a role in the regulation of the amount of information processing. For instance, Scherer (2001) argues that emotion is related to the continuous checking of the environment for important stimuli. More resources are allocated to further evaluate the implications of an event, only if the stimulus appears important enough. Furthermore, in the work of Forgas (2000) the relation between emotion and information processing strategy is made explicit: the influence of mood on thinking depends on the strategy used. In addition to this, it has been found that positive moods favor creative thoughts as well as integrative information processing, while negative moods favor systematic analysis of incoming stimuli (e.g. Ashby, Isen & Turken, 1999; Gasper & Clore, 2002).

Emotion also regulates behavior of others. Obvious in human development, expression (and subsequent recognition) of emotion is important to communicate (dis)approval of the actions of others. This is typically important in parent-child relations. Parents use emotional expression to guide behavior of infants. Emotional interaction is essential for learning. Striking examples are children with an autistic spectrum disorder, typically characterized by a restricted repertoire of behaviors and interests, as well as social and communicative impairments such as difficulty in joint attention, difficulty recognizing and expressing emotion, and lacking of a social smile (for review see Charman & Baird, 2002). Apparently, children suffering from this disorder have both a difficulty in building up a large set of complex behaviors *and* a difficulty understanding emotional expressions and giving the correct social responses to these. This disorder provides a clear example of the interplay between learning behaviors and being able to process emotional cues.

To summarize, emotion and mood influence thought and behavior in a variety of ways, e.g., a persons mood influences processing style and attention, emotions influences how one thinks about objects, situations and persons, and emotion is related to learning behaviors.

In this study we focus on the role of affect in guiding learning in a social human-robot setting. We use affect to denote the positiveness versus negativeness of a situation. We ignore the arousal a certain situation might bring. As such, positive affect characterizes a situation as good, while negative affect characterizes that situation as bad (e.g., Russell, 2003). Further, we use affect to refer to the *short term* timescale: i.e., to emotion. We hypothesize that affect communicated by a human observer can enhance robot learning. In our study we assume that the recognition of affect translates into a reinforcement signal. As such, the robot uses a *social reinforcement* in addition to the reinforcement it receives from its environment while it is building a model of the environment using reinforcement learning mechanisms. In the following sections we first explain our framework after which we detail our method and discuss results and further work.

3 *EARL*: A Computational Framework to Study the Relation between Emotion, Adaptation and Reinforcement Learning.

To study the relation between emotion, adaptation and reinforcement learning, we have developed an end-to-end framework. The framework consists of four parts:

- An emotion recognition module, recognizing emotional facial expression in real time.
- A reinforcement learning agent to which the recognized emotion can be fed as input.
- An artificial emotion module slot, this slot can be used to plug in different models of emotion into the learning agent that produce the artificial emotion of the agent as output. The modules can use all of the information that is available to the agent (such as action repertoire, reward history, etc.). This emotion can be used by the agent as intrinsic reward, as metalearning parameter, or as input for the expression module.
- An expression module, consisting of a robot head with the following degrees of freedom: eyes moving up and down, ears moving up and down on the outside, lips moving up and down, eyelids moving up and down on the outside, and RGB eye colors

Emotion recognition is based on quite a crude mechanism based upon the face tracking abilities of OpenCV (http://www.intel.com/technology/computing/opencv/index.htm). It uses 9 points on the face each defined by a blue sticker: 1 on the tip of the nose, 2 above each eyebrow, 1 at each mouth corner and 1 on the upper and lower lip. The recognition module is configured to store multiple prototype point constellations. The user is prompted to express a certain emotion and press space while doing so. For every emotional expression (in the case of our experiment neutral, happy and afraid), the module records the positions of the 9 points relative to the nose. This is a prototype point vector. After configuration, to determine the current emotional expression in real time the module calculates a weighted distance from the current point vector (read in real-time from a web-cam mounted on the computer screen) to the prototype vectors. Different points get different weights. This results in an error measure for every prototype expression. This error measure is the basis for a normalized vector of recognized emotion intensities. The recognition module sends this vector to the agent (i.e., neutral 0.3, happy 0.6, fear 0.1). Our choice of weights and features has been inspired by work of others (for review see Pantic & Rothkrantz, 2000). Of course the state of the art in emotion recognition is more advanced than our current approach. However, as our focus is affective learning and not the recognition process per se, we contented ourselves with a low fidelity solution (working almost perfectly for neutral, happy and afraid, when the user keeps the head in about the same position).

Note that we do not aim at generically recognizing emotional expressions. Instead, we tune the recognition module to the individual observer to accommodate his/her personal and natural facial expressions.

The reinforcement learning agent receives this recognized emotion and can use this in multiple ways: as reward, as information (additional state input), as metaparameter (e.g., to control learning rate), and as social input directly into its emotion model. In this paper we focus on social reinforcement, and as such focus on the recognized emotion being used as additional reward or punishment. The agent, its learning mechanism and how it uses the recognized emotion as reinforcement are detailed in Sections 4 and 5.

The artificial emotion model slot enables us to plug in different emotion models based on different theories to study their behavior in the context of reinforcement learning. For example, we have developed a model based on the theory by Rolls (1999), who argues that many emotions can be related to reward and punishment and the lack thereof. This model enables us to see if the agent's situation results in a plausible (e.g., scored by a set of human observers) emotion emerging from the model. By scoring the plausibility of the resulting emotion, we can learn about the compatibility of, e.g., Rolls' emotion theory with reinforcement learning. However, in the current study we have not used this module, as we focus on affective input as social reward.

The emotion expression part is a physical robot head. The head can express an arbitrary emotion by mapping it to its facial features, again according to a certain theory. Currently our head expresses emotions according to the Pleasure Arousal Dominance (PAD) model by Mehrabian (1980). We have a continuous mapping from the 3-dimensional PAD space to the features of the robot face. As such we do not need to explicitly work with emotional categories or intensities of the categories. The mapping appears to work quite well, but is in need of validation study (again using human observers). We have not used the robot head for the studies reported upon in this paper.

We now describe in detail how we coupled the recognized human emotion to the social reinforcement signal for the robot. Then we explain in detail our adapted reinforcement learning mechanism (such that it enabled learning in continuous environments), and our method of study as well as our results.

4 Emotional Expressions as Reinforcement Signal.

As mentioned earlier, emotional expressions and facial expressions in particular can be used as social cues for the desirability of a certain action. In other words, an emotional expression can express reward and punishment if directed at an individual. We focus on communicated affect, i.e., the positiveness versus negativeness of the expression. If the human expresses a smile (happy face) this is interpreted as positive affect. If the human expresses fear, this is interpreted as negative affect. We interpret a neutral face as affectless.

We have studied the mechanism of communicated affective feedback in a human-robot interaction setup. The human's face is analyzed (as explained above) and a vector of emotional expression intensities is fed to the learning agent. The agent takes the expression with the highest intensity as dominant, and equates this with a *social reward* of, e.g., 2 (happy), −2 (fear) and 0 (neutral). This is obviously a simplified setup, as the human face communicates much more subtle affective messages and at the very least is able to communicate the degree of reward and punishment. However, to investigate our hypothesis (affective human feedback increases robot learning performance), the just described mechanism is sufficient.

The social reward is simply added to the "normal" reward the agent receives from the environment. So, if the agent walks on a path somewhere in the gridworld, it receives a reward (say 0), but when the user smiles, the resulting actual reward becomes 2, while if the user looks afraid, the resulting reward becomes −2. Additionally, the agent learns (in a way describe in the next Section) to associate its perception with that social reward. So, in RL terms, it builds up a "social reward function". The user expresses emotions during a short time period, after which the learned social reward function takes over. By doing so we were able to study the impact on robot learning of two phenomena: direct social reinforcement and learned social reinforcement.

5 Method

To study the impact of social reinforcement on robot learning, we have used our framework in the following experimental setup.

A simulated robot (agent) "lives" in a continuous gridworld environment consisting of wall, food and path patches (Figure 1). These are the features of the world observable by the agent. The agent cannot walk on walls, but can walk on path and food. Walls and path are neutral (have a reinforcement of 0.0), while food has a reinforcement of 10. One cell in the grid is assumed to be a 20 by 20 object. Even though wall, path and food are placed on a grid, the world is continuous in the following sense: the agent moves by turning or walking in a certain direction using an arbitrary speed (in our experiments set at 3), and perceives its direct surroundings (within a radius of 20) according to its looking direction (one out of 16 possible directions). The agent uses a "relative eight neighbor metric" meaning that it perceives features of the world at 8 points around it, with each point at a distance of 20 from the center point of the agent and each point at an interval of 1/4 PI radians, with the first point always being exactly in front of it (Figure 1). The state perceived by the agent (its percept) is a real-valued vector of inputs between 0 and 1; each input is defined by the relative contribution of a certain feature in the agent-relative direction corresponding to the input. For example, if the agent sees a wall just in front of it (i.e., the center point of a wall object is exactly at a distance of 20 as measured from the current agent location in its looking direction) the first value in its perceived state would be equal to 1. This value can be anywhere between 0 and 1 depending on the distance of that point to the feature. For the three types of features, the agent thus has 3x8=24 real-valued inputs between 0 and 1 as its perceived world state *s* (Figure 1). As such the agent can approach objects (e.g., a wall) from a large number of possible angles and positions, with every intermediate position being possible. For all practical purposes, the learning environment can be considered continuous. States are not discretize to facilitate learning. Instead we chose to use the perceived state as is, to maximize compatibility of our experimental results with real-world robots. However, reinforcement learning in continuous environments introduces several important problems for standard RL techniques, such as Q learning, mainly because a large number of potentially similar states exist as well as a very long path length between start and goal states making value propagation difficult. We now briefly explain our adapted RL mechanism. As RL in continuous environments is not specifically the topic of the paper we have left out some of the rational for our choices.

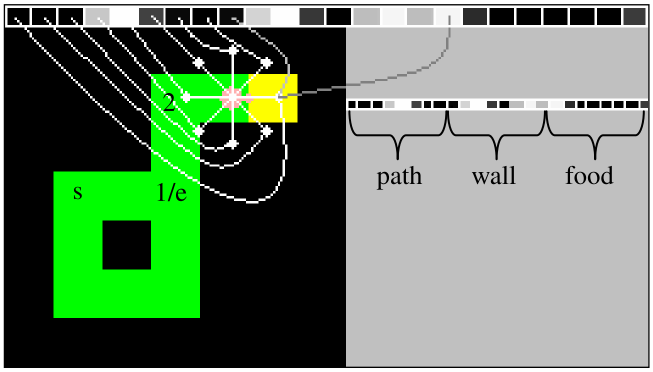


Figure 1. The experimental gridworld. The agent is the "circle with nose" in the top right of the maze, where the nose denotes its direction. The 8 white dots denote the points perceived by the agent. These points are connected to the elements of state *s* (neural input to the MLPs used by the agent) as depicted. This is repeated for all possible features, in our case: path (gray), wall (black), and food (light gray), in that order. The "e" denotes the cell in which social reward can be administered through smiling or expression of fear, the "1" and "2" denote key locations at which the agent has to learn to differentiate its behavior, i.e., either turn left ("1") or right ("2"). The agent starts at "s". The task enforces a non-reactive best solution (by which we mean that there is no direct mapping from reward to action that enables the agent to find the shortest path to the food). If the agent would learn that turning right is good, it would keep walking in circles. If the agent learns that turning left is good, it would not get to the food.

The agent learns to find the path to the food, and optimizes this path. At every step the agent takes, the agent updates its model of the expected benefit of a certain action as follows. It learns to predict the value of actions in a certain perceived state *s*, using an adapted form of *Q* learning. The value function, $Q_a(s)$, is approximated using a multilayer perceptron (MLP), with 3x8=24 input, 24 hidden, and one output neuron(s), with *s* being the real-valued input to the MLP, *a* the action to which the network belongs, and the output neuron converging to $Q_a(s)$. As such, every action of the agent (5 in total: forward, left, right, left and forward, right and forward) has its own network. The output of the action networks are used as action values in a standard Boltzmann action-selection function (Sutton & Barto, 1998). An action network is trained on the *Q* value—i.e., $Q_a(\mathrm{s}) \leftarrow Q_a(s) + \alpha(r+\gamma Q(s') - Q_a(s))$ —where *r* is the reward resulting from action *a* in state *s*, *s'* is the resulting next state, $Q(s')$ the value of state *s'*, α is the learning rate and γ the discount factor (Sutton & Barto, 1998). The learning rate equals 1 in our experiments (because the learning rate of the MLP is used to control speed of learning, not α), and the discount factor equals 0.99. To cope with a continuous gridworld, we adapted standard *Q* learning in the following way:

First, the value $Q_a(s)$ used to train the MLP network for action *a* is topped such that $min(r, Q_a(s')) <= Q_a(s) <= max(r, Q(s'))$. As a result, individual $Q_a(s)$ values can never be larger or smaller than any of the rewards encountered in the world. This enables a discount factor close to or equal to 1, needed to efficiently propagate back the food's reward through a long sequence of steps. In continuous, cyclic, worlds, training the MLP on normal *Q* values using a discount factor close to 1 can result in several problems not further discussed here.

Second, per step of the agent, we train the action-state networks not only on $Q_a(s) \leftarrow Q_a(s) + \alpha(r+\gamma Q(s') - Q_a(s))$ but also on $Q_a(s') \leftarrow Q_a(s')$. The latter seems unnecessary but is quite important. RL assumes that values are propagated *back*, but MLPs generalize while trained. As a result, training an MLP on $Q_a(s)$ also influences its value prediction for *s'* in the same direction, just because the inputs are very close. In effect, part of the value is actually propagated *forward*; credit is partly assigned to what comes next. This violates the RL assumption just mentioned. Note that the value $Q(s')$ is predicted using another MLP, called the value network, that is trained in the same way as the action networks using the topped-off value and forward propagation compensation.

Third, for the agent to better discriminate between situations that are perceptually similar, such as position "1" and "2" in Figure 1, for each action-network the agent also uses a second network trained on the value of *not* taking the action. This network is trained when other actions are taken but not when the action to which the "negation" network belongs is taken. In effect, the agent has two MLPs per action. This enables the agent to better learn that, e.g., "right" is good in situation "2" but *not* in situation "1". Without this "negation" network, the agent learns much less efficient (results not shown). To summarize, our agent has 5 actions, it has 11 MLPs in total: one to train $Q(s)$, 5 to train $Q_a(s)$ and 5 to train $-Q_a(s)$. All networks use forward propagation compensation and a topped-off value to train upon. The MLP predictions for $Q_a(s)$ and $-Q_a(s)$ are simply added, and the result is used for action-selection.

To study the effect of communicated affect as social reward, we created the following setup. First an agent is trained without social reward. The agent repeatedly tries to find the food for 200 trials, i.e., one *run*. The agent continuously learns and acts during these trials. To facilitate learning, we use a common method to vary the MLP learning rate and the Boltzmann action selection β derived from simulated annealing. The Boltzmann β equals to $3+(\mathit{trial}/200)*(6-3)$, effectively varying from 3 in the first trial to 6 in the last. The MLP learning rate equals to $0.1-(\mathit{trial}/200)*(0.1-0.001)$ effectively varying from 0.1 in the first trial to 0.001 in the last. We repeated the experiment 200 times, resulting in 200 runs. Average learning curves are plotted for these 200 runs using a linear smoothing factor equal to 6 (Figure 2).

Second, a new agent is trained *with* social reward, i.e., a human observer looking at the agent with his/her face analyzed by the agent, translating a smile to a positive social reward and a fearful expression to a negative social reward. Again, average learning curves are plotted using a linear smoothing factor equal to 6, but now based on the average per trial over 15 runs (Figure 2). We experimented with three different social settings: (a) social input from trial 20 to 30, where the social reward is either −0.5 or 0.5 (happy vs. fearful, respectively); (b) social input from trial 20 to 25 where social reward is either −2 or 2, i.e., more extreme social rewards but for a shorter period; (c) social input from trial 29 to 45 where social reward is either −2 or 2 while the agent trains an additional MLP to predict the social reward based on the current state s, so the MLP is trained to predict $R_{social}(s)$. After trial 45, the direct social reward from the observer is replaced by the learned social reward $R_{social}(s)$. As a result, the agent learns to predict what its human tutor thinks about certain situations.

The process of giving affective feedback to a reinforcement learning agent appeared to be quite a long, intensive and attention absorbing experience. As a result, it was physically impossible to observe the agent during all runs and all trials in the entire gridworld (after 2 hours of smiling to a computer screen one is completely fed-up with it *and* has burning eyes and painful facial muscles). To be able to test our hypothesis, we restricted social input to (a) a critical learning period defined in terms of a start and end trail (see above), and (b) the cell indicated by “e” (Figure 1). Only when the agent moves around in this cell and is in a social input trial, the simulation speed of the experiment is set to one action per second enabling affective feedback.

6 Results

The results clearly show that learning is facilitated by social reward. In all three social settings (Figure 2a, b and c) the agent needs fewer steps to find the food during the trials in which the observer provides assistance to the agent by expression positive or negative affect. Interestingly, at the moment the observer stops giving social rewards, the agent gradually looses the learning benefit it had accumulated. This is independent of the size of the social reward (both social learning curves in Figure 2a and b show dips that eventually return to the non-social learning curve). This can be easily explained. The social reward was not given long enough for the agent to internalize the path to the food (i.e., propagate back the food’s reward to the beginning of the path). As soon as the observer stops giving social rewards, the agent starts to forget these rewards, i.e., the MLPs are again trained to predict values as they are without social input. So, either the observer should continue to give social rewards until the agent has internalized the solution, or the agent needs to be able to build a representation of the social reward function and uses it when actual social reward is not available. We have experimented with the second (social setting *c*): we enabled the agent to learn the social reward function. Now the agent uses actual social reward at the emotional input spot (“e”, Figure 1) during the critical period, and uses its social reward prediction when social input stops. This is the third social setup. Results clearly show that the agent is now able to keep the benefit it had accumulated from using social rewards (Figure 2c). These results show that a combination of using social reward and learning a social reward function facilitates robot learning, by enabling the robot to quicker learn the optimal solution to the food due to the direct social reward as well as keep that solution by using its learned social reward function when social reward stops.

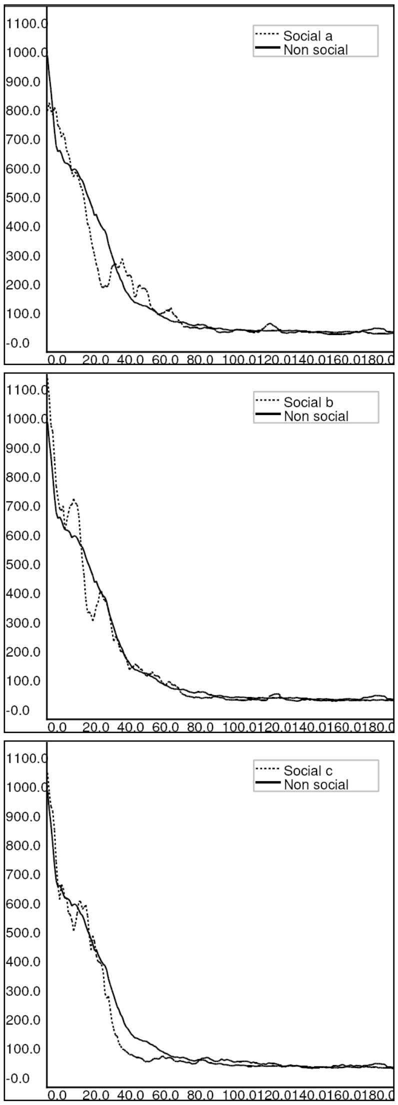


Figure 2. Results of the learning experiments. From top to bottom showing the difference between the non-social setting and social setting *a*, *b*, and *c* respectively.

7 Conclusion, Discussion and Further Work

Our results show that affective interaction in human-in-the-loop learning can provide significant benefit to the efficiency of a reinforcement learning robot in a continuous grid world. We believe our results are particularly important to human-robot interaction for the following reasons. First, advanced robots such as robot companions, robot workers, etc., will need to be able to adapt their behavior according to human feedback. For humans it is important to be able to give such feedback in a natural way, e.g., using emotional expression. Second, humans will not want to give feedback all the time, it is therefore important to be able to define critical learning periods as well as have an efficient social reward system. We have shown the feasibility of both. Social input during the critical learning periods was enough to show a learning benefit, and the relatively easy step of adding an MLP to learn the social reward function enabled the robot to use the social reward when the observer is away.

We have specifically used an experimental setup that is compatible with a real-world robot due: we have used continuous inputs and MLP-based training of which it is known that it can cope with noise and generalize over training examples. As such we believe our results can be generalized to real-world robotics. However, this most certainly needs to be experimented with.

Many interesting computational approaches exist that study emotion in the context of robots and agents, of which we mention one explicitly here as it is particularly related to our work: the adaptive, social chatter bot *Cobot* (Isbell et al., 2001). Cobot learns the information preferences of its chat partners, by analyzing the chat messages for explicit and implicit reward signals. These signals are then used to adapt its model of providing information to that chat partner. So, Cobot effectively uses social feedback as reward, as does our simulated robot. However, there are several important differences. Cobot does not address the issue of a human observer parenting the robot using affective communication. Instead, it learns based on reinforcement extracted from words used by the user during the chat sessions in which Cobot is participating. Also, Cobot is not a real-time behaving robot, but a chat robot. As a consequence, time constraints related to the exact moment of administering reward or punishment are less important. Finally Cobot is restricted regarding its action-taking initiative, while our robot is continuously acting, with the observer reacting in real-time.

Future work includes a broader evaluation of the EARL framework including its ability to express emotions generated by an emotional model plugged into the RL agent. Further, we envision to experiment with controlling metaparameters (such as exploration/exploitation and learning rate) based on the agent's internal emotional state or social rewards (Belavkin, 2004; Broekens, Kosters, Verbeek, 2007; Doya, 2002). Currently we use simulated annealing-like mechanisms to control these parameters. Further, the agent could try to learn what an emotional expression predicts. In this case, the agent would use the emotional expression of the human in a more pure form (e.g., as a real-valued vector of facial feature intensities as part of its perceived state *s*. This might enable the agent to learn what the emotional expression means for itself instead of simply using it as reward. Finally, a somewhat futuristic possibility is actually quite close: affective Robot-Robot interaction. Using our setting, it is quite easy to train one robot in a certain environment (parent), make it observe an untrained robot in that same environment (child), and enable it to express its emotion as generated by its emotion model using its robot head, an expression recognized and translated into social rewards by the child robot. Apart from the fact that it is somewhat dubious if such a setup is actually useful (why not send the social reward as a value through a wireless connection to the child), it would enable robots to use the same communication protocol as humans.

Regarding the "usefulness" argument just put forward, it seems to apply to our experiment as well. Why didn't we just simulate affective feedback by pushing a button for positive reward and pushing another for negative reward (or even worse, by simulating a button press)? From the point of view of the robot this is entirely true, however, from the point of view of the human—and therefore the point of view of the human-robot interaction—not at all. Humans naturally communicate social signals using there face, not by pushing buttons. The process of expressing an emotion is quite different from the process of pushing a button, even if it was only for the fact that it takes more time and cognitive effort to initiate the expression. These are just two of many examples showing that expressing an emotion is quite different from pushing a button, and in a real-world scenario with a mobile robot in front of you it would be quite awkward to have to push buttons instead of just smile when you are happy about its behavior. Further it would be quite useful if the robot could recognize you being happy or sad and gradually learn to adapt its behavior even when you did not intentionally give it a reward or punishment. Abstracting away from the actual affective interaction patterns between the human and the robot in our experiment would have rendered the experiment almost completely trivial. Nobody would be surprised to see that the robot learns better if an intermediate reward is given halfway its route towards food. Our aim was to investigate if affective communication can enhance learning in a reinforcement learning setting. Taking out the affective part would have been quite strange indeed.

Acknowledgements

We would like to sincerely thank all the students who helped us develop the *EARL* system, thereby making this research possible. Joris Slob, Chris Detweiler, Sylvain Vriens, Koen de Geringel, Hugo Scheepens, Remco Waal, Arthur de Vries, Pieter Jordaan, Michiel Helvensteijn, Rogier Heijligers, Willem van Vliet, you were great!

This work was partly funded by the European Union (PACO-PLUS integrated grant, IST-FP6-IP-027657; www.paco-plus.org).

References

[Ashby, F. G., Isen, A. M., & Turken, U., 1999] A Neuropsychological Theory of Positive Affect and its Influence on Cognition. *Psychological Review, 106* (3): 529-550.

[Avila-Garcia, O., & Cañamero, L., 2004] Using hormonal feedback to modulate action selection in a competitive scenario. *From Animals to Animats 8: Proc. 8th Intl. Conf. on Simulation of Adaptive Behavior* (pp. 243-252). Cambridge, MA: MIT Press.

[Belavkin, R. V., 2004] On relation between emotion and entropy. *Proc. of the AISB'04 Symposium on Emotion, Cognition and Affective Computing* (pp. 1-8). AISB Press.

[Berridge, K. C., 2003] Pleasures of the brain. *Brain and Cognition 52*, 106-128.

[Blanchard, A. J., & Cañamero, L., 2006] Modulation of exploratory behavior for adaptation to the context. *Proc. of the AISB'06 Symposium on Biologically Inspired Robotics (Biro-net)* (pp 131-137). AISB Press.

[Breazeal, C., 2001] Affective interaction between humans and robots. In: J. Keleman and P. Sosik (eds), *Proc. of the ECAL 2001, LNAI 2159* (pp. 582-591).

[Broekens, J., Kosters, W.A., & Verbeek, F. J., 2007] On Emotion, anticipation and adaptation: investigating the potential of affect-controlled selection of anticipatory simulation in artificial adaptive agents. *in press*.

[Charman, T., & Baird, G., 2002] Practitioner Review: Diagnosis of autism spectrum disorder in 2- and 3-year-old children. *Journal of Child Psychology and Psychiatry, 43*(3), 289-305.

[Clore, G. L. & Gasper, K., 2000] Feeling is believing: some affective influences on belief. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge Univ. Press, Cambridge, UK.

[Cos-Aguilera, I., Cañamero, L., Hayes, G. M., & Gillies, A., 2005] Ecological integration of affordances and drives for behaviour selection. *Proceedings of the Workshop on Modeling Natural Action Selection* (pp. 225-228).

[Custers, R., & Aarts, H., 2005] Positive affect as implicit motivator" On the nonconscious operation of behavioral goals. *Journal of Personality and Social Psychology, 89*(2), 129-142.

[Damasio, A. R., 1994] *Descartes' error*. New York, NY: Penguin Putnam.

[Doya, K., 2002] Metalearning and neuromodulation. *Neural Networks, 15* (4), 495-506.

[Dreisbach, G., & Goschke, K., 2004] How positive affect modulates cognitive control: Reduced perseveration at the cost of increased distractibility. *Journal of Experimental Psychology: Learning, Memory, and Cognition, 30*(2), 343-353.

[Fong, T., Nourbakhsh, I., & Dautenhahn, K., 2003] A survey of socially interactive robots. *Robots and Autonomous Systems, 42*, 143-166.

[Forgas, J. P., 2000] Feeling is believing? The role of processing strategies in mediating affective influences in beliefs. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

[Frijda, N. H., & Mesquita, B., 2000] Beliefs through Emotions. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

[Frijda, N. H., Manstead, A. S. R. & Bem. S., 2000] The influence of emotions on beliefs. In: Frijda, N., Manstead A. S. R., & Bem, S. (Eds.), *Emotions and Beliefs*. Cambridge, UK: Cambridge University Press.

[Gandanho, S. C., 2003] Learning behavior-selection by emotions and cognition in a multi-goal robot task. *Journal of Machine Learning Research 4*, 385-412.

[Gasper, K., & Clore, L. G., 2002] Attending to the Big Picture: Mood and Global Versus Local Processing of Visual Information. *Psychological Science, 13*(1): 34-40.

[Isbell, C. L. Jr., Shelton, C. R., Kearns, M., Singh, S., Stone, P., 2001] A social reinforcement learning agent. *Proc. of Agents-01*. ACM.

[Lin, L. J., 1993] *Reinforcement learning for robots using neural networks*. Doctoral dissertation. Carnegie Mellon University, Pittsburgh.

[Mehrabian, A., 1980] *Basic Dimensions for a General Psychological Theory*. OG&H Publishers. Cambridge Massachusetts.

[Pantic, M. & Rothkranz, L.J.M., 2000] Automatic analysis of facial expressions: The state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22* (12), 1424-1445.

[Picard, R. W., 1997] *Affective Computing*. MIT Press.

[Picard, R. W., Papert, S., Bender, W., Blumberg, B., Breazeal, C. Cavallo, D., Machover, T., Resnick, M., Roy, D. & Strohecker, C., 2004]. Affective learning — A manifesto. *BT Technology Journal, 22*(4), 253-269.

[Rolls, E. T., 2000] Précis of The brain and emotion. *Behavioral and Brain Sciences, 23*, 177-191.

[Russell, J. A., 2003] Core affect and the psychological construction of emotion. *Psychological Review, 110*(1), 145-72.

[Scherer, K. R., 2001] Appraisal considered as a process of multilevel sequential checking. *Appraisal processes in emotion: Theory, Methods, Research*. Oxford Univ. Press, New York, NY.

[Sutton, R., & Barto, A., 1998] *Reinforcement learning: An introduction*. Cambridge, MA: MIT Press.

[Velasquez, J. D., 1998] A computational framework for emotion-based control. *In: SAB'98 Work-shop on Grounding Emotions in Adaptive Systems*.

Virtual Players in RPG

Diana F. Adamatti †, Jaime S. Sichman † and Helder Coelho ‡
†Escola Politécnica - Universidade de São Paulo - Brasil
diana.adamatti@poli.usp.br, jaime.sichman@poli.usp.br
‡Universidade de Lisboa - Portugal
hcoelho@di.fc.ul.pt

Abstract

Role-Playing Games (RPG) are well known entertainment games where players play (or act, or perform) characters and, by doing this, live different lives, full of fantasy and entertainment. When RPGs were created, they were played with cards and other printed materials. However, such way of playing leads to RPGs with reduced number of players and rules since the game complexity increases with the number of players and rules. Also, players are not able to memorize every data concerning the game to play it accordingly. Nowadays, the automation of RPGs turns them into complex and funny games, with graphic interfaces, a lot of information about other players and environment, etc. Due to this complexity, the minimum number of players needed to start the game may be large and the existence of semi-autonomous players to substitute real players would be very useful. In this paper we present a RPG for the natural resources management. It was implemented using the GMABS technology, which involves multiagent-based simulation techniques.

1 Introduction

Role-Playing Games (RPG) are a type of game where the players perform a character. This character is created inside of a particular scene (an environment). It follows a system of rules, that serves to organize its actions, determining the limits of what can or cannot be done [Klimick, 2003]. In this way, RPGs are games where each player plays a role and takes decisions to reach its owner's objectives. In fact, players use RPG like a ”social laboratory”, because they can try many possibilities, without real consequences [Barreteau *et al.*, 2003].

RPG can be printed (maps, cards of characters, etc.), electronic or oral. RPGs are in a specific category of games, because their purpose is collaboration, not competition. In fact, there are not winners or losers in RPGs, since the players must complete a story using the game rules to reach individual or collective objectives. Therefore, an important factor in RPGs is their integration capacity (played in groups and reached by cooperation). In this type of games, the interaction is very important (talking, dialoguing and changing ideas) to play it. There is a famous proverb in RPG: ”separated groups carry to simultaneous deaths” [Klimick, 2003].

RPG is a technique very used in training, because it can put the players in real situations of decision-making without real consequences. In special, big companies have used RPG during technical courses because this kind of game involves an amused factor, and the training and/or learning can occur in a facilitated way [Barreteau *et al.*, 2001].

Whenever a RPG is played manually (using printed materials, such as cards), the number of rules and players is small. The complexity of the game increases with the number of rules and players. Therefore, the collaboration and integration desired in the RPG could be prevented, since people may not memorize large number of rules and actions from their roles. A new alternative is to automatize the RPG using a software to play the game, to execute its actions and to return the new scenario of the game. This paper presents the implementation of an automated RPG to the natural resources management domain, the JogoMan prototype [Adamatti *et al.*, 2005], combined with virtual players and people.

We organized this paper in 6 sections. In Section 2 is presented the GMABS methodology. The section 3 presents the architectures to insert autonomous players in RPG. In the section 4 is presented the game domain problem, natural resources management, where we implemented an automatization to RPG. The section 5 presents how we implemented autonomous players in RPG and the first results with these players, and in the section 6 there are the conclusions and the further work.

2 GMABS Methodology

Multi-agent Systems (MAS) study the behavior of sets of independent agents with different characteristics, which evolve in a common environment. These agents interact with each other, and try to execute their tasks in cooperative way, sharing information, preventing conflicts and co-ordinating the execution of activities [Alvares and Sichman, 1997].

Additionally, the use of the simulation as an auxiliary tool for the human-being decision-making is very efficient, because it allows the verification of specific details with great precision. The combination of both, multi-agent systems and simulation, generates a new research area called multi-agent-based simulation (MABS) [Gilbert and Troitzsch, 1999], that

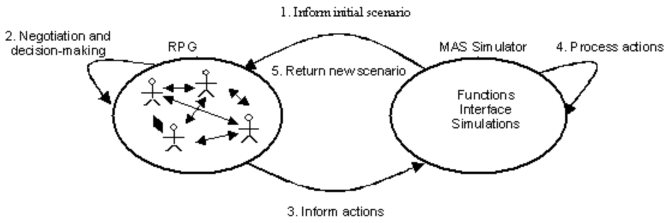


Figure 1: GMABS Methodology.

uses to deal with problems that involves multiple domains.

The use of MABS and RPG (isolated or in an integrated way) has been used in several works [Barreteau *et al.*, 2001; D'Aquino *et al.*, 2003], and they can bring interesting results, such as to join the dynamic capacity of MABS with the discussion and learning capacity of RPG techniques. In this paper, the integration of RPG and MABS is called GMABS (Games and Multi-Agent-Based Simulation) methodology. Figure 1 presents how GMABS is used in this context. The integration steps are described in the following:

1. Players receive all the information about the game (rules and initial scenario). The roles of each player are defined. For example, a game that has the roles of industry manager and ecologist, and whose objective is to verify the water quality of a determined region. Firstly, each player knows what rules each character can execute, the benefits and/or damages its action can cause to the quality of the water, as well as where they are physically located in the game and what are their possessions (money, lands, etc.). For this example, the industry manager role must have knowledge about the size, place, profitability, pollution average, etc., of its industry;
2. Players have all the information necessary to initiate the negotiations, they change information and do their decision-making (according to the rules initially determined) for their chosen roles. Normally, the duration of this step is defined in the beginning of the game (for example, 10 minutes). In some cases, a bigger time for this step is necessary, depending on the number of players, difficulty of the game rules, etc. For example, the industry manager can decide to increase its production, to sell properties, etc;
3. Players inform to the MAS simulator which were the chosen actions;
4. Data are computed by the simulator (process actions): these actions modify the initial scenario. The properties of the environment are modified, which implies the modification of each role data. For example, if the player who plays like the industry manager decides to install a new industry in the scenario, the player who plays like the ecologist realizes the increase of pollution in water. This step is the end of the first turn of the game;
5. MAS simulator returns new scenario. If the time of the game is not exceeded or the maximum number of rounds has not been achieved, returns to step 2.

This sequence could be repeated many times, depending on the objectives of the game. Normally, the first turn of simulation is longer (duration time), because the players are learning the rules and how to manipulate the resources that the game has. The following turns are shorter, since the players already have an objective and strategies to reach it. In the end, independent of the number of rounds, there always be a discussion (called *debriefing*) about the choices that were made for each player. This discussion has the objective of better understanding the specific problem and the possible solutions presented during the game [Dorn, 1989; Egenfeldt-Nielsen, 2004].

3 Semi-Autonomous RPG

Whenever a RPG is played, it needs a certain number of real players to be executed. However, many times the game cannot be played because it does not have the minimum number of players: in some cases, the players are in different places (if the game is played through the Web, for example) and/or at different time schedules (if the game is played in asynchronous way).

In this way, the existence of **Virtual Players** would be very useful, because they can substitute the real players without damaging the game. We understand by damaging the game a situation where real players easily identify easily the virtual players decisions, because the virtual players making-decisions are not realistic (actions very different from the ones real players expected to perform).

To implement the GMABS methodology, we need to analyze two aspects: *players and system operator*. The system operator is the one that feeds the simulator with input data and that fowards the scenario information to players. It can be a person (input data manually) or a system (input data automatically). The game players can be real (that play manually) or virtual (that play automatically). In Figure 2, we present these two levels of integration. In this figure, the computational system is represented by MABS element and the real or virtual players by RPG element. In Figure 2 (a), players and operator system are manual; in Figure 2 (b), system operator is manual and players are automatic; in Figure 2 (c), players are manuals and system operator is automatic; and in Figure 2 (d), all the system is automatic.

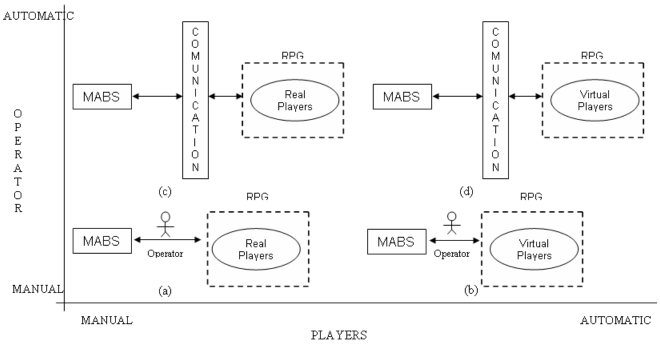


Figure 2: Integration levels of MABS methodology.

3.1 The Semi-Autonomous RPG Architecture

Our objective is developing an RPG in between the third and the fourth cases (Figure 2 (c) and (d)), a "Semi-Autonomous RPG" where we have an automatic system operator and sets of real and virtual players. Therefore, we have expanded the RPG element in two sub-elements: real players and virtual players (see Figure 3). When we insert virtual players in RPG, multi-agent simulation (element MABS) can be modified, because the data input can be done in different way. To prevent it, we included an intermediate communication layer between MABS and RPG elements. From the "MABS point of view", this communication layer brings the information exchange between the MABS and players in the same way. This layer does the communication between all players (real or virtual), during the decision-making.

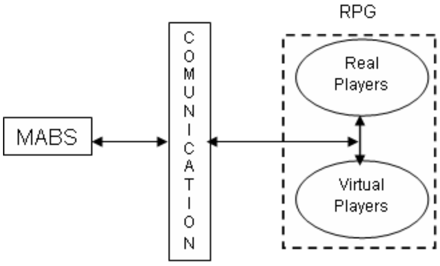


Figure 3: Semi-Autonomous RPG architecture.

The architecture presented in Figure 3 was defined to be a tool and domain independent architecture. We really expect this architecture being *generic*, since the GMABS methodology can be implemented for any knowledge area and for any multi-agent simulation tool. From it, we define some requirements to have communication between MABS and RPG elements:

- The MABS element must supply, in some format (for example ASCII), the system data, such as scenario and current players situation;
- The communication layer must have the "standard knowledge" of the domain. It must receive the information from the MABS and forward it to the real and virtual players in an integral and uniform way, without losing any information (and vice-versa: to return to MABS the players's chosen action, in order to execute them in the multi-agent simulator);
- The virtual players must receive information from the communication layer and must be able to manipulate them. Moreover, if necessary, communicate with other players (real and/or virtual) in order to get new information about the scene and/or players, to make decisions.

3.2 Cognitive Architecture for Autonomous Players

In order to provide virtual players with the ability of receiving and manipulating new information, as well as communicating with other players (real and/or virtual players), we have defined a cognitive architecture for them.

The cognitive architecture allows virtual players to previously define objectives and strategies to be used in the specific domain; to provide means for communication (with virtual and/or real players and the environment); and to take decisions based on their objectives and strategies. It is a layered architecture that present the communication layer as the master layer, since all layers are related to it.

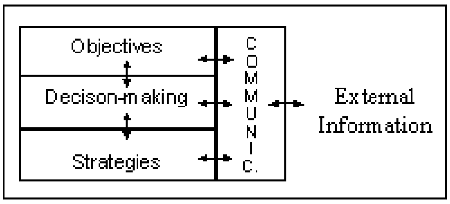


Figure 4: Cognitive Architecture for Virtual Players.

The internal communication layer of each virtual player (Figure 4) must be compatible with the communication layer of the generic architecture (Figure 2 (d)) to allow integration between the two architectures. It is the only constrain the architecture, since the internal configuration of the virtual players architecture must be independent of the GMABS generic architecture implemented.

Figure 4 shows communication arrows between all architecture elements (Objectives, Decision-making, Strategies and Communication). It means that all elements can exchange information and can search for external information, in order to choose the best strategy to achieve the defined objectives. In this way, the cognitive architecture allows:

- To define the objectives and strategies *a priori*;
- To have a technique (method) for decision-making;
- To have communication between all players (real and/or virtual) through the internal communication layer that must be compatible to external communication layer;
- To have exchange information between players and environment through the compatibility between all architecture elements (Objectives, Strategies, Decision-making and Communication).

4 RPG in Natural Resources Management

We have developed a prototype called **JogoMan** (the Portuguese acronym for "**Jogo** dos **Man**anciais" that means in English: Water Sources Game). It was built according to GMABS methodology, which simulates the management of a particular peri-urban catchment, located at Bacia do Alto Tietê, in São Paulo, Brazil (inserted in the Project Negowat[1]). This prototype was implemented in Cormas [Cormas, 2004], a MABS simulator tailored to natural resources.

JogoMan represents a simplification of the real phenomena of interaction between the several actors, in the context of the peri-urban catchment previously described.

[1] Negowat Project:*Facilitating Negotiations Over Land and Water Conflicts in Latin American Peri-Urban Upstream Catchments: Combining Multi-Agent Modeling with Role-Playing Games.*

The specific objective of this game is to determine water quality and quantity in a peri-urban catchment. It involves the management of land and water related problems in different cities. The game environment is a grid divided in portions (parcels). Each portion represents a real state (or a piece of land) that is associated with an owner (the player) and a use (such as agriculture or forest). The game allows players to: change the use of their land; put some infrastruture on them and sell/buy their/other portions. There are four types of players, each one having different goals.

- **Land Owner**: a land owner has some portions of space, each one with a land use type, such as forest or agriculture. For each different land use type, there are different values to maintenance and financial return. Owners can sell or buy their private areas or they can change land use of these areas. Land owners should demand to mayors the construction of infrastructure in their cities.
- **Mayors**: The game has different cities, each one having its mayor. The mayor goals are closely related to the city main activity (urban, agricultural, etc.) For example, the city "C" is a preservation area, and the player in the role of "Mayor C" should preserve this city. The mayors can invest on public infrastructure, such as portable water net or to build schools, hospitals or polices headquarters.
- **AguaPura Company Administrator**: This player can invest on public infrastructure to improve water quality: portable water and sanitation net.
- **Migrant Representative**: This player has a special role in the game, since he/she must allocate a number of new families. These families arrive in the cities (urbanization pressure), and they can be allocated in settlement or in slums. The quality and/or quantity of water of the region is modified depending on where these families are placed (settlement or slums).

Each player chooses his/her actions individually, but he/she should know that these actions have consequences to other players, because the quality and quantity of water depends on the land use and infrastructure. For example: if a mayor decides to decrease the land taxes for land owners that preserve the forests, various land owners can decide to maintain their areas with forest or even decide to plant forest (reforestation). This action influences every players, because the water quality probably will improve. Other example, a land owner decides to build an industry. The industry profit is larger, but the water pollution is larger too.

We perform four tests with prototype JogoMan. The first was the Negowat Project staff (researches and graduate students) and the other by undergraduate students from São Paulo. The next tests will be applied to community groups in São Paulo peri-urban catchment.

The sequence of steps for the test were:

1. General explanation for all participants of the game, presenting objectives and roles (possible players).
2. Each person choose a role (a player).
3. For each different player, specific information are given. For example: mayors know how much money they have and what are the actions they can execute.
4. The first round is stayed. Usually, it is longer, because the players do not have knowlodge about all the actions they can execute and the benefits/damages these actions can cause. A time of 30-40 minutes was defined for the first round.
5. Players inform to the MABS operator which were the chosen actions.
6. Actions are computed by MABS. These actions will modify the initial scenario (first round complete). The operator shows the new scenario to the players.
7. When the rounds finish (normally 3 or 4 rounds), we do a debriefing, to check doubts and suggestions (through a questionnaire). This step is extremely important, because it helps us to improve the prototype.

According to Egenfeldt-Nielsen [Egenfeldt-Nielsen, 2004], one of the main problems of presenting test results is that the game evaluation methods are a problem in themselves, where it has been questioned if we can use traditional methods for measuring learning outcome. Our tests, results are qualitative, instead of quantitative. It means that we do not use a mathematical method to determine if the game is good. Therefore, observe and ask about the game using questionnaires to obtain more information. Some suggestions/information pointed by players during the debriefing questionnaire (step 7 in our application sequence) were:

- Most of the participants thought the game is very interesting and realistic, helping them to understand the reality in peri-urban catchments.
- The participants also affirmed they learned a lot, because RPG is a didactic and funny form to learn a new topic.

The test results bring us some ideas for modifing the game in order to take it closer to the reality. More details about the prototype, can be found in previous paper [Adamatti *et al.*, 2005].

5 Semi-autonomous RPG in JogoMan

The architectures presented on section 3 were implemented in the JogoMan domain, a perin-urban catchment. The goal of the implementation is to prove that it is possible to insert virtual players in a RPG without having damages on it.

We chose the following tools, aiming to attend the previous defined requirements in the section 3 (see Figure 5):

- *MABS Tool*: *Cormas* is used as simulator [Cormas, 2004], because it was used to implement the JogoMan prototype. *Cormas* has specific functions to extract system data in different formats (ASCII, Excel and in the formats for data base Oracle, MSAccess, MySQL or PostGre);
- *Virtual Players*: as the cognitive architecture for decision-making of the virtual players, we chose BDI architecture, because it already has a logic defined from *AgentSpeak(L)* language [Rao, 1996] and tools developed for it, such as *Jason* interpreter [Bordini and Hubner, 2004]. This interpreter allows that each step in BDI

logic can be visualized and analyzed individually. It also allows communication between virtual players, as well as between virtual players and the environment.

- *Communication Layer*: we chose the SACI (*Simple Agent Communication Infrastructure*) tool [Hubner and Sichman, 2000] as the communication layer between real and virtual players. This tool provides communication infrastructure for agents, using the KQML (*Knowledge Query and Manipulation Language*) language [Labrou and Finin, 1997], and is used by *Jason*. The communication layer between MABS and RPG elements used the SOAP (*Simple Object Access Protocol*) protocol [SOA, 2005], because the MABS tool (Cormas) and Jason were implemented in different programming languages (SmallTalk and Java, respectively), and the SOAP technology provides interoperability between both languages through use of XML.

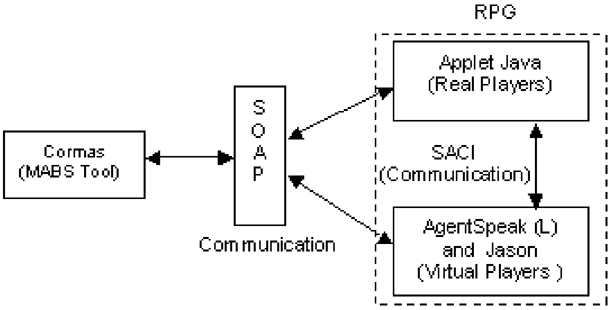

RPG
Applet Java (Real Players)
SACI (Communication)
AgentSpeak (L) and Jason (Virtual Players)
Cormas (MABS Tool)
S O A P
Communication

Figure 5: Selected Tools to Semi-autonomous RPG.

Nevertheless, the technology used was chosen to develop a web-based semi-autonomous RPG, meaning that it executes in a Web Server and it is accessed simultaneously by several players through web-browsers.

5.1 Evaluate Methodology

Before starting the test with virtual players, we have defined a test evaluation methodology, to know exactly what must be tested and what may be analyzed from the test results. We did not find an evaluation methodology for games (RPG) and simulation (MABS), the two areas involved in this work, then we defined two ways to evaluate our system:

1) Definition of behavioral profiles to virtual players: each type of player (Land Owner, Mayor, AguaPura Company Administrator and Migrant Representative) has profiles. Depending on the player, it can have two (Land Owner, AguaPura Administrator and Migrant Representative) or three (Mayor) profiles. To discover these profiles, we mapped objectives and strategies of the real players by questionnaires during *debriefing* during the JogoMan tests without virtual players. Having this, we analyzed and discovered a sequence of actions that each player could execute. Each profile has some specific variables to measure the proposed objective. These profiles were analyzed and evaluated by specialists of natural resources to verify if the possible strategies and actions are similar to real player activities. For example, the Land Owners may have an economic profile, and all their strategies lead to save and earn money. They are not worried about the reservoir pollution level. The analyzed variable will be the quantity of money in their ”cash box”.

2) Application of pre and pos questionnaires to real players: the pre questionnaire verifies the knowledge level of players in this area. We apply the pos questionnaire to verify if the virtual players decision-making was realistic. An important question in the pos questionnaire is: ”Some synthetic players may be included in this game (non human-being players). Do you discern if any player has a non human behavior? Which one?”. The idea is that real players do not discern between real and virtual players during the game. Just to do clear, it is not a ”Turing Test” and probably, some virtual players will be ”discovered”, but we hope that the virtual players behavior will be not so silly in front of real players behavior.

5.2 Preliminary Tests and Results

The initial tests were done with autonomous players in JogoMan to verify objectives and strategies defined for each behavioral profile. It is a game with virtual players only (integration level (d) in Figure 2). We have tested a scenario with 14 players (9 Land Owners, 3 Mayors, 1 Migrant Representative and 1 AguaPura Company Administrator). The following behavioral profiles to the players, with different objectives, were selected:

- 5 Land Owners with economic profile: to save and earn money;
- 4 Land Owners with ecologic profile: to improve the ecological situation of its region;
- 1 Mayor with social profile: to improve the life quality of people in its city;
- 1 Mayor with economic profile: to improve the life quality of people if it has money to do it;
- 1 Mayor with ecological profile: to improve the ecological situation in its city;
- 1 AguaPura Company Administrator with rational profile: to improve water and sanitation networks with a rational use of money;
- 1 Migrant Representative with economic profile: to allocate families without worring about the ecological situation of the region.

Table 1 presents the players ”cash box” data after 4 rounds of the game, in order to compare economic/rational profiles and ecological/social profiles of all players. It shows that the economic profiles earned more money than the ecologic profile, after 4 rounds. Also, most of the players with ecological profiles had negative or low values in their cash boxes.

Figures 6 and 7 present the values of reservoir pollution and water/sanitation network rates in region, respectively, to verify the ecological/social profiles of players. In these figures, the round label starts in 1 (the value presents to players in the first round of game) and they finish in 5 (the values presents to play a fifth round).

Figure 6 shows that reservoir pollution level is decreased, with the initial value was 4.800 and final value was 3.316.

Table 1: Values of cash box after 4 rounds.

Player	Initial Value	Final Value
LandOwner Ecologic 1	2.000	-26.800
LandOwner Economic 2	2.000	29.600
LandOwner Ecologic 3	2.000	-3.304
LandOwner Economic 4	2.000	26.600
LandOwner Ecologic 5	2.000	-13.200
LandOwner Economic 6	2.000	8.800
LandOwner Ecologic 7	2.000	-51.500
LandOwner Economic 8	2.000	32.000
LandOwner Ecologic 9	2.000	2.592
Mayor Economic A	30.000	121.600
Mayor Social B	30.000	-61.600
Mayor Ecologic C	30.000	34.000
AguaPura Rational	60.000	-7.920
MigrantRepr. Economic	0	12.321

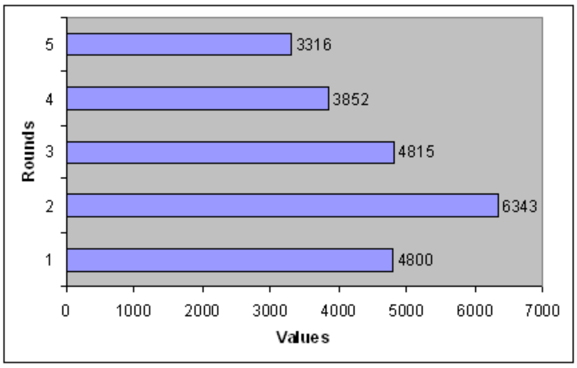


Figure 6: Values of Reservoir Pollution after 4 rounds.

Figure 7 shows the increasing of the water rates from 0,80 (initial value) to 1,0 (final value) and sanitation rates from 0,4 (initial value) to 1,0 (final value). These values indicated that ecological/social profiles are help to improve the ecological situation of the region.

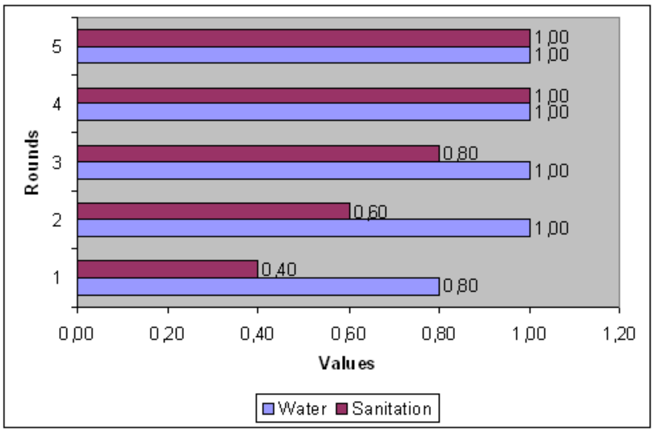


Figure 7: Water and Sanitation Network Rates after 4 rounds.

6 Conclusions and Further Work

The GMABS methodology was developed to be used as a support tool for negotiation, helping to solve conflicts in several areas, as natural resources management. It is because this methodology uses the MABS dynamic capacity and the RPG discussion capacity [Barreteau *et al.*, 2001].

We already developed a new prototype of JogoMan, with autonomous players, the Virtual Players. To do it, we used the architectures defined in section 3, and we chose the tools presented in 5. All this process was very hard, because involved different techniques and programming languages. Besides it, we wanted that it ran in a Web server.

We need to test and analyze the Semi-Autonomous RPG, with a mix of virtual and real players and we will compare these two different test with autonomous players and JogoMan test without them (section 4), where we have a good number of tests and results already analyzed. With these tests, we want to prove that our autonomous players can substitute the real players without damages to the game, and they can help to show new game views. The real players will answer the questionnaires, and we will have more data to analyze and compare our virtual players.

Another good improvement of the prototype could be the implementation of a dynamic knowledge base of semi-autonomous players. Until now, we implemented in static way, but we want to insert new beliefs and plans into the profiles with old actions of the players. It will improve the set of actions to each profile and the game will be more realistic.

Acknowledgment

Diana F. Adamatti is financed by CNPq and CAPES (Brazil), grant number 141851/2004-0. Jaime S. Sichman is partially financed by CNPq (Brazil), grant numbers 482019/2004-2 e 304605/2004-2.

The authors would like to thank Dr. Raphaèle Ducrot and Negowat Project, to help in JogoMan definitions and tests; and Anarosa Brandão to help in structure and organization of this paper.

References

[Adamatti *et al.*, 2005] D.F. Adamatti, J.S. Sichman, P. Bommel, R. Ducrot, C. Rabak, and M.E.S.A Camargo. JogoMan: A prototype using multi-agent-based simulation and role-playing games in water management. In N. Ferrand, editor, *Join Conference on Multi-Agent Modeling for Environmental Management. CABM-HEMA-SMAGET*, Bourg-Saint-Maurice, Les Arcs, France, 2005.

[Alvares and Sichman, 1997] L. O. C. Alvares and J. S Sichman. Introdução aos sistemas multiagentes. In *Jornada De Atualização Em Informática*, pages 1–37, Brasília - UnB, 1997.

[Barreteau *et al.*, 2001] O. Barreteau, F. Bousquet, and J. Attonaty. Role-playing games for opening the black box of multi-agent systems: method and lessons of its application to Senegal River Valley irrigated systems. *JASSS*, 4(2), March 2001. http://www.soc.surrey.ac.uk/JASSS/4/2/5.html.

[Barreteau *et al.*, 2003] O. Barreteau, C. Le Page, and P. D'Aquino. Role-playing games, models and negotiation. *JASSS*, 6(2), March 2003. http://jasss.soc.surrey.ac.uk/6/2/10.html.

[Bordini and Hubner, 2004] R. Bordini and J. Hubner. JASON: A Java-based Agentspeak interpreter used with Saci for multi-agent distribution over the Net. http://jason.sourceforge.net/, 2004.

[Cormas, 2004] Cormas. Natural resources and multi-agent simulations, 2004. http://cormas.cirad.fr.

[D'Aquino *et al.*, 2003] P. D'Aquino, C. Le Page, F. Bousquet, and A. Bah. Using self-designed role-playing games and a multi-agent systems to empower a local decision-making process for land use management: The selfcormas experiment in Senegal. *JASSS*, 6(3), June 2003. http://jasss.soc.surrey.ac.uk/6/3/5.html.

[Dorn, 1989] D. S. Dorn. Simulation games: One more tool on the pedagogical shelf. *Teaching Sociology*, 17(1):1–18, January 1989.

[Egenfeldt-Nielsen, 2004] S. Egenfeldt-Nielsen. Review of the research on educational usage of games. http:///itu.dk/people/sen/public.htm, 2004.

[Gilbert and Troitzsch, 1999] N. Gilbert and K. G. Troitzsch. *Simulation for the Social Scientist*. Buckingham and Philadelphi: Open University Press, 1999.

[Hubner and Sichman, 2000] J. F. Hubner and J. S. Sichman. SACI: Uma ferramenta para implementação e monitoração da comunicação entre agentes. In M.C. Monard and J. S. Sichman, editors, *IBERAMIA/SBIA 2000, Open Discussion Track*, pages 47–56, Atibaia - São Paulo - Brasil, 2000.

[Klimick, 2003] C. Klimick. Construção de personagem & aquisição de linguagem: O desafio do RPG no INES. Master's thesis, Programa de Pós-graduação do Departamento de Artes e Design - PUC, Rio de Janeiro, 2003.

[Labrou and Finin, 1997] Yannis Labrou and Tim Finin. A proposal for a new KQML specification. Technical Report TR CS-97-03, Computer Science and Electrical Engineering Department, UMBC, Baltimore, 1997.

[Rao, 1996] A. S. Rao. AgentSpeak (L): BDI agents speak out in a logical computable language. In Walter Van de Velde and John Perram, editors, *Seventh Workshop on Modelling Autonomous Agents in a Multi-Agent World (MAAMAW'96)*, pages 42–55, London, January 1996. Eindhoven - The Netherlands, Lecture Notes in Articial Inelligence - Springer-Verlag.

[SOA, 2005] W3C - World Wide Web Consortium - SOAP - Simple Object Access Protocol - Specifications. http://www.w3.org/TR/soap/, 2005.

Unexploited Dimensions of Virtual Humans

Zsófia Ruttkay, Dennis Reidsma, Anton Nijholt
Human-Media Interaction, Department of Computer Science,
University of Twente, Enschede, the Netherlands
{zsofi,dennisr,anijholt}@cs.utwente.nl

Abstract

Virtual Humans are on the border of fiction and realism: while it is obvious that they do not exist in reality and function on different principles than real people, they have been endowed with human features such as being emotionally sensitive. In this article we argue that many dimensions, both human-like ones and ones made possible by the computer technology, are still unexploited to increase the effectivity and engagement of interaction with VHs.

1 Introduction

The concept of *Human Computing* unifies several objectives: to make the usage of computers easy and natural, allowing such very 'human' behaviors like being emotional or bored, endow computer systems with adaptive and empathic response, facilitating applications where joy and engagement become more important than the problem-solving oriented 'categorical computing' practice.

To achieve this, application interfaces need to be able to handle more subtle interactions than those afforded by more basic "push-button type" interfaces. Very sophisticated sensing modules are needed to cope with subtleties in the behavior of the user. Also, the application needs to be able to communicate information to the user with a comparable level of complexity.

Virtual Humans (VHs) [Plantec, 2004] – also known under other names like humanoids [Thorisson, 1996], embodied conversational agents [Cassell et al., 2000] or intelligent virtual agents – are computer models resembling humans in their bodily look and their communication capabilities. In this paper we focus on their role in human-computer interaction, assuming a scenario where one or more VHs interact with one real person, in order to accomplish some task. We keep the traditional terms *user* and *task*, but will use it in a broadened sense. Because of their human-likeness, VHs seem to be the ultimate interface in the Human Computing age. From the point of view of HCI, there are two major motivation of 'putting a VH on the screen':

- By replacing the traditional computer-specific interaction modalities (keyboard, mouse) by the natural communicational capabilities of people, the services of computers will be *accessible to a broad population*, without respect to (computer) literacy, cultural and social background;
- VHs make *new applications* possible, where they fulfill traditional human roles, such as tutor, salesperson, and partner to play a game or chat with. Moreover, there are applications where the VH is in an entirely new role, without parallel in real life, such as human-like characters with fictional capabilities in (educational) games [Gustafson et al., 2004], interactive drama [Mateas and Stern, 2003] or Virtual Reality applications [Abaci et al., 2004], or a VH taking partial or split role known from real-life situations [Baylor and Ebbers, 2003].

Often it is not easy to separate the two aspects in an application: e.g. a VH explaining how to operate a device [Badler, 2002], can be seen as a friendly interface replacing queries and search in an on-line textual help, but it also extends the services, by being able to demonstrate operations, possibly in a way tailored to the user's capabilities (e.g. handedness).

An outsider to the field may wonder whether viewers perceive and react to Virtual Humans as to real humans. The increasing body of experimental studies indicates that the answer is yes. A subtle difference in facial expression [Walker et al., 1994], meta-speech characteristic or posture [Isbister and Nass, 2000] of VHs resulted in different subjective judgment as well as task performance of the people interacting with them. In these experiments the VH was not to be mistaken for a real person, and in some cases the design was clearly non-realistic. All the same, the nonverbal cues were interpreted with reference to the practice of real-life conversation. It has also been proven that with time people build up a relationship with a VH e.g. as with a coach.

The potentials of VHs are huge. Challenges to current state of the art of separate disciplines – e.g. speech recognition and language understanding, or computer vision – , have been identified as a must to improve technical quality, and related to this, engagement and effectivity of VHs [Gratch et al., 2002]. Also, the necessity of cooperation of different disciplines, and particularly, dedicated studies providing basis for computational models of human-human interaction, have been underlined [Isbister and Doyle, 2004].

In this paper, we look at VHs from the Human Computing perspective, and dwell on possible improvements of VHs of a different nature. Namely, we outline features which have not (or hardly) been exploited, and relate the virtual and the human domain. Several of these are technically feasible, or would require some routine engineering development. For instance, current VHs look as puppets took out of a box, new forever and unchanged, very much unlike real people, who may change dress every day, look more tired at the end of a busy day than in the morning, and their mood and mental state is not exactly the same every morning. This paper is a 'twin' of another recent paper by us [Ruttkay et al. 2006a], where we concentrated among other things on the practical value of some subtle aspects of human communication which have often been left out of the repertoire of 'machine humans' since they are seen as 'incorrect' or 'undesirable' in some way, such as disfluency and ambiguity. Here we look at the *fictional/real* polemic: a VH, on the one hand, is a fictional character, created and empowered by computer technology, so there is no need to model certain human characteristics, especially those which 'make no sense' or mean limitations for a fictional character, such as getting tired. Hence it is clear, and from the fictional point of view natural, that a VH would never ever get exhausted physically or mentally as real people do. On the other hand, if they remain ever fit, they are a mismatch to the human partner, who does get tired after some time. Such a mismatch could not only produce a feeling of inferiority and thus discomfort by the user, but may put him to danger, if he would try to keep up with the constant high performance of e.g. a virtual fitness trainer.

We see two reasons why the above and other similar aspects of VHs have not been considered:

- There is uncertainty about how *human-like* [Koda and Maes, 1996] and *realistic* [Dautenhahn, 2004] VHs should be. Is it at all wanted that a VH, who is not made of flesh and blood, gets tired, or has an own wardrobe that also reflects the seasons?
- Besides the hard core issues of e.g. understanding language, subtle features like style may be considered as not necessary, or not of high priority to be dealt with.

Hence, sophisticated behavior is to be seen not as a requirement for an entirely realistic VH for its own sake, but as an instrument to achieve the preferred perception of the VH by the user (e.g. in-group with the VH), which will subsequently influence the engagement and task performance of the user. In this paper we only concentrate on the 'unexploited' aspects and do not cover characteristics that have been extensively addressed earlier, such as 'showing empathy' or 'providing positive feedback'.

In the next section, we introduce three novel applications being developed at our group: the Virtual Dancer, the Virtual Conductor, and the Virtual Trainer. In Section 4 we discuss one by one the yet unexplored aspects of VHs and explain their potential merits. While our arguments are intended as general guidelines for next-generation VHs, we will illustrate them with examples from our own applications, in part already implemented and in part only envisioned. Finally, in the Discussion we return to the question of fiction and reality of VHs.

3 VHs in novel applications

In this section, we present three applications currently being developed at the HMI (Human Media Interaction) research group: the Virtual Dancer [Reidsma et al., 2006], the Virtual Conductor [Bos et al., 2006], and the Virtual Trainer [Ruttkay et al., 2006b]. These three applications are shown in Figure 1. These seemingly very different applications share some basic features, and have actually been developed relying on a similar framework. In all three applications, the VH:

- has visual and/or acoustic perception capabilities,
- has to monitor and react to the user continuously,
- has to use subtle variants of a motion repertoire generated on the fly, and
- uses both acoustic (music, speech) and nonverbal modalities in a balanced and strongly interwoven manner.

3.1 The Virtual Dancer

In a recent application built at HMI, a virtual human – the Virtual Dancer – invites a real partner to dance with her [Reidsma et al., 2006] (see Figure 2). The Virtual Dancer dances together with a human 'user', aligning its motion to the beat in the music input and responding to whatever the

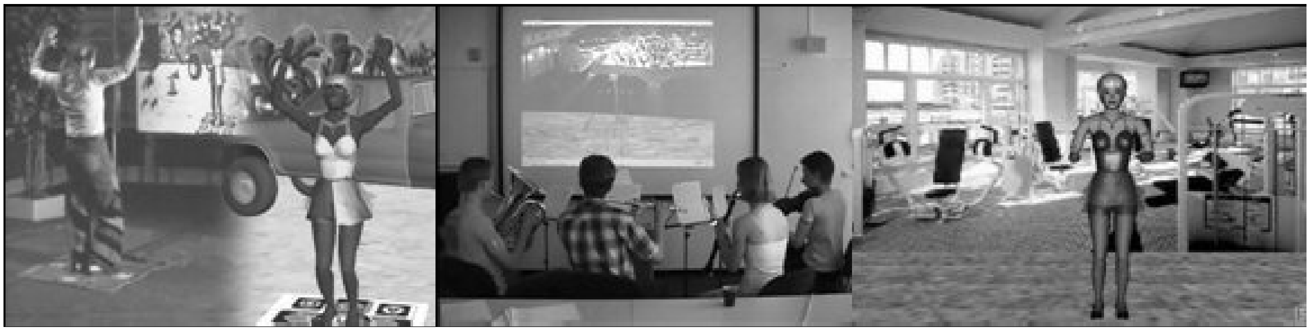


Figure 1. Three novel applications: Virtual Dancer, Virtual Conductor, and Virtual Trainer

human user is doing. The system observes the movements of the human partner by using a dance pad to register feet activity and the computer vision system to gain information about arm and body movements. Using several robust processors, the system extracts global characteristics about the movements of the human dancer like how much (s)he moves around or how much (s)he waves with the arms. Such characteristics can then be used to select moves from the database that are in some way ‘appropriate’ to the dancing style of the human dancer.

There is a (non-deterministic) mapping from the characteristics of the observed dance moves to desirable dance moves of the Virtual Dancer. The interaction model reflects the intelligence of the Virtual Dancer. By alternating patterns of following the user or taking the lead with new types of dance moves, the system attempts to achieve a mutual dancing interaction where both human and virtual dancer influence each other. Finding the appropriate nonverbal interaction patterns that allow us to have a system that establishes rapport with its visitors is one of the longer term issues being addressed in this research.

Clearly, the domain of dancing is interesting for animation technology. We focus on the interaction between human and virtual dancer. The interaction needs to be engaging, that is, interesting and entertaining. First experiences with demonstration setups at exhibitions indicate that people are certainly willing to react to the Virtual Dancer (see Figure 1)**.**

3.2 A Virtual Conductor

We have designed and implemented a virtual conductor [Bos et al., 2006] that is capable of leading, and reacting to, live musicians in real time. The conductor possesses knowledge of the music to be conducted, and it is able to translate this knowledge to gestures and to produce these gestures. The conductor extracts features from the music and reacts to them, based on information of the knowledge of the score. The reactions are tailored to elicit the desired response from the musicians.

Clearly, if an ensemble is playing too slow or too fast, a (human) conductor should lead them back to the correct tempo. She can choose to lead strictly or more leniently, but completely ignoring the musicians’ tempo and conducting like a metronome set at the right tempo will not work. A conductor must incorporate some sense of the actual tempo at which the musicians play in her conducting, or else she will lose control. If the musicians play too slowly, the virtual conductor will conduct a little bit faster than they are playing. When the musicians follow him, he will conduct faster yet, till the correct tempo is reached again.

The input of the virtual conductor consists of the audio from the human musicians. From this input volume and tempo are detected. These features are evaluated against the original score (currently stored in MIDI) to determine the conducting style (lead, follow, dynamic indications, required corrective feedback to musicians, etc) and then the appropriate conducting movements of the virtual conductor are generated. A first informal evaluation showed that the Virtual Conductor is capable of leading musicians through tempo changes and of correcting tempo mistakes from the musicians. Computer vision has not yet been added to the system. That is, musicians can only interact with the conductor through their music. In a future implementation we can look at the possibility to have the conducting behavior directed to (the location of) one or more particular instruments and their players.

3.3 The Virtual Trainer

The *Virtual Trainer* (VT) application framework is currently under development [Ruttkay et al., 2006a] and involves a virtual human on a PC, who presents physical exercises that are to be performed by a user, monitors the user’s performance, and provides feedback accordingly at different levels. Hence, our VT should fulfill most of the functions of a real trainer: it not only demonstrates the exercises to be followed, it should also provide professionally and psychologically sound, human-like coaching. Depending on the motivation and the application context, the exercises may be general fitness exercises that improve the user’s physical condition, special exercises to be performed from time to time during work to prevent for example RSI (Repetitive Strain Injury), or physiotherapy exercises with medical indications. The focus is on the reactivity of the VT, manifested in natural language comments relating to readjusting the tempo, pointing out mistakes or rescheduling the exercises. When choosing how to react, the static and dynamic charac-

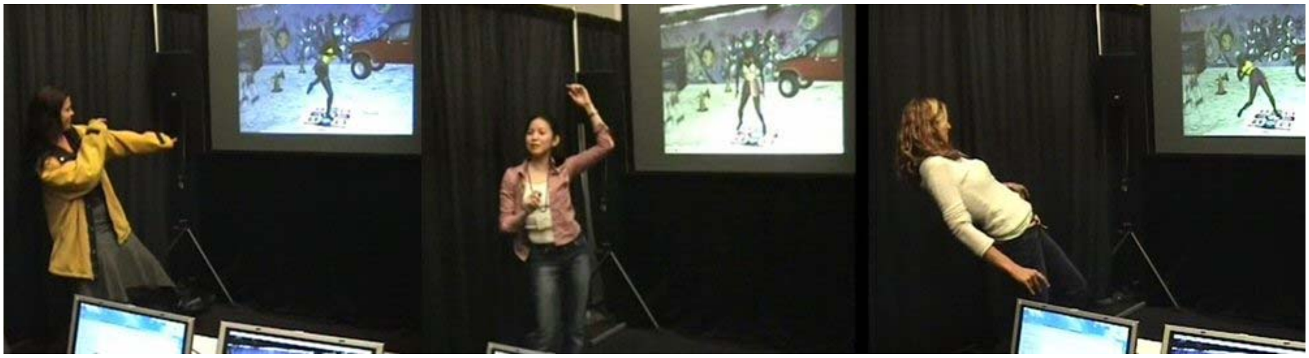


Figure 2. Interacting with the Virtual Dancer

teristics of the user and the objectives to be achieved are to be taken into account and evaluated with respect to biomechanical knowledge and psychological considerations of real experts. For example, if the user is just slowing down, the VT will urge him in a friendly way to keep up with the tempo, acknowledge with cheerful feedback good performance and engage in a small talk every now and then to keep the user motivated.

The VT is adaptable and adaptive in several respects. The embodiment can be chosen such that it reflects the geometrical and physiological motion characteristics of the user. For this purpose, some data (age, gender, weight, goal of the training) may be asked for, or gained by computer vision. The motion characteristics may be gained by in an initial calibration session by computer vision, analyzing the user's motion perform a few special moves. The exercises to be presented may be authored by an authorized person, such as a real physiotherapist to whom the VT acts as an 'assistant'. The motion and exercise repertoire of the VT may be extended, by providing an exercise editing interface, allowing also the incorporation of complex motions which were pre-acted and motion captured.

The VT keeps record of the sessions with the user, and interprets his performance in the light of short and long-term history. Also the VT addresses the user in a personal manner, using his name and a style most appropriate for the given user's age.

4 Unexploited aspects of VHs

In the past decade, much effort has been spent on improving the human-likeness of individual modalities of VHs, such as improving the quality of synthesized speech [Van Moppes 2002], modeling expressive gesturing of humans [Hartmann et al., 2005], deriving computational models to capture the kinematics [Wachsmuth and Kopp, 2002], providing means to fine-tune the effort and shape characteristics of facial expressions and hand gestures [Chi et al., 2000], model gaze and head behavior, add biological motions like blinking or idle body motion [Egges et al., 2004].

The fusion of multiple modalities has been dealt with, from the point of view of timing of the generated behavior, and the added value of using multiple modalities in a redundant way. It has been suggested that VHs, just as real people, should be endowed with a style, reflected in the usage of verbal and non-verbal modalities to express some meaning, and the intonational and motion characteristics of the single modal signals [Ruttkay et al., to appear]. Besides the features typical for the VH as an individual, his (assumed) social, cultural and professional background should be the components which contribute to the style. The importance of cultural and social connotation of a VH has been pointed out [Payr and Trappl, 2004, Prendinger and Ishizuka, 2001].

Modeling emotions, mood and personality [Gratch and Marsella, 2001] and their benefits in judging the VHs have been extensively addressed. Initially, the 6 basic emotions were to be shown on the face [Ekman, 1989], which has been followed by research on taking into account display rules, resulting in emotions to be hidden [Poggi et al., 2001], or overcast by fake expressions e.g. to hide lies [Rehm and Andre, 2005], studying principles to show mixed emotions on the face, to reflect emotions in gesturing [Noot and Ruttkay, 2004].

In addition to emotions, the importance of small talk in building a common ground and trusting the VH [Bickmore and Cassell, 2000], as well as back-channeling have been emphasized. Having long-term attitude towards a VH, like friendship, has been pointed out [Stronks et al., 2002].

4.1 Embodiment: beyond the perfect and generic

VHs should go beyond the present state of generic, doll-like and usually perfect-looking (symmetrical, young and spotless face and body) design. To begin with, the embodiment should reflect age, ethnicity and social status most appropriate for the given application and the user group. Besides, a VH should have individual features, which may make him easier to identify, remember and enjoy. The individuality may be in granularity of detail of photorealistic nature (e.g. applying subtle texture for the skin), or in exaggerated cartoonish features [Liang et al., 2002] or variety resulted from 'noisy' parameters [DiPaola, 1991]. The phenomena 'uncanny valley', formulated by Mori [Mori, 1982] stating that increasing realism, after a certain degree, dramatically decreases the perceived human quality of a robot, is widely assumed to apply for VHs too.

The technology allows that the embodiment (and also, the communicational and mental characteristics) of a VH should be chosen according to the given user. The 'mirroring phenomena', stating that people are positively prejudiced to others who resemble, by and large, themselves, could be turned to good use: by 'looking' at the user first, the best matching VH embodiment could be chosen, based on the assumption that such a VH will be the most trusted and effective [Bailenson, and Yee, 2005]. We believe that there is much use for this type of 'adaptive appearance' for at least the Virtual Dancer and the Virtual Trainer, certainly concerning the gender and age parameters. For the VT, the body geometry in general or based on measurements could be a useful parameter to be adjusted. A recent study on preference of the virtual eHealth advisors points in the direction that a somewhat bulky figure is more appreciated than one with the ideal weight.

Another source of variety is in subtle temporal differences of the appearance of a given VH. Changes in outfit (hair, clothing) and signs of physical state would make the VH more enjoyable and life-like. Moreover, such variations in appearance could be used to reinforce the presence in the geographical location and time of the user (see below). How about having the VT pop up on a hot day in appropriate summer dress, sun-burnt? And would not it be nice if the VT would start sweating too after some time, not only the user? These temporal changes should be consistent with each other and with the identity and history of the VT (see below).

4.2 VH- with an own history and identity

“Who are you?” do people ask (usually as one of the first questions) from their VH interlocutor. The answer is a name, may be extended with the services the VH can offer. In case of chat bots, a date of birth may be given, and the creator may be named as ‘father’, such as in the case of Cybelle [AgentLand, 2006]. Notably, the date of ‘creation’ makes sense in the fictional framework only. Moreover, any inquiry about further family members is not understood. The personal history is similarly shallow and inconsistent as of her hobbies: she has a favorite author, but cannot name any title by him. Deviations from this common solution can be found, when the VH is to stand for a real, dead person [Bernsen et al., 2004], and the very application is to introduce the reincarnated real person and his history to the user. The other extreme is feasible when the VH is in a role like a museum guide [Kopp et al., 2003], where his refusal ‘to talk about any personal matters’ sounds to be a natural reaction. But in other applications, where it would be appropriate, we would never know about the family, schooling, living conditions, acquaintances and other experiences of the VH, neither about his favorite food or hobbies. One may argue that that is enough, or even preferred, to remain ‘to the point’ in well-defined task-oriented application like a weather reporter or trainer. However, even in such cases in real life some well-placed reference to the expert’s ‘own life and identity’ breaks the business-like monotonicy of the service, and can contribute to create common ground and build up trust. B. Hayes-Roth endowed her Extempo characters with some own history as part of their ‘anima’ [Hayes-Roth and Doyle, 1998]. From the recent past, we recall a Dutch weather forecast TV reporter who added, when a certain never heard-of Polish town was mentioned as the coldest place in Europe, that this town is special for him as his father was born there. But he could have noted about some other aspects like special food or customs he experienced or knows of from that place. In case of a real fitness trainer’s video, it is remarkable how the task-related talk is interwoven with references to the presenter’s personal experience on where she learnt the exercises, what she found difficult, etc. A VH could use his personal background to generate just some ‘noise-like small talk’ in addition to the task-related conversation, or to relate it to the stage of task completion or difficulty and the reactions from the user, in order to increase the user’s commitment. So for instance, a VT may include not task-related small talk at the beginning or during resting times, or add task-related background information to keep the user motivated during a long and/or difficult exercise.

In order to make a VH ‘personal’, it is not enough to endow him with a ‘personal history’. Some mechanisms should be provided to be able to decide when and what piece of personal information to tell. E.g. to derive if there is something in the personal knowledge of the VH which could be related to the factual, task-oriented information to be told. This may span from simple tasks as discovering dates, names and locations, to the really complex AI task of associative and analogical reasoning.

Finally, the disclosure of the personal information and identity is a manifestation of personality: open, extrovert people (and VHs) may interweave more their story with personal references than introvert ones.

An interesting question is that a VH’s ‘personal history’ may be also adapted to a situation, or a given user (group), not only its conversational style as suggested for robots [Dautenhahn, 2004] and VHs [Ruttkay et al. to appear]. However, consistency within different interaction sessions with the same user (group) should be taken care of.

4.3 Conversational style

There is much to be exploited as of the conversational style of VHs. To begin with, variety should be aimed at in language usage. The variations should be modulated according to the identity of the VH and to the characteristics of the user. For instance, the communicative act ‘greeting’ should be realized differently by the VT, depending on the age of the user, weather addressing him the first time or already acquainted. Within these constraints still remains space for a couple of different utterances to choose from, including special, individual language usage.

The conversational style should also reflect the role and personality of the VH. For instance, if a VT is to be in an assistant role rather than a tutor, more informal language usage is appropriate, than in case of a physiotherapist consultant.

Similar to their ‘perfect and spotless’ embodiment, today’s VHs talk a ‘perfect and spotless’ language. This is contrary to real-life conversation, where people abandon or correct sentences, use pauses and non-speech elements interwoven with words and sentences, as we have experienced, as we have also noticed in studies of conversation in meetings and in talk shows [Heylen and Op den Akker, 2006]. These ‘imperfections’ are not to be seen as weaknesses of real-life speech to be eliminated from the repertoire of VHs. Just the opposite, they do carry subtle information about the person (personality, level of knowledge and expertise in a filed), about the mental processes (e.g. thinking, recalling information) and conversational state (e.g. filled pauses indicate the intention to keep the floor in the conversation), or have functions related to the content, such as a filled pause used in front of a for the user negative answer to decrease the disappointment, or a pause to highlight difficult or important piece of information to come.

‘Imperfect’ language usage is characteristic especially in the case of the VT, where we are currently experimenting with speech elongation and alignment strategies which are beyond the normal customs, but are convenient in explaining rhythmic motions. One problem we have bumped into that TTS engines do not provide access to such ‘beyond normal’ control of timing.

4.4 Humor and laughter

In [Ekman, 2001] the various functions of smiles are discussed. It is generally agreed that natural interaction between humans and virtual humans requires models from which these functions can emerge. For example, in a study on visual cues for feedback the smile turned out to be the strongest cue for affirmative feedback [Granström et al., 2002]. Smiles are important in regulating interactions, but humor is also an important factor. In some environments we discussed (virtual dancer, virtual trainer and virtual conductor) it is quite natural to expect nonverbal humor in the interaction between virtual human and his human partners. In particular when one of the partners in the interaction fails to do things 'right' he or she can recover by doing something funny and unexpected or, the other way around, his or her partner can give someone the opportunity to recover by a humorous nonverbal reaction.

The role of verbal humor, with the aim to design virtual humans that are able to generate and understand verbal humor, in conversations, task-related interactions, meetings, and education, is discussed in [Nijholt, 2007]. Generating and understanding verbal humor and using humor in an appropriate way during an interaction requires natural language and common sense understanding by computers that is too far away from current research achievements in artificial intelligence. However, especially in situations as mentioned above, the combination of limited verbal intelligence, knowledge about the task or the goal of the interaction, and the possibility to use nonverbal means can be employed by a virtual human to generate humorous acts at appropriate moments during an interaction. Adding laughter during such interactions is another issue that needs to be addressed [Trouvain and Schröder, 2004].

4.5 Here and today - situatedness

VHs hardly give the impression that they know about the time and situation they converse in with their user. Some VHs do reflect the time of the day by choosing an appropriate greeting. But much more could be done: keeping track of the day, including holidays, and commenting accordingly, providing 'geographical update' capability when placing a VH-enabled service in a location, endowed may be some social and political information about the place. Imagine a VT who knows that it is today a public holiday in Italy where the given VT is 'active'. Some special words to the user keeping up her exercise scheme on a holiday would be appropriate. But on a tropical summer day, the heat may lead the VT to revise its strategy, remind the user the necessity of drinking, or even shorten the exercises, or suggest doing it in the morning.

The identity of the user may be a source of further situatedness. As a minimum, a VH should 'remember' earlier encounters with the user. Asking the name or telling the same piece of small talk to the same person each time is disappointing. But how nice it sounds if a VT refers to yesterday's performance, knows of the user's religion does not allowing her to do exercises on Saturday, greets her specially on her birthday.

Finally, in order to perceive a VH as 'present', the VH must have means to gather information about the user and react to it. To begin with, the mere presence of the user and her identity should be detected, and her task-related performance should be monitored. But think of a real trainer or tutor, who would very likely comment on changes like not wearing glasses, change in hair style, being sunburn or showing signs of a cold. A Virtual Trainer could do similar comments.

5 Discussion

We have argued that there are dimensions to be still exploited to turn VHs more life-like, entertaining and in cases effective. We discussed the potentials of:

- making VHs look more individual and imperfect, may be configured to a given user's preferences;
- endowing VHs with identity and personal history;
- grounding VHs to the geographical and sociological place and time of the application being used;
- taking care of styled and natural conversation with phenomena of 'imperfections' reminiscent in real life.

The above features do not require, first of all, further perfection of single or multi-modal communication, but they do pose challenges on modeling mental capabilities like associative storytelling or require further socio-psychological studies of the nature and effect of social conversation in task-related situations. What is needed, for several of the above enrichments, is multi-signal perception, first of all, vision, of the conversant of a VH.

This, however, leads to a more general issue: the relationship of VHs to real ones. In our discussion we recalled examples from real human practice, which would be beneficial to endow the communicative capabilities of VHs. Due to the novelty of the field and the many parameters influencing the judgment of a VH, we cannot make conclusions as of the following questions:

- Which (as many as possible, ideally all?) phenomena of real human behaviors should be reproduced by VHs?
- How to exploit the 'beyond human' possibilities of VHs, both in perception and mental capabilities?

For both issues, dedicated evaluation studies are needed to put together a huge jigsaw image. It is clear already that the objective to engage the user in an activity and to perform a task well and efficiently may require different VH design, along several dimensions. Also, the application context (real-fictional) puts the user in different frame of mind to judge the VH, On the other hand, even less is known of the judgments of non-human capabilities of VHs. For example, it has turned out that a VH could 'read from the eye' of the user better than most of the people are capable of. What to do with such a super-human power of a VH? Or, another example is the reasoning speed and capability of a VH: do people take it as natural (from a VH) that he can recall multiple telephone books? Or should he 'fake' the human limitation of recalling data in a register? How to get away with shallow, or not deep/complete enough, models?

Finally, we mention arts as an exploited source of design principles for VHs. Arts, especially (portrait) painting, animation and theater can provide elicit knowledge about human communication, facial and gesture expressions, reflection of personality and emotions in nonverbal signals and speech. Further on, some principles of realizing certain effects, may be by non-realistic features, can be beneficial for enhanced expressivity. The cartoonish exaggerated effects in gesturing can underline e.g. personality characteristics present in real-life speech. One step further, one envisions the next-generation, ideal VHs as seamlessly integrating elements of practices from human conversational behavior, the enhanced interaction and reasoning capabilities of computer technology and the expressivity and aesthetics of arts.

References

[Abaci et al., 2004] T. Abaci, R. de Bondeli, J. Ciger, M. Clavien, F. Erol, M. Gutierrez, S. Noverraz, O. Renault, F. Vexo and D. Thalmann. Magic Wand and Enigma of the Sphinx. *Computers & Graphics*, 28:4, 477-484, 2004.

[AgentLand, 2006] AgentLand, http://www.agentland.com/

[Badler, 2002] N. Badler. LiveActor: A virtual training environment with reactive embodied agents. Workshop on *Intelligent Human Augmentation and Virtual Environments*, University of North Carolina at Chapel Hill, Oct. 2002.

[Bailenson and Yee, 2005] J. N. Bailenson and N. Yee. Digital Chameleons: Automatic assimilation of nonverbal gestures in immersive virtual environments. *Psychological Science*, 16, 814-819. 2005.

[Baylor and Ebbers, 2003] A. L. S. Baylor and S. Ebbers. The Pedagogical Agent Split-Persona Effect: When Two Agents are Better than One. Paper presented at the *ED-MEDIA*, Honolulu, Hawaii, 2003.

[Bernsen et al., 2004] N.O. Bernsen, M. Charfuela` n, A. Corradini, L. Dybkjær, T., Hansen, S. Kiilerich, M. Kolodnytsky, D. Kupkin and M. Mehta. First prototype of conversational H.C. Andersen. International Working Conference on *Advanced Visual Interfaces (AVI'2004)*, ACM, New York, 458-461, 2004.

[Bickmore and Cassell, 2000] T. Bickmore and J. Cassell: Small Talk and Conversational Storytelling in Embodied Interface Agents. In *Proceedings of the AAAI Fall Symposium on Narrative Intelligence,* 87–92, 2000.

[Bos, 2006] P. Bos, D. Reidsma, Z.M. Ruttkay, and A. Nijholt. Interacting with a Virtual Conductor. In: 5th *International Conference on Entertainment Computing,* LNCS 4161, Springer Verlag, 25-54, 2006.

[Cassell et al., 2000] J. Cassell, J. Sullivan, S. Prevost, and E. Churchill (Eds.), *Embodied Conversational Agents*, MIT Press, 2000.

[Chi et al., 2000] D.M. Chi, M. Costa, L. Zhao and N.I. Badler. The EMOTE Model for Effort and Shape. *Siggraph 2000, Computer Graphics Proceedings,* ACM Press / ACM SIGGRAPH / Addison Wesley Longman, 173-182, 2000.

[Dautenhahn, 2004] K. Dautenhahn. Socially Intelligent Agents in Human Primate Culture. In: *Agent Culture: Human-Agent Interaction in a Multicultural World*, 45-71 (Chapter 3), R. Trappl and S. Payr (Eds.), Lawrence Erlbaum Associates, 2004.

[DiPaola 1991] S. DiPaola. Extending the Range of Facial Types. *Journal of Visualization and Computer Animation*, Vol.2, No. 4, 129-131. 1991.

[Ekman 1989] P. Ekman. The argument and evidence about universals in facial expressions of emotion. In: Wagner, H., Monstead, A. (Eds.): *Handbook of Social Psychology*. John Wiley, Chic ester, 1989, 143-146.

[Ekman, 2001]. P. Ekman. Telling Lies. Clues to deceit in the marketplace, politics, and marriage. New York and London: W.W. Norton and Company 2001 (reissued with a new chapter).

[Egges et al. 2004] A. Egges, T. Molet and N. Magnenat-Thalmann. Personalized Real-Time Idle Motion Synthesis. *Computer Graphics and Applications*, 12th Pacific Conference on (PG'04), 121-130, 2004.

[Granström et al., 2002]. B. Granström, D. House, and M.G. Swerts. Multimodal Feedback Cues in Human-Machine Interactions. In B. Bel and I. Marlien (Eds.), Proceedings of the *Speech Prosody 2002 Conference*, Aixen-Provence: Laboratoire Parole et Langage, 347–350, 2002.

[Gratch and Marsella, 2001] J. Gratch and S. Marsella. Tears and fears: Modeling emotions and emotional behaviors in synthetic agents. In Proceedings of the *Fifth International Conference on Autonomous Agents,* 2001.

[Gratch et al., 2002] J. Gratch, J. Rickel, E. Andre, N. Badler, J. Cassell, and E. Petajan. Creating interactive virtual humans: Some assembly required. *IEEE Intelligent Systems*, 2002.

[Gustafson et al., 2004] J. Gustafson, L. Bell, J. Boye, A. Lindström, and M. Wiren. The NICE Fairy-tale Game System. Proceedings of *SIGdial 04*, Boston, 2004.

[Hartmann et al., 2005] B. Hartmann, M. Mancini, S. Buisine, and C. Pelachaud. Design and Evaluation of Expressive Gesture Synthesis for Embodied Conversational Agents. *AAMAS'05*, Utrecht, July 2005.

[Hayes-Roth and Doyle, 1998] B. Hayes-Roth and P. Doyle. Animate characters. *Autonomous Agents and Multi-Agent Systems*, Volume 1, Number 2, 1998.

[Heylen and Op den Akker, 2006] D. Heylen and R. op den Akker. Investigations into the distribution of backchannels in argumentative multi-party discourse and their functional determinants. Manuscript, HMI Research group, University of Twente, September 2006.

[Isbister et al., 2000] K. Isbister, H. Nakanishi, T. Ishida and C. Nass. Helper agent: Designing an assistant for human-human interaction in a virtual meeting space. In *Proceeding of CHI'2000*, 57-64. 2000.

[Isbister and Nass, 2000] K. Isbister and C. Nass. Consistency of personality in interactive characters: Verbal cues, non-verbal cues, and user characteristics. *International Journal of Human-Computer Studies*, 53(2):251–267. 2000.

[Isbister and Doyle, 2004] K. Isbister and P. Doyle. The Blind Man and the Elephant Revisited: A Multidisciplinary Approach to Evaluating Conversational Agents. In (Z. Ruttkay and C. Pelachaud, Eds.) *From Brows to Trust: Evaluating Embodied Conversational Agents*, Volume 7, Human Computer Interaction series, Kluwer Press. 2004.

[Koda and Maes, 1996] T. Koda and P. Maes. Agents with faces: The effects of personification of agents. Proc. of HCI'96, 1996.

[Liang et al., 2002] L. Liang, H. Chen, Y. Q. Xu and H. Y. Shum, Example-Based Caricature Generation with Exaggeration, *in Proc. 10th Pacific Conf. on Computer Graphics and Applications*, 2002.

[Kopp et al., 2003] S. Kopp, B. Jung, N. Lessmann, and I. Wachsmuth. Max--A Multimodal Assistant in Virtual Reality Construction. *KI-Künstliche Intelligenz* 4/03: 11-17, 2003.

[Mateas and Stern, 2003] M. Mateas and A. Stern. Facade: An Experiment in Building a Fully-Realized Interactive Drama. *Game Developer's Conference:* Game Design Track, San Jose, California.

[Mori 1982] M. Mori. *The Buddha in the Robot.* Charles E. Tuttle Co., 1982.

[Nijholt, 2007] A. Nijholt. Conversational Agents and the Construction of Humorous Acts. Chapter 2 in: *Engineering Approaches to Conversational Informatics*. T. Nishida (Ed.), John Wiley & Sons, Chichester, England, 2007, to appear.

[Noot and Ruttkay, 2004] H. Noot, Zs. Ruttkay: Style in Gesture. In: A. Camurri, G. Volpe (Eds.), *Gesture-Based Communication in Human-Computer Interaction*. LNCS 2915, Springer-Verlag, 2004.

[Payr and Trappl, 2004] S. Payr, R. Trappl (Eds.). *Agent Culture - Human-agent Interaction in a Multicultural World*. Lawrence Erlbaum Associates, 2004.

[Plantec, 2004] P. Plantec. Virtual Humans. *AMACOM*, 2004.

[Poggi et al., 2001] I. Poggi, C. Pelachaud, and B. De Carolis. To display or not to display? Towards the architecture of a reflexive agent. In *Proc. of the 2nd Workshop on Attitude, Personality and Emotions in User-adapted Interaction. User Modeling 2001*, 13-17. 2001.

[Prendinger and Ishizuka, 2001] H. Prendinger and M. Ishizuka. S*ocial role awareness in animated agents*, Proc. of Autonomous Agents Conference, 270-277, Montreal, Canada. 2001.

[Prendinger and Ishizuka, 2004] H. Prendinger and M. Ishizuka (Eds.). *Life-Like Characters. Tools, Affective Functions, and Applications*, Cognitive Technologies Series, Springer, Berlin Heidelberg, 2004.

[Reidsma et al., 2006] D. Reidsma, H. Van Welbergen, R.W. Poppe, P. Bos, and A. Nijholt. Towards Bi-directional Dancing Interaction. In *Proc. of 5th International Conference on Entertainment Computing*, LNCS 4161, Springer-Verlag, 2006, 1-12, 2006.

[Rehm and Andre, 2005] M. Rehm and E. Andre. Catch me if you can -- exploring lying agents in social settings. Proc. of the Int. Conf. on Autonomous Agents and Multiagent Systems, 937-944, 2005.

[Ruttkay and Pelachaud, 2004] Z. Ruttkay and C. Pelachaud (Eds.). *From Brows to Trust. Evaluating Embodied Conversational Agents*. Kluwer's Human-Computer Interaction Series – volume 7, 2004.

[Ruttkay et al., 2006a] Z. M. Ruttkay, D. Reidsma, and A. Nijholt. Human Computing, Virtual Humans and Artificial Imperfections. *ACM-ICMI'06*, 2006.

[Ruttkay et al., 2006b] Zs. Ruttkay, J. Zwiers, H. van Welbergen, and D. Reidsma. Towards a Reactive Virtual Trainer. Proc. of *IVA 2006*, LNAI 4133, Springer-Verlag, 292-303. 2006.

[Ruttkay et al., to appear] Zs. Ruttkay, C. Pelachaud, I., Poggi, and H. Noot. Exercises of Style for Virtual Humans. In: L. Canamero, R. Aylett (Eds.), *Advances in Consciousness Research Series*, John Benjamins Publishing Company, to appear.

[Stronks et al., 2002] B. Stronks, A. Nijholt, P. van der Vet, and D. Heylen. Designing for friendship: Becoming friends with your ECA. In: Proc. *Embodied conversational agents - let's specify and evaluate them!*, 91-97. 2002.

[Thorisson, 1996] K. Thorisson. *Communicative Humanoids: A Computational Model of Psychosocial Skills*. PhD thesis, MIT Media Laboratory, 1996.

[Trouvain and Schröder, 2004] J. Trouvain and M. Schröder. How (Not) to Add Laughter to Synthetic Speech. In: Proc. Workshop on *Affective Dialogue Systems (ADS 04)*, E. André (Ed.), LNAI 3068, Springer Verlag, Kloster Irsee, Germany, 2004, 229-232.

[Van Moppes 2002] V. Van Moppes. *Improving the quality of synthesized speech through mark-up of input text with emotions*. Master Thesis, VU, Amsterdam, 2002.

[Wachsmuth and Kopp, 2002] I. Wachsmuth and S. Kopp. Lifelike Gesture Synthesis and Timing for Conversational Agents. In Wachsmuth & Sowa (eds.), *Gesture and Sign Language in Human-Computer Interaction*, 120-133, LNAI 2298, Springer-Verlag, 2002.

[Walker et al., 1994] J. Walker, L. Sproull, and R. Subramani: Using a Human Face in an Interface, *Proceedings of CHI'94*, 85-91. 1994.

Mixed-Initiative Interfaces to Recognize Regulate and Reflect Programming Styles

Shilpi Rao*, Vive Kumar#, Marek Hatala*, Dragan Gasevic*
#Institute of Information Sciences and Technology, Massey University
63 Wallace St, Room 3D09, Post box 756, Wellington, New Zealand
srao@sfu.ca, V.S.Kumar@Massey.AC.NZ

Abstract

Programming Style refers to the ability to follow code conventions, to engineer code in a disciplined manner, to systematically debug code, to optimize code delivery through appropriate settings in the IDE (Integrated Development Environment), to regulate completion rates and quality of programming tasks, and finally to efficiently collaborate with other programmers and resources. This research investigates whether programming styles of individual programmers can be computationally recognized; If styles can be recognized by the machine, can they then be regulated so that programmers can reflect on their own programming styles; finally, can a mixed-initiative computational mechanism assist programmers to identify good programming styles and repair bad programming habits. The paper presents a real-time architecture called MICE (Mixed-Initiative Coding Environment) that employs a formal computational representation of the theory of Self-Regulated Learning (SRL) as the context for human-initiated and system-initiated interactions. The architecture uses ontologies to model-trace programming styles, employs rules to assist programmers to regulate their programming styles, and engages mixed-initiative scaffolding tactics and strategies to provide feedback.

1 Introduction

We define Programming Style as processes that a programmer adopts to achieve specific programming goals. Programming Style, among other aspects, refers to the ability of programmers

- to follow code conventions[1],
- to engineer code in a disciplined manner [Doherty et al., 2005; Kumar, 2004]
- to systematically debug code,
- to optimize code development and delivery through appropriate settings in the IDE,
- to regulate completion rates and quality of programming tasks, and
- to efficiently collaborate with other programmers and resources.

The abovementioned components mold the blueprint of what we consider as the Programming Style of an individual programmer. This research aims to develop a system to assist programmers regulate their coding styles when they code in Java using an Integrated Development Environment (IDE). The proposed system called MICE, abbreviated for Mixed-Initiative Coding Environment, targets three objectives.

- First, MICE aims to capture programming style components from interactions of programmers when they develop task-specific code. It exploits Model-Tracing techniques to map interaction data to style components. The traced data updates programmers' models accordingly.
- Second, MICE aims to engage programmers in Mixed-Initiative (MI) interactions. Mixed-Initiative strategies enable conversants, in our case, programmers and the MICE system, to contribute appropriate task specific information, when it is best suited, towards mutually negotiated goals [Hearst, 1999]. This means that both MICE and a programmer can share their respective goals with each other as well as initiate a feedback process independent of each other.
- Third, MICE aims to leverage the benefits of the theory of Self-Regulated Learning (SRL) in its feedback mechanism. SRL views learning as an activity that students perform proactively, rather than as a covert event that happens to them in reaction to teaching [Winne, 1997]. MICE embeds an ontological representation of Zimmerman's SRL model, which includes the phases of forethought (planning, task analysis, self-motivation, goal setting), performance (self-monitoring, self-monitoring, self-recording), and

[1] http://java.sun.com/docs/codeconv

self-reflection (self-judgment, self-reaction, self-evaluation, self-satisfaction) [Zimmerman, 2002].

Targeting these objectives, MICE initiates one or more of the following feedback methods at opportune moments:

- Engage the programmer with a pre-defined conversation model [Karen et al., 2002]
- Introduce the programmer to a ready, able, and willing human helper [Brooks et al., 2006; Kumar, 2001]
- Provide the programmer with timely clues and hints [Morris et al., 2006]
- Scaffold the programmer in a guided practice session [Lesgold et al., 1992]
- SRL-specific feedback [Samin, 2004; Shakya, 2005]

We designed the MICE system as an ontology-centric framework consisting of two key ontologies; first, a Programming Style ontology that corresponds to the components of programming styles identified earlier; second, an Interaction ontology that captures interactions of a programmer within the MICE environment. In addition to these two, MICE also avails ontologies such as an ontology of a model of SRL theory, a Learner ontology, and a Time ontology. MICE makes use of these ontologies to profile the interaction data that are captured at run time. MICE then processes these data in a reactive manner and updates the facts in a production rule system. These facts trigger specific rules that initiate feedback processes.

The next section discusses the programming style components. Section 3 briefly discusses the architectures of MICE. Section 4 explains the key ontologies in MICE. Section 5 outlines what we mean by system-initiated feedback and how we plan to employ such feedback in MICE. The final Section of the paper concludes our views and also outlines our plan for future work.

2. Programming Style

In this section we define all the components of programming styles as well as give a research background for each of them. Before we describe them, we first introduce some important presumptions related to programming styles on which we based our research.

The first presumption is that a programmer does not have to adhere to a single recognized programming style. He/She can use one or more styles. Instead of attempting to stereotype the observed styles of a programmer to a particular type, the scope of MICE enables to identify a variety of styles exhibited by a programmer, over a period of time. Importantly, these observed styles are stored in an ontological form and MICE can communicate with the programmer about the changes in his/her programming styles over a period of time across different contexts.

The second presumption is that there is no one good programming style. The best-suited programming style/s may vary from programmer to programmer. The feedback given to programmers now-a-days is limited to syntactic errors and code conventions. Such feedback are mostly summative in nature; that is, they are provided not during code construction but mostly at compile time. Feedback could be given to programmers based on their code design, their code presentation style, and their debugging style in a formative fashion.

We extend the notion of programming style in two main aspects: first, programming style is a process and hence can change from context to context over a longer period of time; second, programming style includes a number of other factors outlined below and a programmer can engage in one or more of these factors that determines his/her programming style.

2.1 Code Conventions

Code Convention is defined as the ability of a programmer to adhere to specific conventions prescribed for a particular language. For example, Java Code Conventions include usage of tabs, indents, blank lines, spaces, alignments, braces, wrapping, naming, file organization, documentation, language construct statements, and imports[2].

Many Java code convention checkers/verifiers are available in public domain[3] as well as in the commercial market[4]. However, almost all these software only provide summative feedback as opposed to MICE's formative mixed-initiative feedback approach. Also, unlike MICE, the conventional checkers and verifiers do not allow programmers to change existing conventions to their liking. MICE allows programmers to negotiate code convention preferences with code built-in convention checkers and verifiers.

2.2 Code Engineering

Code Engineering in MICE is defined as the ability of a programmer to construct code in a disciplined manner, preferably using sound software engineering principles. Normally, code engineering, among others, involves designing code, typing-in or pasting-in language constructs, compiling, version control, code refactoring, and using templates/patterns.

2.2.1 Code Construction

Kumar [2004] and Doherty et al. [2005] discuss a simple tool that recognizes code construction styles of programmers based on compile-time code segments (CT-SEG). Compile-time code segments are code (partial or complete) submitted to the compiler by programmers for verification of correctness. That is, every time a programmer submits code for compilation a version of the code (CT-SEG) is saved, thus enabling the tool to trace the ability of the programmer to incrementally construct code.

A study conducted by Kumar [2004] shows that the number of CT-SEG and the pattern of CT-SEG vary across pro-

[2] http://jalopy.sourceforge.net/existing/links.html

[3] http://www.tiobe.com/jacobe.htm and http://pmd.sourceforge.net/

[4] Page: 2 http://www.jindent.com/

grammers. That is, programmers compile code at varying time intervals for the same task.

In the same study, Kumar [2004] also observed the lines of code (LOC) between compiles. For example, some programmers compiled code at the end of subtasks while progressively increasing the number of lines of code.

The study also showed code construction styles of programmers where they compiled only when the code for the entire task was completed. Also, there are programmers who tend to discard large chunks of code when the code developed so far failed to deliver results at subtask levels.

Yet another code construction style identified by the study shows how programmers develop code using different language constructs across compiles. Constructs from Java's Abstract Syntax Tree[5] such as comments, control structures, function declarations, variables, and so on were identified in each CT-SEG. The study showed that, for the same task, participants of the study, in general, employed a variety of language constructs for programming. Programmers changed the constructs significantly in between compiles when faced the task of debugging a considerably large numbers of errors and warnings. The study showed that a programmer's behavior can be tracked as a function of change in language constructs in specific debugging contexts.

2.2.2 Code Quality

Code is expected to be of good quality. That is, it should be less complex, should have undergone rigorous testing, should have been refactored, and should have been critically analyzed by code experts. MICE enables programmers to reflect on how these factors affect the quality of their code.

By less complexity we mean, easy to read, easy to understand, easy to extend, easy to maintain, easy to test the code, and so on. Further, code can be made more readable and less confusing by removing unreachable methods and redundant fields from the code. Lines of Code (LOC) method and Function Points Analysis (FPA) are commonly accepted complexity determination methods[6] that can be easily incorporated in MICE. Rather than simply presenting the LOC and FPA results, MICE attempts to present this information to the programmer only when the moment is right. For example, the FPA value of a Java method under development along with a commentary on the trend of the programmer to write complex methods can be presented to the programmer when he/she attempts to send code for an official code review.

Testing plays an important role in determining the quality of code. More number of errors can be detected during testing if the test cases are more varied and explore more number of test paths. MICE can remind programmers about the importance of testing and also about their current testing habits.

Refactoring is yet another important aspect of coding. Refactoring is the process of rewriting a computer program or other material to improve its structure or readability, while explicitly preserving its meaning or behavior. In Software Engineering, the term *refactoring* means modifying source code without changing its external behavior[7]. Examples of refactoring include modifying all import statements in all java files when a file is moved from one package to another and changing all references to the class type when a class is renamed. Refactoring tools[8] save much manual work in coding that is necessary, tedious, and time-consuming. However, rather than simply performing refactoring behind-the-scenes, MICE externalizes refactoring outcomes and presents a summary of the same.

In general, it is quite possible to critically and automatically analyze code using tools[9] and suggest good design and style improvements. Rather than simply and passively presenting these suggestions to programmers, MICE attempts to present these suggestions at opportune moments advocated by the theory of SRL.

2.3 Code Debugging

Debugging is an art and is associated closely with code-engineering. However, because of the complexity involved in tracing programmers' debugging tactics and strategies, we treat code debugging outside the scope of code engineering. Typically, programmers employ a range of automated debugging techniques that are listed below.

- delta debugging – automatically narrows down the difference between a passing and a failing run
- program slices – separates the part of a program or program run relevant to the bug
- observing state – uses a debugger to observe the values of variables
- watching state – uses a debugger to watch small parts of the program state to determine if they change during execution
- assertions – uses comparison of observed values with the intended values when observing a program state

Software tools that enable these automated debugging techniques do not capture debugging patterns over a period of time. In MICE, we are interested in observing how well programmers are able to debug code in between compilations. That is, the number of types of errors and warnings produced by the compiler can be stored whenever a programmer submits code for compilation. MICE can track a programmer's errors and warnings across multiple compiles and record whether he/she tries to solve errors and warnings

[5] https://jabstract.dev.java.net/

[6] http://www.verifysoft.com/en_code_complexity_measures.html, http://www.gotdotnet.com/Community/UserSamples/Details.aspx?SampleGuid=791516bd-b0ef-491a-be1e-0d622776197b

[7] http://en.wikipedia.org/wiki/Refactoring

[8] http://jrefactory.sourceforge.net/

[9] http://www.serc.nl/people/florijn/work/designchecking/RevJava.htm

as soon as they appear, or debugs only a select few errors and warnings, or continues coding without correcting them.

From our personal experiences, we recognize that expert programmers tend to develop a range of debugging skills, particularly skills that help them identify specific errors and/or warnings that maximizes their productivity. This research attempts to track errors and warnings resolved by programmers across compiles in an effort to identify the debugging styles of the programmers.

The study conducted by Kumar [2004] indicates various patterns of debugging. Most of the participants in the study tried to eliminate errors as soon as they appeared and completely neglected the warnings. When these errors and warnings were compared with the LOC across various compiles, a pattern that indicated a marked change in LOC was observed. This change in LOC can vary from changing a few lines of code to changing or eliminating a major portion of the code, depending on the programmer's debugging style. MICE records and presents observations on the debugging patterns of programmers, that usually go unnoticed. Further, MICE also presents expert debugging behavior, as case studies, to programmers.

At this time, the design of MICE is restricted to observing and recording the debugging patterns of programmers. The correlational and causational effects that exist between code engineering and code debugging processes of programmers will be explored elsewhere, as part of the first author's thesis.

2.4 Optimal IDE Settings for Coding

The Integrated Development Environment (IDE) plays an important role in programmer productivity. An IDE is an environment that integrates multiple software engineering toolkits and presents the same to the programmer in a single interface. For example, the IntelliJ IDEA IDE[10] integrates and customizes a number of toolkits including project management, appearance, language editor, code compilation, compiler errors, colors and fonts, libraries, debugger, resources, IDE history, templates, plugins, and intention settings. One of the key goals of MICE is to be able to guide programmers towards an optimal IDE setting to suit their individual programming styles based on pre-defined models of IDE settings of experts.

2.5 Regulating Coding Tasks

Programming tasks can be classified across different dimensions. Bloom's revised taxonomy could serve to classify programming tasks in the cognitive dimension. For instance, a programming assignment could include components that explicitly demand students to exhibit their coding skills with respect to *remembering*, *understanding*, *applying*, *analyzing*, *evaluating*, and *creating* language constructs[11]. Leopard Tutor [Kemp et al., 2005] classifies tasks under program readability, program understanding, program tracing, and program debugging. PHelpS [Collins et al., 1997] classifies tasks based on functionalities of the system employed by the Corrections Services of Canada.

Following the footsteps of the Leopard Tutor, MICE encourages programmers to construct task models before they start to code and also to track their progress with respect to the task model. Based on the time estimates provided by the programmers themselves, MICE presents proactive and non-intrusive feedback about the speed of their coding and a probabilistic estimate of when the system expects them to complete the complete the code.

2.6 Collaboration While Coding

Effectively collaborating with colleagues is crucial in extreme programming[12] and other agile software development methodologies[13] but is also important in a normal coding environment. For example, a programmer may casually shout across the room for clarification on a particular type of bug or share code with a chat friend for an informal code review. A number of tools support collaboration in terms of chat, discussion boards, and so on. In our view, these tools support collaboration *passively*. By this we mean that these tools do not actively promote and guide users in appropriate and productive collaboration strategies and tactics. Morris et al. [2006] discuss ways in which software can passively as well as actively promote collaboration. Based on iHelp's model of code collaboration [Brooks et al., 2006], the interaction interface of MICE has been designed so that programmers can share code with each other and critique the same under a guided environment.

3. MICE Architecture

We present MICE's architecture under two categories: functional and technical.

3.1 The Functional Architecture of MICE

The functional architecture describes the flow of functionality in the system. As depicted in Figure 4, the flow starts with programmer interactions in an IDE. The current MICE software uses BlueJ as the IDE. These interactions trigger events to instantiate appropriate elements in the ontologies. Changes in ontologies trigger execution of rules. Specifically, the purpose of MICE rules are threefold:

- rules are used to computationally recognize programming style components;
- rules are used to identify opportunities for system-initiated interaction such as programmers spending too much time debugging a piece of code or programmers consistently failing to construct task models;

[10] http://www.jetbrains.com/idea/

[11] http://rite.ed.qut.edu.au/oz-teacher-net/index.php?module=ContentExpress&func=display&ceid=29

[12] http://en.wikipedia.org/wiki/Extreme_programming

[13] http://en.wikipedia.org/wiki/Agile_software_development

- rules are used to engage programmers in mixed-initiative dialogues [Shakya, 2005] with MICE. For example, the MICE system and the programmer can engage in a well-defined, role-playing conversational model when situation warrants it.

The feedback of MICE are marshaled at real-time to the BlueJ IDE as well as to external systems including iHelp and gStudy. In return, events observed at external systems can be recorded in the interaction ontology. For example, collaboration sessions in iHELP [Brooks et al., 2006] between programmers (via chatting, posting, and program-sharing editor) and gStudy events related to links-creation, highlighting, browsing, and searching can be recorded in the interaction ontology.

The functional architecture also includes a module that accumulates the overall programming skill development, which can then be used to revise the rules. This part of the architecture has not been implemented yet. The double-dotted line bifurcate the system into two parts – the interaction environment that contains the external interfaces for the programmers and the MICE environment that contains the model-tracing components.

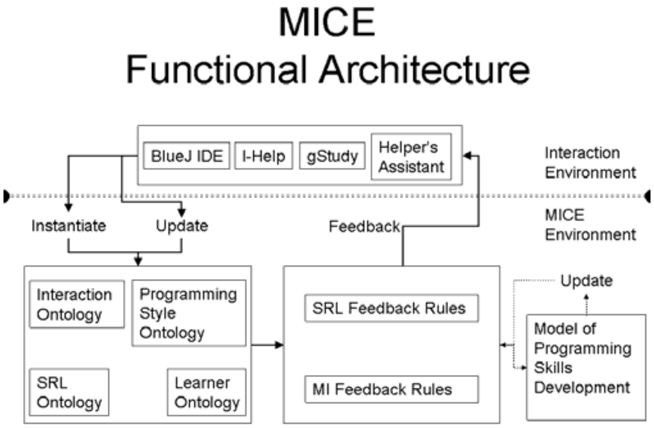


Figure 4. MICE Functional Architecture

3.2 The Technical Architecture of MICE

MICE presently uses BlueJ as the coding environment. Programmer interactions within BlueJ are tracked using BlueJ *extensions*. These *extensions* listen to BlueJ events such as compiling, clicking on a menu item, moving the mouse cursor, and so on. Once an event is triggered by the extension, Jena[14], a Semantic Web toolkit, instantiates or updates the Interaction Ontology at run-time.

Any change in the Interaction Ontology is recognized by MICE as a definite change in the facts list that is stored in the working memory of the JESS inference engine[15]. Any change in the facts list automatically executes rules in JESS. The rules may detect the presence of a particular programming style and as a consequence may instantiate or update the Programming Style Ontology. Further, the rules may also generate system-initiated feedback and send the feedback to external systems (e.g., iHelp, gStudy) via callbacks.

4 Key Ontologies in MICE

The MICE architecture proposes to develop two key ontologies: a Programming Style Ontology and an Interaction Ontology.

4.1 Programming Style Ontology

As mentioned earlier, Programming Style consists of six components, namely, *code convention*, *code engineering*, *code debugging*, *optimal IDE settings for coding*, *regulating coding tasks*, and *collaboration while coding*. The Programming Style Ontology includes these components as top-level classes. Sub-classes are built within each top-level class to capture the presence of style components in programmer interactions. For example, the top-level class *code-convention* contains a subclass for *comment* that recognizes 3 styles of commenting code. They are, a) *comments-goody*, where the code has extensive comments from the programmer, b) *comments-versioning*, where the programmer minimally includes comments about the name of the code, the author, date of creation, last edited date, and so on, and c) *comments-nil*, where the programmer completely ignores to comment the code.

Another programming style recognized by MICE involves the class *code-convention*. Suppose programmers write code in a arbitrary convention but then uses a code verifier to change the convention to a preferred convention. MICE recognizes this style deriving input from *code convention* as well as *code engineering*.

Other programming styles recognized by MICE include compile-for-pop[16], code-till-you-drop[17], hill-climbing-code-construction[18], end-of-days-debugging[19], end-of-world-debugging[20], and SRL-sincere[21]. The first author is in consultation with expert programmers to identify a number of other programming styles.

4.2 Interaction Ontology

As the programmer starts to code, most of his/her interactions with the BlueJ IDE and gStudy are populated in the Interaction Ontology. We are currently in the process of bringing iHELP interactions into the ontology. The interac-

[14] http://jena.sourceforge.net

[15] http://www.jessrules.com

[16] Programmer compiles code at every opportunity such as every time he/she takes a sip of pop

[17] Programmer develops code non-stop for longer terms and compiles only toward the end of coding

[18] Programmer constructs code incrementally with reasonable number of breaks, compiling intermediate code from time to time

[19] Programmer starts to debug only at the very end of task completion

[20] Programmer debugs at every opportunity

[21] Programmer sincerely adheres to SRL and regularly reflects on his/her programming habits

tions are captured in terms of system events. Events that are currently being tracked in BlueJ and populated in the interaction ontology are:

- Compile Event – For every compile event, the Compile class stores a new instance that contains the compile ID, timestamp of the compile, pointers to the next and the previous versions of the compile code, line numbers of the content (code) added/deleted/modified, and LOC while compiling.
- Run Event – Similar to the compile event, when a programmer executes the code, a run event creates an instance of the Run class with the run ID, timestamp of the execution, pointers to the next and the previous versions of the execution, and LOC.
- Errors and Warnings Event – Compilation of code also creates instances of the Error and Warning class that gets instantiated with information about the compile ID and the number of errors and warnings produced by the compile. Further, the event also identifies the types of errors and warnings produced by each compile leading to a cumulative summary of the types of errors and warnings for a programmer.
- Added/Deleted Constructs Event – When a programming language construct is added or deleted an event is triggered that updates and an instance in Added Construct class or Deleted Construct class. Various language constructs at various levels of abstraction , such as comment, if-else control structure, while loop, for loop, do-while loop, template, switch, return, class, throw, functions, expression statements, declaration statements, function definition statements, and so on, can be identified using JavaML[22]. JavaML, an XML-based source code representation for Java programs, is used to uncover the deep structure of the program from a piece of code at various levels of abstraction. The information stored for each event includes the type of construct added/deleted, the associated compile ID, and the content that have been added or deleted.

MICE can receive a number of events from external applications such as gStudy and iHelp. For example, gStudy records the following events to be distributed when a programmer attempts to *read* a programming task, *refers* to various online resources, *comprehends* the task in terms of code design, and *chats* with a fellow programmer to validate the design.

- Link Event – This event is triggered when a programmer creates a link between two sections of content. The content could be a description of the programming task of any multimedia (text, graphics, audio, and video) resource that the programmer is referring to. This event triggers and instantiates data such as link type, link created from, link created to, and so on.
- Highlight Event – This event is triggered when the programmer attempts to highlight any portion of the content. Further, the programmer can attach qualitative clues such as ‘important’, ‘doubt’, ‘to discuss’, and so on.
- Browse Event – This event records links that the programmer has followed while performing the coding task.
- Search Event – This event records when the programmer engages in a search activity within a custom-built search tool.
- Chat Event – This event is recorded when the programmer chats with another programmer from within gStudy’s gChat tool. The gChat tool also enables programmers to use pre-built, semi-constructed, or freely formulated queries. The event records who chatted with who, when, for how long, on what content, using what queries, sent what responses, and so on.
- Posting Event – This event is triggered when the programmer posts an article to the custom-built discussion board or visits the discussion board to read and respond to others’ postings. This event records the type of participation by the programmer, the time of the participation, and so on.

In summary, the interaction ontology is populated with events observed in the interaction environment; further, the programming style ontology is instantiated whenever a predefined style is recognized by the MICE system. The next section discusses the types of programmer-system interactions facilitated by MICE.

5. Mixed-Initiative Interactions

The very premise of MICE is that, unlike other programming environments, the system can monitor, within limits, a programmer’s interactions across various tools and interpret these interactions to various levels of abstractions of programming styles. Further, MICE can proactively engage the programmer in a SRL-induced dialogue.

MICE’s interactions can be passive (programmer-initiated query) or active (system-initiated suggestions to the programmer without the programmer asking for it).

If the programmer is taking too much time to code, or alternatively writing the code and deleting most of it, or consistently getting too many errors and warnings of specific types, or some other pre-defined programming styles detected by the MICE system, MICE can step in and proactively interact with the programmer with appropriate feedback.

The antecedent part of rules that trigger such proactive interaction incorporates variables corresponding to the instantiated programming styles as captured in the Programming

[22] http://www.badros.com/greg/JavaML/

Style Ontology. Also, the antecedent part of rules incorporates variables that correspond to the stages of SRL that a programmer can go through. The consequent part of the rules suggests that the programmer use alternative coding style/s or perform a SRL-specific activity.

MICE is not intrusive. Programmers may or may not accept MICE's proactive feedback. Further, they can question the reasoning (summary of the antecedents) that lead to the system-initiated feedback.

The programmer is encouraged by the system to use the principles of SRL [Winne, 1997]. The programmer is encouraged to plan, self-reflect, and self-evaluate the code, preferably in the same order. For example, reading, taking notes, making flow-chart designs, and engaging in chats prior to coding can be construed of as planning. If the planning stage is not performed by the programmer, as observed by MICE, the system can then prompt (by providing a system-initiated, non-intrusive feedback) the programmer to perform specific planning actions before starting to code. Similarly, observing how often the programmer compiles, executes, and modifies code according to errors can determine if the programmer is engaged in self-reflection and self-evaluating.

A number of other types of feedback interactions can also be initiated by MICE. A programmer might want to chat with a human helper. MICE, in collaboration with the iHelp system, can find a ready, able, and willing helper.

Further, MICE engages learners in interactions specific to data that are captured under various events such as 'compile event', 'run event', 'error and warnings event', and 'added/deleted constructs event'. For example, a programmer's coding style can be pictorially presented as a compile ID Vs. time plot. Similarly, the debugging practices of the programmer can also be pictorially captured and presented. Such depictions also present opportunities for mixed-initiative interactions between programmers and MICE.

The pointers to the next and previous versions of the compiled code enable one to traverse across the programmer's compilation behavior. Any change in the code in between compiles leading to a new set of errors and warnings can be observed and presented to the programmer as causes of new errors and warnings.

One of the simplest types of interaction in MICE is based exclusively on the type of qualitative note or link created by the programmer in gStudy. For instance, a programmer can create a note or link of type *important*, *doubtful*, *contradictory*, and *supporting*. Such qualitative notes allow the MICE system, to identify the stage in the SRL process, as well as the type of feedback the programmer would appreciate.

Depending on the type of tag associated with a highlight, MICE can determine where a programmer is having difficulty or what of the programmer considers as important. This information can be used to produce feedback based on how many other programmers found similar material difficult or important.

In concluding this section, we can say that MICE is designed to provide real-time system-initiated interactions at all stages of coding including comprehension, design, development, deliberation, testing, and reviewing.

6. Conclusion and Future Work

MICE is designed to provide real-time interactive, system-initiated feedback to programmers based on their programming styles. Further, MICE is also designed to incorporate the principles of SRL as part of its feedback. The feedback from MICE can be programmer-initiated or system-initiated. Further, we plan to incorporate models of mixed-initiative dialogues to enable MICE to engage the programmers in constructive dialogues. We plan to collect data by conducting an empirical study. The study is aimed at the following questions:

- Are programming styles of individual programmers computationally recognizable?
- Can the programming styles recognized by the machine be regulated so that programmers can reflect on their own programming styles?
- Can a mixed-initiative computational mechanism assist programmers to identify good programming styles and repair bad programming habits?

We believe that the study will show the impact of theory-oriented mixed-initiative interactions and interface design in human-computing. Further, the ontological capture of programming styles is seen as a knowledge repository of programming experiences that can be employed in many educational applications of human computing.

Acknowledgments

This research was supported by SSHRC[23]'s Learning Kit project and NSERC[24]'s LORNET project. A number of student researchers contributed to this work, notably, Jurika Shakya, Mayo Jordanov, Liam Doherty, and Max Whitney.

References

[Brooks et al., 2006] Brooks, C., Panesar, R., Greer, J., "Awareness and Collaboration in the iHelp Courses Content Management System, " In Proceedings of the 1st European Conference on Technology Enhanced Learning, Crete, Greece, 2006 (forthcoming).

[Collins et al., 1997] Collins, J., Greer, J., Kumar, V., Mccalla, G., Meager, P., Tkatch, R., Inspectable user models for just-in-time workplace training, The Sixth International Conference on User Modelling (UM'97), Italy, pp. 327-337, 1997.

[Doherty et al., 2005] Doherty L., Shakya J., Jordanov M., Lougheed P., Brokenshire D., Rao S., Kumar V.S., Recognizing Opportunities for Mixed-Initiative Interactions

[23] Social Sciences and Humanities Research Council of Canada

[24] Natural Sciences and Engineering Research Council of Canada

in Novice Programming, AAAI Fall Symposium on Mixed-Initiative Problem-Solving Assistants, 2005.

[Karen et al., 2002] Karen, L. and Myers, W. and Tyson, M. and Wolverton, M. J. and Jarvis, P. A. and Lee, T. J. and desJardins, M., PASSAT: A User-centric Planning Framework, Proceedings of the 3rd International NASA Workshop on Planning and Scheduling for Space, 2002.

[Hearst, 1999] Hearst, M.A., "Mixed-Initiative Interaction" IEEE Intelligent Systems, vol. 14, no. 5, pp. 14-23, 1999.

[Kemp et al., 2005] Kemp, R., Todd, E., Krsinich, R., Testing the Effectiveness of the Leopard Tutor under Experimental Conditions, Proceedings of the 12th International Conference on Artificial Intelligence in Education (poster), pp. 839-841, 2005.

[Kumar, 2004] Kumar, V.S., An instrument for providing formative feedback to novice programmers, In Proceeding of the Annual Meeting of American Educational Research Association (AERA), Division I – Education in the professions, Paper session –Relationship between teaching and learning (13.032), 71, San Diego, CA, USA, 2004.

[Kumar, 2001] Kumar, V., Helping the Helper in Peer Help Networks, PhD Thesis, University of Saskachewan, Canada, 2001.

[Lesgold et al., 1992] Lesgold, A., Lajoie, S., Bunzo M., Eggan, G., SHERLOCK: A coached practice environment for an electronics troubleshooting job. Computer Aided Instruction and Intelligent Tutoring Systems, USA, pp. 201-238, 1992.

[Samin, 2004] Samin B., "Effects of Self-Regulated Learning in Programming", Masters Thesis, Simon Fraser University – Surrey campus, Canada, 2004.

[Shakya, 2005] Shakya, J., Knowledge Engineering and Knowledge Dissemination in a Mixed-Initiative Ontological Framework, MSc Thesis, Simon Fraser University, Canada, 2005

[Zimmerman, 2002] Zimmerman, B. J., "Becoming a self-regulated learner: An overview," Theory into Practice, vol. 41, 2, pp. 64-72, 2002.

[Winne, 1997] Winne, P. H. Experimenting to bootstrap self-regulated learning. *Journal of Educational Psychology*, *89*, 397-410, 1997.

[Morris et al., 2006] Morris R., Church H., Hadwin A.F, Gress C.L., Winne P.H. *The Use of Roles, Scripts, and Prompts to Support CSCL in gStudy*. Poster and paper presented at the Annual Meeting of the Canadian Society for the Study of Education, London, ON, Canada, 2006.

Smart Environments for Collaborative Design, Implementation, and Interpretation of Scientific Experiments

Paul van der Vet[1], Olga Kulyk[1], Ingo Wassink[1], Wim Fikkert[1], Han Rauwerda[2], Betsy van Dijk[1], Gerrit van der Veer[1,3], Timo Breit[2], Anton Nijholt[1]

[1] Human Media Interaction Group
University of Twente, The Netherlands
biorange@ewi.utwente.nl

[2] MicroArray Department/Integrative Bioinformatics Unit
University of Amsterdam, The Netherlands
{rauwerda, breit}@science.uva.nl

[3] Human-Computer Interaction
Open University, The Netherlands
gerrit.vanderveer@ou.nl

Abstract

Ambient intelligence promises to enable humans to smoothly interact with their environment, mediated by computer technology. In the literature on ambient intelligence, empirical scientists are not often mentioned. Yet they form an interesting target group for this technology. In this position paper, we describe a project aimed at realising an ambient intelligence environment for face-to-face meetings of researchers with different academic backgrounds involved in molecular biology "omics" experiments. In particular, microarray experiments are a focus of attention because these experiments require multidisciplinary collaboration for their design, analysis, and interpretation. Such an environment is characterised by a high degree of complexity that has to be mitigated by ambient intelligence technology. By experimenting in a real-life setting, we will learn more about life scientists as a user group.

1 Introduction

In visions of future computing, humans are surrounded and supported by smart environments and smart objects that are attentive and pro-active. The environments use their sensors to observe and their intelligence to interpret the activities of their inhabitants and provide support. Ubiquitous computing, ambient intelligence, and pervasive computing are among the names that are used in the literature to refer to this vision. Depending on the domain and the users or inhabitants of these environments we can also speak of smart offices, smart home environments, smart meeting rooms, or smart public environments. Some of the environments are task-oriented, e.g., they aim at providing technology that support efficient meetings or problem-solving sessions, while others aim at supporting home or leisure activities. While currently, due to the possibility of commercial home applications, much emphasis is on sensor-equipped physical environments, we also see interest in virtual environments made up from distributed and connected physical environments. Clearly, one impetus for research in this latter direction came from the development of teleconferencing systems. Another impetus came from developments in the area of computer supported collaborative work (CSCW). Originally this work assumed a rather restricted way of communication between users. For example, the 'Coordinator' system introduced by Winograd [Medina-Mora *et al.*, 1992; Winograd, 1987] to coordinate the communication between collaborators has been called "fascist software" [Spinoza *et al.*, 1995]. This qualification is given because 'Coordinator' is some kind of management surveillance software rather than a system that stimulates cooperation and joint problem solving. However, in more recent years these CSCW environments have developed into so-called Future Workspaces [Fernando *et al.*, 2003]. This development is due to the ability to capture more aspects of human verbal and nonverbal communication behaviour and due to advancements in artificial intelligence, allowing us not only to represent and use domain knowledge, but also to reason about domain knowledge. Apart from supporting, in a global way, issues such as workflow systems, design practices and brain storming sessions, these 'spaces' or environments are meant to provide users with mixed reality cooperation and support. That is, virtual environments are created in which scientists, designers, and technology advisers cooperate while not necessarily being present in the same physical environment and manipulate objects and tools that are both virtual and physical. Joint virtual workspaces allowing access from remote places and offering tools for designers and scientists to design and experiment are the future workspaces.

They may be the future, but when we look at current research practices, there still is a rather large distance between, on the one hand, research on ambient intelligence and smart environments, and, on the other hand, research on future workspaces. Rather independent from these points of view there is the development of ambient intelligence and smart environment technology that can be used in all kinds of smart environments, whether they are inhabited by family members or by collaborating scientists. This includes the development of sensor technology, computer vision, multimodal interaction systems, artificial intelligence, and multimedia presentation technologies. Maybe more interesting are the views expressed in [Pantic *et al.*, 2006] on 'human computing'. As mentioned in this paper, "The key to human computing and anticipatory interfaces is the ease of use, in this case the ability to unobtrusively sense certain behavioural cues of the users and to adapt automatically to his or her typical behavioural patterns and the context in which he or she acts." That is, we need to focus on human behaviour and (joint) activities in smart environments, rather than focussing on intelligent devices (isolated gadgets) and we need to change from a function-oriented view of an environment to a user's goal oriented view [Hellenschmidt and Wichert, 2005].

We are interested in human computing for (life) scientists based on the behaviour of individual scientists and group processes of co-operating scientists. In this paper, we aim to discuss some of the key issues involved in adapting developments in human computing for use in the context of empirical science. Empirical scientists are not often mentioned in the literature on smart environments and ambient intelligence. Yet they form an interesting target group because preliminary studies suggest they differ in certain respects from better studied groups like gamers, patients, and home residents. In particular, scientists seem to prefer to remain in full control.

This paper is organised as follows. We first introduce the habitat of empirical scientists. We then turn to scientific collaborative environments and discuss how workflows may support collaboration within a multidisciplinary team. Part of our work is concerned with the e-BioLab, an environment developed at the University of Amsterdam. We further discuss ways of interacting in the e-BioLab. We round off with a discussion.

2 Ambient Intelligence for Science

We will take molecular biology as an example here. Molecular biology has been the subject of a famous ethnographic study by Latour and Woolgar [1979]. Molecular biologists study the chemistry of life or, more precisely, chemical interactions in and of living cells. They experiment with living organisms (*in-vivo*) and with living cells or material that has been extracted from cells or synthesised (*in-vitro*). Since the time of Latour and Woolgar, the explosion of digital resources (databases and programs) has made a third type of experiment possible, nicknamed *in-silico* or dry-lab. For contrast, the *in-vivo* and *in-vitro* experiments are now also collectively known as wet-lab experiments. A large part of the molecular biologist's work consists of designing experiments and interpreting their results, in both cases heavily aided by the published literature. Living cells are incredibly complex [Papin *et al.*, 2005]. They are studied with the help of modern "omics" technologies that allow large-scale, high-throughput experiments to generate data at a massive scale. The biologist's task of making sense of these data would be infeasible without appropriate software tools.

Roughly, scientific activity of molecular biologists takes place in three different contexts: in the lab, at the desk, and in meetings. All three contexts may profit from ambient intelligence techniques, making scientific research more efficient, effective, and pleasant. In all three contexts, situation awareness implies at least some awareness of the scientific task at hand. This is a challenge because the steps involved in scientific discovery are only to some extent repetitive. It may turn out that a scenario evolves as the discovery process unfolds. We briefly elaborate on the potential benefits of ambient intelligence for each of these contexts.

Lab apparatus is increasingly equipped with sensors and actuators. Many of these devices can communicate with each other and with a base station because they are derived from designs for hard-to-reach or dangerous situations. The tasks such devices can perform are often fixed and they can only obey a few simple commands from the base station. Situation awareness can be achieved by making these devices responsive, enabling two-way communication, and by allowing interaction with lab personnel.

The typical scientist's desktop has a computer with high-speed connections to local servers and the Internet. These systems are still very much classical PCs with some scientific software installed that, however, still falls short of the scientific discovery environment proposed by De Jong and Rip [1997] some ten years ago. The current desktop machine is ill-equipped for high-definition visualisations, interaction with visualisations, and similar multimedia tasks. It requires near-prohibitive overhead to operate it. A desktop PC is, in fact, the wrong tool for much scientific work. Recently, progress has been achieved in packaging recurring task sequences in a single environment. For example, in bioinformatics the Taverna workflow tool [Oinn *et al.*, 2004; Oinn *et al.*, 2002] can in principle perform all computer tasks involved in an *in-silico* experiment. In a similar vein, the Problem Solving Environment (PSE) of the VL-e project [Zhao *et al.*, 2005] packages calls to programs, possibly over a Grid, and information exchange between heterogeneous, distributed computers. Taverna workflows and PSEs resemble scenarios in Crowley's sense, "a description of possible actions or events in the future [...]" [Crowley, 2006]. At the same time, they are also autistic in Crowley's sense: once started, they run to completion. Turning a desktop PC into a scientist's assistant will take a major redesign of both aspects of interaction: the ways the user operates the system and the ways in which the system can convey information to the user.

Meetings, the third context, have been the subject of a lot of research [Rienks *et al.*, 2006]. For example, the AMI project [Nijholt *et al.*, 2006] and its successor project AMIDA investigate fundamental and practical issues one encounters in situation-aware meeting support tools. Even though current proposals do not address the practices and needs of scientists, they form a good starting point for situation-aware support for scientific meetings. AMI, for example, investigated meetings of a multidisciplinary team involved in a creative activity, the design of a remote control. Distributed participants can meet in a virtual meeting room in which the design can sit on the (virtual) table. It is easily imagined that instead of a remote control, a representation of an experiment sits on the table for all participants to see and manipulate.

Our own research is conducted in the framework of the BioRange[1] project, a large, national project aimed at strengthening the bioinformatics infrastructure of The Netherlands. We concentrate on enhancing the exploration of bioinformatics resources through user-centred design, resulting in enriched interactions.

In a part of our project, we focus on face-to-face meetings that serve the purpose of interpreting the results of a particular class of molecular biology experiment, namely microarray experiments. Microarray experiments are high-tech experiments aimed at finding out the expression levels of typically a large number of genes, either absolutely or relative to expression under different circumstances. The experiments involve many sources of noise and the interpretation of the results is far from straightforward [Stekel, 2003]. Nevertheless, there are stakes involved. For example, breast cancer treatment can currently be based on the result of a microarray experiment [Van 't Veer *et al.*, 2002]. In the experiment itself and its interpretation, practitioners from various disciplines are involved: microarray experts, biologists, bioinformaticians, and statisticians. The MAD/IBU group of the University of Amsterdam is building the e-BioLab, a meeting room equipped with a large display, electronic, interactive whiteboards, and other devices [Rauwerda *et al.*, 2006], see Figure 1. Its aim is to facilitate meetings of the various professions involved in experiments that require multidisciplinary collaboration for their design and interpretation, such as a microarray experiment.

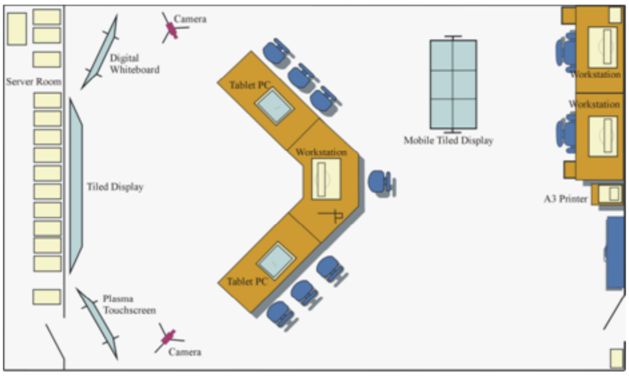


Figure 1. A first design overview of the e-BioLab.

The e-BioLab environment is characterised by a high degree of complexity. This complexity derives from four factors. First, the meeting participants come from different disciplines and they attempt to understand each other. Second, a microarray experiment itself is complex, as are the procedures to clean the data and to validate the results. Third, to molecular biologists it is a new task to find biological meaning in the diverse, multidimensional and huge (whole-genome) datasets. Methodology for inference of biological models from "omics" data is still in its infancy. Fourth, the devices in the meeting room have to be operated. A smart e-BioLab needs attentive and proactive interfaces to mitigate the complexity of the meeting environment.

Put briefly, we want to contribute to the design of the e-BioLab, and in particular to the interactions of the users and the devices. We are not only interested in the meeting aspect, although it is an important focus of our research. But we imagine that, as prices of large, high-resolution displays drop, these displays and the associated interactions will also find their way into the lab and the scientists' workrooms.

3 Scientific collaborative environments and workflows

Work on smart, supportive environments has been reported in the literature. The environment itself is called by different names, depending on the aspect one wants to emphasise: for example, war room (enabling extreme collaboration [Gloria, 2002] or for managing crisis situations [Sharma *et al.*, 2003]), collaborative interactive environment [Borchers, 2006], ubiquitous computing room [Brad *et al.*, 2002], multi-sensor meeting room [McCowan *et al.*, 2003], among many others. The use of large displays to support meeting participants has itself been the subject of a strand in the literature [Borchers, 2006; Fitzmaurice *et al.*, 2005; Huang, 2006; Rogers and Lindley, 2004]. Much of this work is relevant but has to be adapted to the specific needs of the users of the e-BioLab: molecular biologists, microarray experts, bioinformaticians, and statisticians. As was found for scientists in general by Dunbar [1995], the practitioners of the various disciplines involved in our research bring with them a rich and often implicit background knowledge.

As in any user-centred approach, user studies and task analysis are a core activity [Bartlett and Toms, 2005; Homa *et al.*, 2004; Kulyk *et al.*, 2006; Van Welie and Van der Veer, 2003]. Recently, we conducted an empirical user study to explore working practices and experiences of users from different bioinformatics sub-domains and with different levels of expertise [Kulyk and Wassink, 2006]. We aim to identify, among other things, the key aspects and user requirements for a scientific collaborative environment. Our respondents mention the advantages of large displays for multiple visualisations but at the same time stress the danger of overwhelming the viewer. They strongly prefer to meet face-to-face, and they tend to forget discussion points and decisions of previous meetings. This is corroborated in other research for general users [McCowan *et al.*, 2003; Nijholt *et al.*, 2006; Rienks *et al.*, 2006] and for scientific teams [Dunbar, 1995, 1997]. Our results are preliminary and more work has to be done to obtain a comprehensive picture. In

[1] http://www.nbic.nl/biorange/

particular, we aim to build a fairly detailed and complete task model of a microarray experiment.

Molecular biology is a highly visual discipline, as any textbook will testify [Alberts *et al.*, 2002; Campbell and Heyer, 2006; Lewin, 2006; Lodish *et al.*, 2004]. In interpreting a microarray experiment in the e-BioLab, results of the experiment itself and of statistical operations on the data can be displayed in the form of visualisations on the large display, as in the example on Figure 2. Moreover, in a multidisciplinary set-up a large display connected to high-performance computing facilities could be used to construct models of biological mechanisms, perform *in-silico* experimentation with these and adapt the models after interpretation of the results. The large display will frequently be split in a number of separate displays. Additionally, other devices in the room can be allocated display tasks. The visualisations on the various displays are obviously related in the sense that they refer to the same experiment, but it will not always be evident what the precise relation is. To prevent users from getting lost, visual aids will have to identify the relations between the various sub-screens.

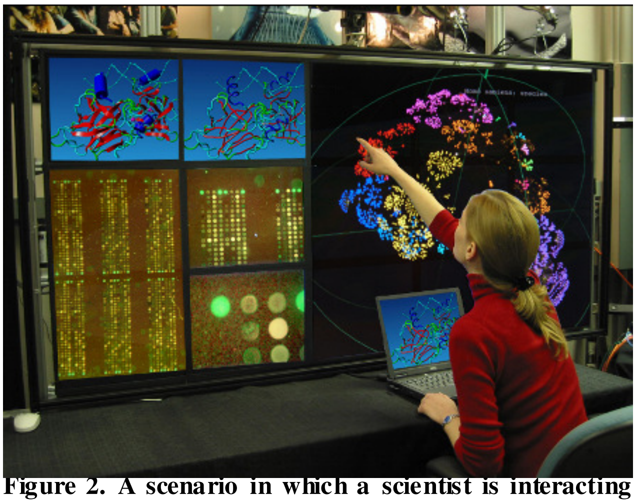


Figure 2. A scenario in which a scientist is interacting with multiple visualizations.

The visualisations may be so closely related that a change in a visualisation on one display will have to be propagated to related visualisations on other displays in a manner pioneered by the Spotfire system[2]. In our case, however, the propagation is far more complex. For example, one display may reveal a number of distinct clusters of gene expression profiles. These clusters then are analyzed on their enrichment with regard to certain pathways. These pathways are visualized on another area of the large display while the up and down regulation and the occurrence of the genes in other clusters is marked in these visualisations. Another example is the design of microarray experimentation. It may take a statistician to establish confidence intervals and statistical power of an analysis. However, only molecular biologists and microarray experts can assess whether it is experimentally possible in the wet-lab to increase statistical power or to avoid confounding by choosing a different experimental setup.

The complexity of multiple displays showing often complex material can, as we stated earlier, be mitigated by employing attentive and proactive interfaces. Such interfaces need to have intelligence built in. At the very least, they need to know a scenario [Crowley, 2006]; in fact, a visualisation of the scenario would be very helpful for the users in any case. Workflow tools have proved useful in modelling business processes [Van der Aalst, 1998]; in our view, they can also be of use for building scenarios. In molecular biology, workflow tools have been proposed for modelling signal pathways in cells [Peleg *et al.*, 2001], for scheduling and supporting tasks in a distributed genomics project [Kochut *et al.*, 2003], and for performing *in-silico* experiments on the Grid [Oinn *et al.*, 2004; Oinn *et al.*, 2002]. As Taverna makes clear, among molecular biologists the data perspective on workflow is dominant. The two other perspectives distinguished by Van der Aalst [1998], the process perspective and the actor perspective, receive less attention. In the task analysis we are performing, all three perspectives receive equal attention.

When using a workflow tool for representing a scenario, we come close to using this tool for designing the experiment and its interpretation. In this sense, an experiment's workflow representation may also support ambient intelligence in the other situations mentioned in the introduction. Different stages in the use of a workflow are distinguished:

- **The design stage**, in which the experiment is designed. This is probably the most difficult stage. There is a close relationship between the design stage and task analysis. The difference between the two is that currently molecular biologists are the actors in the design stage while software designers are the actors in the task analysis. Interaction design can be speeded up if we succeed in bringing these two worlds closer together.
- **The execution stage**, in which the experiment is performed. Part of the experiment can be *in-silico*, in which case the workflow tool can also control the execution of that part. For real-life molecular biology experiments, this stage will have to be subdivided further: there are several steps in the wet-lab, followed by a number of interpretation steps including validation.
- **The archive stage**, in which the conditions of the experiment, the raw results, the settings in the post-processing steps, and the final results are archived. In molecular biology parlance, this is called *provenance* [Goble et al., 2003]. Provenance is important; reproducibility is a major quality control in empirical science. In addition, government bodies want to be able to see such information in medical applications and in drug design. Obviously, the work necessary for this stage has to be done at execution time.

Concerning workflow tool requirements, the BioRange programme has a strong preference for open-source freeware. Further requirements a workflow tool has to fulfil are:

- For **the design stage**, the tool has to enable all three perspectives (data, processes, resources). Hierarchical

[2] www.spotfire.com

modelling is preferred. As workflows become more complicated, validation becomes a concern. The tool therefore has to be based on a formal model and has to incorporate automated validation. Petri nets are a *de facto* standard for formal model of workflows [Van der Aalst, 1998]. YAWL is an example of an open-source freeware workflow tool based on Petri nets [Van der Aalst and Ter Hofstede, 2005]. Finally, the users have to be able to interact naturally with the tool as the design takes shape.

- For **the execution stage**, the *in-silico* parts should run (semi-)automatically. In a large display, such a tool can propagate the results of one part of the display to other parts. Interaction with other programs and remote resources should be automatic and can be configured easily. This kind of technology is researched in Grid projects. For example the Taverna workflow tool [Oinn *et al.*, 2004; Oinn *et al.*, 2002] interoperates smoothly with web resources, using BioMoby or WSDL/SOAP. Interaction with users during execution goes beyond the Grid paradigm, yet is important in many situations. For example, in validating a microarray result, different parameter settings are tried out until the users are satisfied. The number of iterations is not determined beforehand and there has to be a possibility to fiat the result, enabling the workflow tool to move to the next process step. In the wet-lab and at the biologist's desk, interaction with users, sensors and actuators, and human intervention are mandatory. In the wet-lab this tool will eventually coincide with Laboratory Information Management Systems (LIMS) software and will offer a tight integration with the desktop and meeting context. This approach may also facilitate easier implementation of LIMS software in research labs. In the execution stage, it is also important to be able to use the tool for navigation: in which step are we now, what is to follow, and what are the consequences of particular outcomes of the former step. Also it must be possible to capture remarks made in discussions, notions, ideas and hunches and to retrieve these at another time. In other words, the tool must offer the possibility to annotate analyses and biological models. This annotation can be in different forms, for example as text entries in a database or as remarks or sketches drawn on an electronic whiteboard on top of a drawing of a biological model.
- For **the archive stage**, designs, settings, intermediate and final results of the former two stages should be stored. The organisation of this storage is a cause for concern because the amount of data can easily grow enormously. Choices may be necessary about what to keep and what to delete. Such choices are best made at the design stage and the tool has to cater for that. Also, archiving makes no sense if the archived material cannot be retrieved quickly and effectively later. This means that the archive has to conform to standards if they are there. For microarray experiments, for example, the *Minimum Information About a Microarray Experiment* (MIAME) standard[3], even if still under development, is accepted by most institutes and is mandatory upon submission to a large number of scientific journals.

As far as we are aware, no open-source freeware workflow tool meets all these requirements. For example, YAWL aids design by allowing validation of (complex) workflows, but it does not support execution and archiving of the kind required for molecular biology experiments. Taverna does not smoothly interoperate with resources other than BioMoby and WSDL/SOAP web resources. For instance, microarray experts prefer to use the R statistical package[4] for the interpretation and validation of microarray results. Interaction between Taverna and R proved to be cumbersome. No workflow tool we know allows interaction with human users during execution, for example in iterative parameter fitting exercises. For these reasons, it seems we will have to build our own workflow tool, reusing components from YAWL, Taverna and similar tools.

4 Interaction in the scientific environments

There are three modes of interaction in the e-BioLab, human-human interaction, human-display interaction, and inter-system interaction. We briefly discussed the last category implicitly in the former section. Our focus in this paper is on how users may interact with the scientific collaborative environment, and in particular with the large display in the e-BioLab. Due to the size and high resolution of the large display, classical interaction devices will not suffice [Fikkert *et al.*, 2006]. The lab will have to have characteristics of an ambient intelligence environment [Bowman *et al.*, 2004; Fikkert *et al.*, 2006; Jaimes and Sebe, 2005; Oviatt, 1999; Oviatt *et al.*, 2003; Pantic *et al.*, 2006; Tao *et al.*, 2006], making complex systems accessible for a large variety of users, without the need for explicit, tedious, or extensive training. There is no research on ubiquitous computing environments for empirical scientists or, even more specific, our target group composed of microarray experts, bioinformaticians, molecular biologists, and statisticians. We will provide a short overview of issues we think are important, and point out aspects we believe are particularly relevant for our user group.

In conversations, humans can express themselves through numerous modalities that are held to be associated with the human senses [Fikkert *et al.*, 2006]. For example, speech intonation and gestures accompanying an uttered sentence can change the message completely. Multimodal research has focused on systems that combine speech and pointing gestures as input modalities [Oviatt, 1999]. Other modalities such as facial expressions, gaze direction, and body gestures are thought to be mandatory for automatic human behaviour analysis [Ambady and Rosenthal, 1992; Oviatt, 1999; Oviatt *et al.*, 2003; Pantic *et al.*, 2006]. Ambient intelligence derives its knowledge of the current situation from observational clues. In particular, it has to be able to assess the so-called W5+ questions [Pantic *et al.*, 2006] (what is commu-

[3] http://www.mged.org/Workgroups/MIAME/miame.html

[4] http://www.r-project.org/

nicated when, where, why, by/to whom, and how) from observed behaviour. Only multimodal observation can provide the necessary information. Issues to be solved include the following. Which modalities have to be used and when? What is the optimal combination of modalities given the current context? At which level should observed information be fused, at feature or semantic level? It is important to determine how these communicative modalities can be observed in a scene; the interpretation of observed behavioural cues is highly context-based. Understanding behaviour enables a system to fully support and anticipate on its users.

In the e-BioLab, scientists with different scientific backgrounds will use large displays to show their preferred types of visualization in order to discuss progress and results of experiments. Visualisations will be 2D and 3D; for example, protein sequence alignment produces a 2D image but the function of a protein may be better illustrated by its 3D shape. Of the many ways to interact with these visualisations, manual gestures are natural way of expression for many researchers [Buxton and Myers, 1986; Balakrishnan and Hinckley, 2000; Tao *et al.*, 2006; Czwerwinski *et al.*, 2006; Guiard, 1987]. We therefore want to further explore gesture interaction. As users become familiar with gesture interaction, a repertoire of gestures will develop that is in principle new. However, as with most new technology, the repertoire will be rooted in the way life scientists currently use gestures in communication.

Many questions are to be solved for natural gesture interaction to be possible. For example, does the size of the display influence gestures? Are there cultural differences in gesture language, in our case possibly along disciplinary boundaries? In the e-BioLab setup, scientists gesture at each other and at the display; can the two kinds of gesture be distinguished? How can gesture information be fused with other information users may provide, for example through hand-held devices? The detection of gestures is a problem in itself. There are many and diverse techniques for gesture detection [Bowman *et al.*, 2004; Fikkert *et al.*, 2006]; in recent years research has focused on unobtrusive detection, for example using computer vision techniques. Other approaches make use of special devices such as coloured gloves, tethered data gloves, and full-body tracking suits. A current point of research in unobtrusive detection of user gestures when several users are present is how to attribute gestures to the user who made them. The next step, gesture recognition, is an active research topic [Aggarwal and Cai, 1999; Jaimes and Sebe, 2005; Moeslund and Granum, 2001]. Techniques are mostly model-based, using skeletons or geometric shapes, or appearance-based, using motion, texture, or colour information in a scene. The current state of the art does not allow detection and tracking of human hands in multi-party unconstrained environments with dynamic illumination and backgrounds; the e-BioLab is an example of such an environment. Automated gesture recognition is not mature. A representation of the gesture repertoire expressed in a technology-neutral language allows us to quickly adapt to another method of gesture recognition.

Natural interaction will not only be found in dry-lab data analysis settings, but will in all likelihood also be found in wet-lab environments in which media, e.g., augmented reality, can support a laboratory technician in performing her tasks. Recall the three contexts mentioned in Chapter 2 here.

5 Discussion

One of the most fascinating questions in this kind of endeavour is: will it help? Expensive equipment and complex software are brought together in the e-BioLab expecting that molecular biology will profit. User studies and iterative design are employed to improve the initial set-up, but that does not validate the design in the sense that it does not answer the question whether molecular biology has changed or, even better, has improved through this technology. The question immediately raises another question: how do we measure this? It is obvious that the problem space is highly multidimensional. Data can be gathered during meetings of which there are only so many in a year. But that leaves more indirect effects like swifter publication or better molecular biology experiments out of view. Statistical significance is out of reach. We could revert to anecdotal evidence, but we think the point can be strengthened somewhat by performing an analysis in the tradition of sociology of science.

Contemporary science is driven by groups having members of different levels of expertise and various scientific backgrounds [Dunbar, 1995]. Scientist's meetings, if recorded, provide a far more complete record of the evolution of their ideas than other sources of information. That is why our target group, a multidisciplinary scientific team working on microarray experiments, is so interesting. We believe that by experimenting in a real-life setting, the e-BioLab and its users, we can learn more about scientists as a user group. Ambient intelligence for science will profit from this; we expect ambient intelligence in general, too. A natural interface able to decipher and anticipate user activities and desires truly immerses our scientists in their cognitive task and thus truly supports them [Butz *et al.*, 2003]. Human-centred design or, in our case, scientist-centred design is a necessary condition to achieve this.

Acknowledgements

This work is part of the BioRange program of The Netherlands Bioinformatics Center (NBIC), supported by a BSIK grant through The Netherlands Genomics Initiative (NGI).

References

[Aggarwal and Cai, 1999] Aggarwal, J. K. and Cai, Q., Human motion analysis: A review. *Computer Vision and Image Understanding, 73*(3), 428-440, 1999.

[Alberts *et al.*, 2002] Alberts, B., Johnson, A., Lewis, J., Raff, M., Roberts, K., and Walter, P., *Molecular biology of the cell*: Garland Publishing, 2002.

[Ambady and Rosenthal, 1992] Ambady, N. and Rosenthal, R., Thin slices of expressive behavior as predictors of interpersonal consequences: A meta-analysis. *Psychological Bulletin, 111*(2), 256-274, 1992.

[Balakrishnan and Hinckley, 2000] Balakrishnan, R. and Hinckley, K., *Symmetric bimanual interaction.* In Proceedings of the Human Factors in Computing Systems, The Hague, The Netherlands, 2000.

[Bartlett and Toms, 2005] Bartlett, J. and Toms, E., Developing a protocol for bioinformatics analysis: An integrated information behavior and task analysis approach. *Journal of the American Society for Information Science and Technology, 56*(5), 469–482., 2005.

[Borchers, 2006] Borchers, J., *The Aachen media space: Multiple displays in collaborative interactive environments.* In Workshop on Information Visualization and Interaction Techniques for Collaboration across Multiple Displays in conjunction with CHI 2006, Montreal, Canada, 2006.

[Bowman *et al.*, 2004] Bowman, D., Kruijff, E., LaViola, J., and Poupyrev, I., *3D user interfaces: Theory and practice.* Redwood City, USA: Addison Wesley Longman Publishing Co., Inc., 2004.

[Brad *et al.*, 2002] Brad, J., Armando, F., and Terry, W., The interactive workspaces project: Experiences with ubiquitous computing rooms. *IEEE Pervasive Computing, 1*(2), 67-74, 2002.

[Butz *et al.*, 2003] Butz, M., Sigaud, O., and Gérard, P., *Anticipatory behavior in adaptive learning systems, foundations, theories, and systems* (Vol. 2684): Springer, 2003.

[Buxton and Myers, 1986] Buxton, W. and Myers, B., *A study in two-handed input.* In Proceedings of the SIGCHI conference on Human factors in computing systems, Boston, Massachusetts, 1986.

[Campbell and Heyer, 2006] Campbell, A. M. and Heyer, L. J., *Discovering genomics, proteomics & bioinformatics* (Second ed.): Cold Spring Harbor Laboratory Press and Benjamin Cummings, 2006.

[Crowley, 2006] Crowley, J. L., Social perception. *Queue, 4*(6), 34-43, 2006.

[Czerwinski *et al.*, 2006] Czerwinski, M., Robertson, G., Meyers, B., Smith, G., Robbins, D., and Tan, D., *Large display research overview.* In Extended abstracts on Human factors in computing systems of CHI 2006, Montreal, Canada, 2006.

[De Jong and Rip, 1997] De Jong, H. and Rip, A., The computer revolution in science: Steps towards the realization of computer-supported discovery environments. *Artificial Intelligence, 91*(2), 225-256, 1997.

[Dunbar, 1995] Dunbar, K., How scientists really reason: Scientific reasoning in real-world laboratories. In R. J. Sternberg & J. E. Davidson (Eds.), *The nature of insight* (pp. 365--395). Cambridge, MA: MIT Press, 1995.

[Dunbar, 1997] Dunbar, K., How scientists think: On-line creativity and conceptual change in science. *Creative Thought: An Investigation of Conceptual Structures and Processes*, 1997.

[Fernando *et al.*, 2003] Fernando, T., Wilson, J., Dalton, G., Dangelmaier, M., Cros, P.-H., Baudier, Y., et al., *Future workspaces. A strategic roadmap for defining distributed engineering workspaces of the future. Final roadmap.* (No. IST-2001-38346), 2003.

[Fikkert *et al.*, 2006] Fikkert, W., Bierz, T., D'Ambros, M., and Jankun-Kelly, T., Interacting with visualizations. In A. Kerren, A. Ebert & J. Meyer (Eds.), *Human-centered visualization environments.* Springer, 2006.

[Fitzmaurice *et al.*, 2005] Fitzmaurice, G., Khan, A., Kurtenbach, G., and Binks, G., Cinematic meeting facilities using large displays. *IEEE Computer Graphics and Applications, 25*(4), 17-21, 2005.

[Gloria, 2002] Gloria, M., Extreme collaboration. *Communications of the ACM, 45*(6), 89-93, 2002.

[Goble *et al.*, 2003] Goble, C., Stevens, R., Glover, K., Greenhalgh, C., Jennings, C., Pearce, S., et al., *The mygrid project: Services, architectures and demonstrator.* In Proc. of the UK e-Science All Hands Meeting, 2003.

[Guiard, 1987] Guiard, Y., Asymmetric division of labor in human skilled bimanual action: The kinematic chain as a model. *Journal of Motor Behavior, 19*, 486-517, 1987.

[Hellenschmidt and Wichert, 2005] Hellenschmidt, M. and Wichert, R., Goal-oriented assistance in ambient intelligence, *Workshop on Experience Research in Ambient Intelligence.* Eindhoven, The Netherlands, 2005.

[Homa *et al.*, 2004] Homa, J., Ahmed, S., and Thiruvengadam, R., Beyond power: Making bioinformatics tools user-centered. *Comm. of the ACM, 47*(11), 58-63, 2004.

[Huang, 2006] Huang, E. M., *Evaluating the MER display ecology.* In Workshop on Information Visualization and Interaction Techniques for Collaboration across Multiple Displays in conjunction with CHI 2006, Montreal, Canada, 2006.

[Jaimes and Sebe, 2005] Jaimes, A. and Sebe, N., Multimodal human computer interaction: A survey. In *Computer vision in human-computer interaction* (Vol. 3766, pp. 1-15). Heidelberg: Springer Berlin, 2005.

[Kochut *et al.*, 2003] Kochut, K., Arnold, J., Sheth, A., Miller, J., Kraemer, E., Arpinar, B., et al., Intelligen: A distributed workflow system for discovering protein-protein interactions. *Distributed and Parallel Databases, 13*, 43-72, 2003.

[Kulyk *et al.*, 2006] Kulyk, O., Kosara, R., Urquiza, J., and Wassink, I., Human-centered aspects. In A. Kerren, A. Ebert & J. Meyer (Eds.), *Human-centered visualization environments.* Springer, 2006.

[Kulyk and Wassink, 2006] Kulyk, O. and Wassink, I., *Getting to know bioinformaticians: Results of an exploratory user study.* In Workshop on Combining Visualisation and Interaction to Facilitate Scientific Exploration and Discovery in conjunction with British HCI 2006, London, UK, 2006.

[Latour and Woolgar, 1979] Latour, B. and Woolgar, S., *Laboratory life: The social construction of the scientific facts*. Beverly Hills, CA: Sage publications, 1979.

[Lewin, 2006] Lewin, B., *Genes IX*. Sudbury, MA: Jones and Bartlett Publishers, 2006.

[Lodish *et al.*, 2004] Lodish, H., Berk, A., Matsudaira, P., Kaiser, C., Krieger, M., and Scott, M., *Molecular cell biology* (Fifth ed.). New York: W.H.Freeman and Company, 2004.

[McCowan *et al.*, 2003] McCowan, I., Gatica-Perez, D., Bengio, S., and Moore, D., Towards computer understanding of human interactions. In E. Aarts, R. Collier, E. van Loenen & B. de Ruyter (Eds.), *Ambient intelligence (EUSAI 2003)* (pp. 235-251). Springer, 2003.

[Medina-Mora *et al.*, 1992] Medina-Mora, R., Winograd, T., Flores, R., and Flores, F., *The action workflow approach to workflow management technology*. In Proceedings of the Computer-supported cooperative work, Toronto, Canada, 1992.

[Moeslund and Granum, 2001] Moeslund, T. B. and Granum, E., A survey of computer vision-based human motion capture. *Computer Vision and Image Understanding, 81*(3), 231-268, 2001.

[Nijholt *et al.*, 2006] Nijholt, A., Rienks, R. J., Zwiers, J., and Reidsma, D., Online and off-line visualization of meeting information and meeting support. *The Visual Computer*, 2006.

[Oinn *et al.*, 2004] Oinn, T., Addis, M., Ferris, J., Marvin, D., Senger, M., Greenwood, M., et al., Taverna: A tool for the composition and enactment of bioinformatics workflows. *Bioinformatics, 20*(17), 3045-3054, 2004.

[Oinn *et al.*, 2002] Oinn, T., Greenwood, M., Addis, M., Alpdemir, M., Ferris, J., Glover, K., et al., Taverna: Lessons in creating a workflow environment for life sciences. *Concurrency and Computation: Practice & Experience, 18*(10), 1067 - 1100, 2002.

[Oviatt, 1999] Oviatt, S., Ten myths of multimodal interaction. *Communications of the ACM, 42*(11), 74-81, 1999.

[Oviatt *et al.*, 2003] Oviatt, S., Coulston, R., Tomko, S., Xiao, B., Lunsford, R., Wesson, M., et al., *Toward a theory of organized multimodal integration patterns during human-computer interaction*. In Proceedings of Multimodal Interfaces (ICMI), 2003.

[Pantic *et al.*, 2006] Pantic, M., Pentland, A., Nijholt, A., and Huang, T. S., *Human computing and machine understanding of human behavior: A survey*. In Proceedings of Multimodal Interfaces (ICMI), Banff, Canada, 2006.

[Papin *et al.*, 2005] Papin, J. A., Hunter, T., Palsson, B. O., and Subramaniam, S., Reconstruction of cellular signalling networks and analysis of their properties. *Nature Reviews Molecular Cell Biology, 6*(2), 99-111, 2005.

[Peleg *et al.*, 2001] Peleg, M., Yeh, I., and Altman, R. B., Modelling biological processes using workflow and petri net models. *Bioinformatics, 18*(6), 825-837., 2001.

[Rauwerda *et al.*, 2006] Rauwerda, H., Roos, M., Hertzberger, B. O., and Breit, T. M., The promise of a virtual lab in drug discovery. *Drug Discovery Today, 11*(5-6), 228-236, 2006.

[Rienks *et al.*, 2006] Rienks, R., Nijholt, A., and Reidsma, D., Meetings and meeting support in ambient intelligence. In T. Vasilakos, A. & W. Pedrycz (Eds.), *Ambient intelligence, wireless networking, ubiquitous computing* (pp. 205-214). Norwood, USA: Artech House, 2006.

[Rogers and Lindley, 2004]Rogers, Y. and Lindley, S., Collaborating around vertical and horizontal large interactive displays: Which way is best? *Interacting with Computers, 16*(6), 1133, 2004.

[Sharma *et al.*, 2003] Sharma, R., Yeasin, M., Krahnstoever, N., Rauschert, I., Cai, G., Brewer, I., et al., Speech-gesture driven multimodal interfaces for crisis management. *Proceedings of the IEEE, 91*(9), 1327-1353, 2003.

[Spinoza *et al.*, 1995] Spinoza, C., Flores, F., and Dreyfus, H., Disclosing new worlds: Entrepreneurship, democratic action, and the cultivation of solidarity. *Inquiry, 38*, 3-64, 1995.

[Stekel, 2003] Stekel, D., *Microarray bioinformatics*: Cambridge University Press, 2003.

[Tao *et al.*, 2006]Tao, N., Schmidt, G. S., Staadt, O. G., Livingston, M. A., Ball, R., and May, R., *A survey of large high-resolution display technologies, techniques, and applications*. In Proceedings of the IEEE Virtual Reality Conference, 2006.

[Van 't Veer *et al.*, 2002]Van 't Veer, L. J., Dai, H., Van de Vijver, M. J., He, Y. D., Hart, A. A., Mao, M., et al., Gene expression profiling predicts clinical outcome of breast cancer. *Nature, 415*, 530-536, 2002.

[Van der Aalst, 1998] Van der Aalst, W. M. P., The application of petri nets to workflow management. *The journal of circuits, systems and computers, 8*(1), 21-66, 1998.

[Van der Aalst and Ter Hofstede, 2005] Van der Aalst, W. M. P. and Ter Hofstede, A. H. M., Yawl: Yet another workflow language. *Information Systems, 30*, 245-275, 2005.

[Van Welie and Van der Veer, 2003] Van Welie, M. and Van der Veer, G. C., Groupware task analysis. In E. Hollnagel (Ed.), *Handbook of cognitive task design* (pp. 447-476). New Jersey, USA: Lawrence Erlbaum Associates Inc., 2003.

[Winograd, 1987] Winograd, T., A language/action perspective on the design of cooperative work. *Human-Computer Interaction, 3*(1), 3-30, 1987.

[Zhao *et al.*, 2005] Zhao, Z, Belloum, A., Wibisono, A. d. T. F., de Boer, P. T., Sloot, P., and Herzberger, B., *Scientific workflow management: Between generality and applicability*. In Workshop on Grid and Peer-to-Peer based Workflows in conjunction with the 5th International Conference on Quality Software, 2005.

Invited Talk

Making (Virtual) Friends and Influencing (Virtual) People: Building Rapport in Humans and Virtual Humans

Justine Cassell
Departments of Communication Studies & Computer Science
Northwestern University
http://www.soc.northwestern.edu/justine/

Harmony or rapport between people is essential for relationships as diverse as seller-buyer and teacher-learner. In this talk I describe the kinds of verbal behaviors -- such as using the same words and adapting ones accent -- and non-verbal behaviors-- such as attention, positivity, and coordination -- that function together to establish a sense of rapport between two people in conversation. These studies are used as the basis for the implementation of embodied conversational agents (virtual humans) who/that are capable of acting as friends and collaborators. Applications of this work have ranged from direction-giving systems that can be trusted, to virtual peers that help children acquire literacy skills, and systems to help children with autism learn about reciprocal social interaction.

Justine Cassell is a full professor in the departments of Communication Studies and Computer Science at Northwestern University and the director of the Center for Technology and Social Behavior. Before coming to Northwestern, Cassell was a tenured professor at the MIT Media Lab where she directed the Gesture and Narrative Language Research Group. In 2001, Cassell was awarded the Edgerton Faculty Award at MIT.

Cassell holds undergraduate degrees in Comparative Literature from Dartmouth and in Lettres Modernes from the Universite de Besançon (France). She holds a M.Phil in Linguistics from the University of Edinburgh (Scotland) and a double Ph.D. from the University of Chicago in Linguistics and Psychology. After having spent ten years studying verbal and non-verbal aspects of human communication through microanalysis of videotaped data she began to bring her knowledge of human conversation to the design of computational systems.

Cassell's research concentrates on better understanding everyday kinds of conversation and narrative as practiced by children and adults, and on building computational systems that simulate, mediate, and facilitate those everyday kinds of talk. These technologies, such as Embodied Conversational Agents, Story Listening Systems, and Online Communities, in turn allow her to study the nature of human interaction with and through technology.

Evaluating the Future of HCI: Challenges for the Evaluation of Emerging Applications

Ronald Poppe and Rutger Rienks
University of Twente
Human Media Interaction Group
{poppe,rienks}@ewi.utwente.nl

Abstract

Current evaluation methods are inappropriate for emerging HCI applications. In this paper, we give three examples of these applications and show that traditional evaluation methods fail. We identify trends in HCI development and discuss the issues that arise with evaluation. We aim at achieving increased awareness that evaluation too has to evolve in order to support the emerging trends in HCI systems.

1 Introduction

Human-Computer Interaction (HCI) is concerned with the research into, and design and implementation of systems that allow human users to interact with them. Traditionally, the goal of HCI systems is to aid human users in performing an explicit or implicit task. Currently, there is a shift in emphasis towards interfaces that are not task-oriented but rather stress the beauty, surprise, diversion or intimacy of a system [Alben, 1996; Gaver and Martin, 2000].

A vast body of literature deals with evaluation of traditional HCI systems. These evaluation methods are widely used. However, given the new directions of HCI, it is unlikely that these evaluation methods are appropriate.

In this paper, we outline new trends in HCI systems in Section 2. Section 3 presents three examples that illustrate the need for new evaluation methods. In Section 4, we discuss common evaluation methods, argue why these are inappropriate and identify challenges for evaluation of many emerging HCI systems.

2 HCI systems

2.1 Traditional HCI systems

Traditional HCI systems allow human users to input commands using keyboards, mice or touch screens (e.g. ATM machines, web browsers, online reservation systems). These input devices are reliable in the sense that they are unambiguous. Traditionally, systems are single-user, task-oriented and the place and manner in which the interaction takes place are largely determined by the projected task and expected users. This allows system designers to specify the syntax and style of the interaction. Since both input and output interfaces are physical, an explicit dialogue between the user and the computer can be established.

2.2 Emerging HCI systems

Emerging HCI systems and environments have a tendency to become *multi-modal* and *embedded* and thereby allowing people to interact with them in natural ways. In some cases, the design of computer interfaces is merging with the design of everyday appliances where they should facilitate tasks historically outside the normal range of human-computer interaction. Instead of making computer interfaces for people, people have started to make people interfaces for computers [Coen, 1998].

The nature of applications is changing. Looking beyond traditional productivity-oriented workplace technologies where performance is a key objective, HCI is increasingly considering applications for everyday life. HCI interface design now encompasses leisure, play, culture and art. Compared to traditional HCI systems, we can identify four main trends in HCI systems:

1. **New sensing possibilities** New sensing technologies allow for the design of interfaces that go beyond the traditional keyboard and mouse. Automatic speech recognition is common in many telephone applications. The current state of video tracking allows not only for localization of human users, but also to detect their actions, identity and facial expressions [Pantic *et al.*, 2006]. This opens up possibilities to make interfaces more natural. Humans will be able to interact in ways that are intuitive. However, this comes at a cost of having to reconsider the syntax of the application. When using speech or gestures, the vocabulary is almost infinite. Moreover, many of the ‘behaviors’ that we can recognize, must be interpreted in relation to the context. Context aware applications employ a broad range of sensors such as electronic tags, light sensing and physiological sensing. However, integration and the subsequent interpretation of these signals is hard, and context aware systems are likely to consider contexts differently than users do [Intille *et al.*, 2003]. Related to the use of a multiplicity of sensors is the trend that sensors are moving to the background [Streitz and Nixon, 2005]. This moves interfaces away from the object-oriented approach that is traditionally considered [Nielsen, 1993a]. This trend has large

implications for interaction design since it restricts the traditional dialog-oriented way of interaction, and effort must be paid to the design of implicit interactions [Ju and Leifer, to appear].

2. **Shift in initiative** Traditional HCI systems embrace the explicit way in which the dialog with the user is maintained. Consequently, these systems are responsive in nature. Nowadays, pro-active systems are more common. Some HCI systems even aim at fulfilling the role of social actor or companion. Ju and Leifer [to appear] define an initiative dimension in their framework for classifying implicit interactions. They state that, when regarded more generally, there is direct manipulation at the one end, and autonomy at the other. They argue that, for HCI, neither of these states are appropriate. Instead, the interaction is likely to be mixed-initiative. This implies that there must be a way to coordinate the interaction, which should be the focus of interaction design.

3. **Diversifying physical interfaces** The physical forms of interfaces are diversifying [Benford *et al.*, 2005], as was foreseen by Weiser [1991]. One movement is to make interfaces bigger, such as immersive displays and interactive billboards. Another movement is to make interfaces smaller, such as wearable and embedded displays. This last movement is largely motivated by the popularity of mobile devices. The market for mobile phones is still growing, and so is the number of applications. With the increased connectivity and bandwidth, it is possible that people interact remotely with the same application. The trend of diversifying physical interfaces is most visible for general purpose desktop computers. These are increasingly often replaced by more purpose-designed and specialized appliances [Benford *et al.*, 2005].

4. **Shift in application purpose** There is a shift in application purpose for HCI systems. This shift is partly a consequence of new technology, and partly motivates the development of technology. Whereas traditional systems are, in general, task-based, new applications are more focussed on everyday life [Benford *et al.*, 2005], thus on the user. User Experience (UX), although associated with a wide variety of meanings [Forlizzi and Battarbee, 2004], can be seen as the countermovement of the dominant task and work related 'usability' paradigm.

   UX is a consequence of a user's internal state (e.g. predispositions, expectations, needs, motivation and mood). The literature on UX reveals three major perspectives [Hassenzahl and Tractinsky, 2006]: human needs beyond the instrumental; affective and emotional aspects of interaction; and the nature of experience. Hassenzahl and Sandweg [2004] argue that future HCI must be concerned about the pragmatic aspects of interactive products as well as about hedonic aspects, such as stimulation (personal growth, increase of knowledge and skills) identification (self-expression, interaction with relevant others) and evocation (self maintenance, memory). The task is no longer the goal, but rather the interaction itself (e.g. Reidsma *et al.* [2006]).

   Typical UX applications are focussed on leisure, play, culture and art. Consequently, this focus affects the interface. Factors as pleasure, aesthetics, expressiveness and creativity play an increasingly important role in the design of both interface and interaction. Video games are a clear example of UX applications.

   Also, interfaces are not only more centered around the user and user interaction, but also show a trend towards product integration. Domestic technology is becoming increasingly complex [Thomas and Macredie, 2002]. Our microwaves function also as stoves, we can listen to music on our mobile phones and our washing machines can also dry the laundry for us. Ubiquitous computing (UC), although radically different from traditional HCI on a number of criteria, is one extreme example where functionality is integrated.

3 Stressing the need for evaluation: three examples of emerging HCI applications

In this section, we discuss three examples of emerging HCI systems. These serve to demonstrate the observed trends in HCI system development, and allow to pinpoint the difficulties with traditional evaluation methods in Section 4.3.

3.1 Groupware systems

One example of an area where a lot of money has been invested into the development of a product because of its expected scenario gains is the area of group support systems (GSS) or groupware. De Vreede *et al.* [2003] conclude after extensive research that 15 years after the introduction of the first group support system, these systems indeed provide added value to meetings. They are said to provide savings, and increase efficiency. It was a rather complex and non-straightforward exercise to come to this conclusion.

One of the reasons that it took so long was the fact that people were facing difficulties when using the system, as they were not familiar with the changes in work practice that were introduced by them [Nunamaker Jr. *et al.*, 1995]. People were forced to use tools during meetings and had to abandon common meeting practice. As a consequence also its benefits proved hard to measure as people objected its use.

GSS are a clear example of systems that establish *a shift in application purpose*. Although Grudin [1994] already noted that adequate understanding of the political and social factors at work were to be considered in the design and implementation phases in order to avoid an initial reject from the public, the task of supporting the meeting process (e.g. facilitate brainstorming) was considered more important than its use. It was therefore not strange that people found it difficult to understand what the system was supposed to do for them and their group [Briggs *et al.*, 2003]. Design for intuitive interaction with the user as focal point would have facilitated its adoption, without any doubt.

3.2 Smart homes

Smart home systems are a typical example of a ubiquitous system, characterized by its pervasive nature. Users are observed using a large number of sensors, ranging from cameras and microphones to pressure and heat sensors. From a user point of view, ubiquitous systems do not necessarily have a task. They can be anywhere between responsive and pro-active. An example that lies somewhere in between responsive and pro-active is for instance the smart home described in Intille *et al.* [2003] where the system *suggests* users which cloth to wear given the outside temperature.

When the environment itself becomes the interface, people go about their daily lives and perform their tasks while the computing technologies are there to support them transparently [Weiser, 1991]. People start to implicitly interact with computers and technology disappears into the background. Despite being written over 10 years ago, many aspects of Mark Weisers vision of ubiquitous computing appear as futuristic today as they did in 1991 [Davies and Gellersens, 2002; Schmidt *et al.*, 2005].

As Davies and Gellersens [2002] mention there are many aspects that need to be resolved before ubiquitous interfaces really will break through. They mention, amongst others, the need for fusion models and context awareness. Due to the lack of an explicit interface, users are required to communicate naturally with the system. This requires fusion of multiple communication channels. The system must be aware of the context, and interpret the users action in this context. On the other hand, the user must be familiar with the system's abilities, and system's state.

Compared to Groupware systems, the complexity and black box characteristics of smart homes make them even more difficult to evaluate. This is due to the fact that smart homes not only introduce a *shift in application purpose*, but also employ *new sensing possibilities*. There is a radical *change in physical interface* since the smart home has become the interface itself. Some smart homes are pro-active, which is a clear *shift in initiative*.

3.3 Virtual dancer

Fun and entertainment are becoming increasingly important in almost all uses of information technology [Wiberg, 2005]. One example of an entertainment application is the Virtual dancer, as described in Reidsma *et al.* [2006]. It is an interactive installation where users can dance together with a virtual character. The virtual character reacts to the observed movements of the user, and tries to influence the movements of the user in turn. During the dance, there is a constant *shift in initiative*. The goal of the application is to entertain the user, without the provision of an explicit task. Instead, the interaction itself is the goal of the application, a clear *shift in application purpose*.

This so-called taskless interaction cannot be evaluated using traditional task-based evaluation methods. Attempts so far to evaluate the interaction have been limited to analyzing video recordings of the user in order to determine engagement in the interaction. This does not allow for reliable assessment of aspects that improve the user's experience during the interaction, let alone which parts of the system should be improved. One important aspect is that the responses of the user to certain actions of the systems have to be measured. This requires the knowledge of system states, i.e. the context. While this information proves valuable in the assessment of the participation level of the user, it does not provide much information about the actual user experience. Instead, this information could be collected using questionnaires or by employing bio-sensors that measure heart rate and the respiratory level.

4 Evaluation

Evaluation is broad concept. Here we adopt the definition of Preece *et al.* [1994]:

> Evaluation is concerned with gathering data about the usability of a design or product by a specific group of users for a particular activity within a specified group of uses or work context.

The use of evaluation methods for the assessment of the suitability of HCI systems has become a standard tool in the design process. Many HCI systems are designed iteratively, where in each cycle design issues of the previous one are addressed. These issues are identified in an evaluation step. We discuss the design criteria of HCI systems first in Section 4.1. We then focus on current evaluation practice in the HCI field in Section 4.2. Section 4.3 discusses issues that appear when dealing with evaluation for emerging HCI applications.

4.1 Design criteria in HCI

Much has been written about the design of HCI systems (e.g. Dix *et al.* [2004]. Designed well, interactive systems can allow us to reap the benefits of computation and communication away from the desktop, assisting us when we are physically, socially or cognitively engaged, or when we ourselves do not know what should happen next. Designed poorly, these same devices can wreck havoc on our productivity and performance, creating irritation and frustration in their wake [Ju and Leifer, to appear]. Good practice is to explicitly formulate design choices.

Norman [1998] identifies four principles for good interaction design. The controls should be visually obvious, they should be intuitive and part of a natural process, there should be proper feedback on the actions performed, and there should be a natural mapping between input and output.

Traditionally, HCI systems are designed for a certain task, in a given context, and with a certain user profile in mind. Key point is that the HCI system must be useful, usually referred to as usability. There are many different approaches to making a product usable and there is no accepted definition. Nielsen [1993b] identifies at least five components of usability: learnability, efficiency, memorability, errors, and satisfaction. In addition, usability can be regarded from three distinct viewpoints [Bevan *et al.*, 1991; Rauterberg, 1993]: product-oriented, user-oriented and user performance-oriented.

The product-oriented view can be measured in terms of ergonomic attributes of the product. The user-oriented view in terms of mental effort and attitude of the user and the user performance-view by examining how the user interacts with the product with emphasis on either the ease of use or the acceptability of the product in the real world.

The above views are complemented by the contextual view, which tells us that usability of a product is a function of a particular user class of users being studied, the application at hand and the environment in which they work.

Besides usability, in the interaction between the human and the computer also the user interface and user experience come into play. The notion of the user is important, and forms the basis of User-Centered Design (UCD). UCD is a multidisciplinary design approach based on the active involvement of users to improve the understanding of user and task requirements, and the iteration of design and evaluation [Mao *et al.*, 2005]. It has been mentioned that this approach is the key to product usefulness and usability and overcomes the limitations of traditional system-centered design.

One view of UCD is to design HCI as close as possible to natural human-human interaction [Reeves *et al.*, 2004]. The rationale is that users do not have to learn new communication protocols, which leads to increased interaction robustness. This aids the user experience and provides guidelines for designing the user interface. A drawback is that one should be familiar with the application to know what to expect from it.

4.2 Current evaluation practice in HCI

As stated before, evaluation is nowadays common practice in the field of HCI. The use of evaluation methods is motivated by the reported increased return on investments.

In general, we can identify two broad classes of evaluation methods: expert-based evaluation (e.g. cognitive walkthrough, heuristic evaluation, model based evaluation) and user-based evaluation (e.g. experimental evaluation, user observation, use of questionnaires, monitoring physiological responses). The bulk of early HCI designers and evaluators were cognitive psychologists. Cognitive models like GOMS [Card *et al.*, 1983] were very influential, as were laboratory experiments. Nielsen [1993b] took a more pragmatic approach, stating that full-scale evaluation of usability is too complicated in many cases, so that 'discount' methods are useful instead. His work has been very influential, partly due to the ease of application, partly due to the relative low cost. His vision has lead to an enormous number of different methods in regular use for the evaluation of usability.

Since its early days, HCI research focussed almost exclusively on the achievement of behavioral goals in work settings. The task that had to be performed by the user was the pivotal point of user centered analysis and evaluation. Rengger [1991] defined four classes of performance measures:

1. Goal achievement (accuracy and effectiveness)
2. Work rate (productivity and efficiency)
3. Operability (function usage)
4. Knowledge acquisition (learning rate)

As we discussed before, emerging HCI systems require other measures, and other evaluation practice. In the next section, we identify challenges for evaluation of emerging HCI systems, and use the examples in Section 3 as an illustration.

4.3 Challenges for evaluation of emerging HCI systems

The characteristics of emerging HCI systems imply that traditional approaches to usability engineering and evaluation are likely to prove inappropriate to the needs of its users. As a result of the trends that we discussed in Section 2.2, problems emerge in the design and evaluation of HCI systems. We discuss these below.

Human sensing

The use of keyboards, buttons and mice for interaction with HCI systems is found to be inconvenient since these devices do not support the natural ways in which humans interact. Although debated, the use of natural communication is often considered more intuitive, and therefore expected to be more efficient from a user's point of view. Voice, gestures, gaze and facial expressions are all natural human ways of expression. In natural contexts, humans will use all these channels. To make truly natural interfaces, this implies that all these channels should be taken into account. This, however, is difficult for at least three reasons:

1. The recognition is error-prone
2. The lexicon of expression is much larger than with 'artificial input'
3. Integration of multiple channels often leads to ambiguities

Error-prone recognition When using natural channels, the data obtained from sensors (microphone, camera) needs to be analyzed. From the streams of data, we need to recognize the communicative acts (words, gestures, facial expressions). Although much research is currently devoted to making automatic recognition more accurate, these systems will never be error-free. Another aspect is that automatic recognition is probably less fine-grained than what human observers are able to perceive [Abowd and Mynatt, 2000]. Subtleties might easily go unnoticed.

Reduction of errors is probably the most convenient way of improving the usability. However, as recognition will never be error-free, repair mechanisms need to be present. Feedback or insight in the system state are useful because they give the user insight in how the input is interpreted. Still, there are many challenges in how to present the feedback or system state [Bellotti *et al.*, 2002].

Assessment of the input reliability is an important aspect of usability evaluation. One way to do this is by applying standard benchmark sets. Well-known benchmark sets are the NIST RT sets [Fiscus *et al.*, 2006] for automatic speech recognition or FRVT and FRGC for face recognition [Phillips *et al.*, 2006]. These sets are specific for a given context and task. Since they contain ground truth and the error metrics are known, they allow for good comparison of recognition algorithms. However, they still evaluate only the reliability of the input. In addition to this, the system must be evaluated together with the (unreliable) input.

Large lexicon In natural human-human interaction, humans use a large lexicon of speech, and eye, head and body movements, both conscious and unconscious. When allowing humans to communicate with HCI systems in a natural way, the input devices should be able to recognize the whole range of signals. This poses severe requirements on the recognition.

Two factors are important when evaluating the lexicon. First, the lexicon should be sufficiently large to allow for all foreseen (and unforeseen) actions. For a system such as the Virtual dancer (see Section 3.3), this implies that the whole range of dance movements that a user can make, should be included into the lexicon.

Second, the choice of the lexicon should be intuitive. In many cases, an *ad hoc* lexicon is chosen, often to maximize the recognition. Ideally, the lexicon should contain signals that users naturally make when interacting with the HCI system. Note that, although this interaction is natural, the lack of a clear interface might prove that it is also not intuitive [Nijholt *et al.*, 2004]. A preliminary investigation should be conducted to see what these movements and sounds are, for example by conducting Wizard of Oz experiments.

When dealing with attentive or pro-active systems, not only the communicative actions are of importance. These systems require to be aware of things as user state and intentions, which generally can be deducted from behavior that is non-communicative.

Integration of channels Human behavior is multi-modal in nature. For example, humans use gestures and facial expressions while speaking. Understanding of this behavior does not only require recognition of the input of individual channels, but rather the recognition of the input as a whole. Despite considerable research effort in the field of multi-modal fusion (see e.g. Oviatt [2003]), our knowledge about how humans combine different channels is still limited. When dealing with multi-user systems, the problem is even harder since also the group behavior needs to be understood. Furthermore, due to the disappearing interfaces, the lack of explicit turn-taking will cause users to employ many alternate sequences of input, and requires HCI systems to be more flexible in handling these in turn [Nielsen, 1993a].

Similar to the performance evaluation of single communication channels, the recognition of the fused channel information need to be assessed. Integration of multiple channels can lead to reduction of signal ambiguity, provided that the context is known. Therefore, accurate assessment of the context is needed.

Context awareness

It is often mentioned that human behavior is to be interpreted in a given context. For example, a smile in a conversation can be a sign of appreciation, whereas, during negotiation, it can show disagreement. So for reliable interpretation of the human behavior, it is important to be aware of the context of the situation. Till date, there is no consensus of what context is precisely, and how we should specify this [Van Bunningen *et al.*, 2005]. Without a good representation for context, developers are left to develop *ad hoc* and limited schemes for storing and manipulating this key information [Abowd and Mynatt, 2000]. This is acceptable for small domains, but is inappropriate for more complex applications.

Usually, the context is specified as the identity and location of the users, and the characteristics and timing of the action performed. Ideally, even the intentions of the user should also be taken into account. This is particularly difficult since these can not be measured. These components of context are referred to as the 5 W's [Abowd and Mynatt, 2000; Pantic *et al.*, 2006]: who, what, where, when, why. These basic components are limited, and one might include the identity and locations of all objects of interest, as well as the current goal of the user. Also, the history of all environment changes and user actions are considered important for reasoning about the context.

It difficult to assess the right values for all these properties, and context aware systems are likely to consider contexts differently than users do. Intille *et al.* [2003] observe that, for smart homes (see Section 3.2), the user naturally considers contexts that the system has not, and propose to use suggestive systems, rather than pro-active ones.

Performance metrics

In contrast to Rengger [1991], as discussed in Section 4.2, emerging HCI applications often do not have well-defined tasks, which asks for novel measures. There are many factors in HCI that have a substantial impact on the success of applications that are not easily quantified. Amongst them are user experience [Thomas and Macredie, 2002], fun [Blythe *et al.*, 2003], ethical issues [Nardi *et al.*, 1995], social relationships [Grudin, 1988] and aesthetical issues [Alben, 1996]. For example, for the Virtual dancer (see Section 3.3), it remains a challenge to define proper measures to evaluate the interaction. These critical parameters are also required in order to compare similar applications [Newman, 1997].

Reference tasks

Whittaker *et al.* [2000] observed that many developed HCI systems can be considered radical inventions. They do not build further on established knowledge about user activities, tasks and techniques but rather push the technology envelope and invent new paradigms. Although we lack basic understanding of current users, tasks and technologies, the field of HCI is encouraged to try out even more radical solutions, without pausing to do the analysis and investigation required to gain systematic understanding. The absence of shared task or goal information makes it difficult to focus on research problems, to compare research results and to determine when a new solution is better, rather than different. This prevents proper consolidation of knowledge.

When the users are not familiar with the task or goal the application supports, users are likely to use the system in a different way. This makes evaluation of the fitness of the system difficult. For example, interfaces that support creative thinking are designed for a specific task that is new to the users. Without proper familiarization, these interfaces are less effective (see for example the Groupware example in Section 3.1).

Learnability

Given the increasing complexity of HCI systems, it is to be expected that the time needed to learn to work with a system grows along. Currently, evaluation of these systems focusses on 'snap shots', but fail to focus on the learning [Petersen *et al.*, 2002]. Longitudinal studies that assess how the use of a system develops from the first encounter are needed to gain insight in what kind of barriers users encounter when using the system, and how they solve these.

Context of authentic use

HCI systems should be evaluated in a context as close as possible to the context of authentic use [Abowd and Mynatt, 2000]. The context is often difficult to realize. Evaluating HCI systems in laboratory settings is likely to cause unnatural behavior of the users.

Another drawback of using laboratory testing is that parameters can be controlled (background noise, lightning conditions) that cannot be controlled in the context of authentic use. As a consequence, there is a difference in how these systems perform in reality.

As an example, the live-in laboratory PlaceLab [Intille, 2006] has been built to ensure that assumptions about behavior in the lab correspond to behavior in more realistic (and complex) situations in real smart homes.

5 Conclusion

New HCI systems are emerging that differ from traditional single-user, task-based, physical-interface HCI systems. We identify four trends: new sensing possibilities, a shift in initiative, diversifying physical interfaces, and a shift in application purpose. Traditional evaluation practice does not suffice for these new trends.

The use of more natural interaction forms poses problems when the input is ambiguous, the communication lexicon is potentially large, and when interpreting signals from multiple communication channels, ambiguities might arise. Identifying the context of use is important because interpretation of input is often dependent on the context. For complex systems, sensing the context is increasingly difficult. Evaluation of context aware systems is consequently difficult.

There is no consensus about appropriate performance metrics for emerging HCI systems. Task-specific measures are useless for evaluation of task-less systems. Related to this is the lack of common reference tasks. The ‘radical invention’ practice in the field of HCI prevents proper consolidation of knowledge about application tasks and goals, and user activities. Therefore, it is difficult to compare HCI systems.

As HCI systems are becoming more complex, the learning process of users is more and more important. This is currently a neglected part of evaluation. The introduction of longitudinal evaluation studies is needed to gain insight in the learning mechanisms. A final practical issue is the lack of authentic usage contexts. Many systems are only evaluated in a laboratory setting, instead in their projected context.

We summarized trends in HCI systems and pointed out where problems appear. We discussed three examples of complex HCI systems, and argued the need for appropriate evaluation. With this paper, we aimed at achieving increased awareness that evaluation too has to evolve to support the emerging trends in HCI systems.

Acknowledgments

This work was partly supported by the European Union 6th FWP IST Integrated Project AMI (Augmented Multi-party Interaction, FP6-506811, publication 195, and is part of the ICIS program. ICIS is sponsored by the Dutch government under contract BSIK03024. The authors wish to thank Betsy van Dijk and Anton Nijholt for their valuable comments.

References

Gregory D. Abowd and Elizabeth D. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, 7(1):29–58, 2000.

Lauralee Alben. Quality of experience: defining criteria for effective interaction design. *Interactions*, 3(3):11–15, 1996.

Victoria Bellotti, Maribeth Back, Keith Edwards, Rebecca E. Grinter, Austin Henderson Jr., and Christina V. Lopes. Making sense of sensing systems: Five questions for designers and researchers. In *Proceedings of the Conference on Human Factors in Computing Systems (CHI’02)*, pages 415–422, Minneapolis, MN, 2002.

Steve Benford, Holger Schndelbach, Boriana Koleva, Rob Anastasi, Chris Greenhalgh, Tom Rodden, Jonathan Green, Ahmed Ghali, Tony Pridmore, Bill Gaver, Andy Boucher, Brendan Walker, Sarah Pennington, Albrecht Schmidt, Hans-Werner Gellersen, and Anthony Steed. Expected, sensed, and desired: A framework for designing sensing-based interaction. *ACM Transactions on Computer-Human Interaction*, 12(1):3–30, 2005.

Nigel Bevan, Jurek Kirakowski, and Jonathan Maissel. What is usability? In *Proceedings of the international Conference on HCI*, pages 651–655, Stuttgart, Germany, 1991.

Mark A. Blythe, Kees J. Overbeeke, Andrew F. Monk, and Peter C. Wright. *Funology: From Usability to Enjoyment*, volume 3 of *Human-Computer Interaction Series*. Kluwer Academic Publishers, 2003.

Robert O. Briggs, Gert-Jan de Vreede, and Jay F. Nunamaker Jr. Collaboration engineering with thinklets to pursue sustained success with group support systems. *Journal of Management Information Systems*, 19(4):31–64, 2003.

Stuart K. Card, Allen Newell, and Thomas P. Moran. *The Psychology of Human-Computer Interaction*. Lawrence Erlbaum Associates, Mahwah, NJ, 1983.

Michael H. Coen. Design principles for intelligent environments. In *Proceedings of the National Conference on Artificial Intelligence (AAAI’98)*, pages 547–554, Madison, WI, 1998.

Nigel Davies and Hans-Werner Gellersens. Beyond prototypes: Challenges in deploying ubiquitous systems. *IEEE Pervasive Computing*, 2(1):26–35, 2002.

Gert-Jan de Vreede, Douglas R. Vogel, Gwendolyn L. Kolfschoten, and Jeroen Wien. Fifteen years of GSS in the field: A comparison across time and national boundaries. In *Proceedings of the Hawaii International Conference on System Sciences (HICSS’03)*, page 9, Big Island, HA, 2003.

Alan Dix, Janet Finlay, Gregory Abowd, and Russell Beale. *Human Computer Interaction, third edition*. Prentice Hall, 2004.

Jonathan G. Fiscus, Nicolas Radde, John S. Garofolo, Audrey Le, Jerome Ajot, and Christophe Laprun. The rich transcription 2005 spring meeting recognition evaluation. In Steve Renals and Samy Bengio, editors, *Revised Selected Paper of the Machine Learning for Multimodal Interaction Workshop 2005 (MLMI'05)*, volume 3869 of *Lecture Notes in Computer Science*, pages 369–389, Edinburgh, United Kingdom, 2006.

Jodi Forlizzi and Katja Battarbee. Understanding experience in interactive systems. In *Proceedings of the conference on Designing Interactive Systems (DIS'04)*, pages 261–268, Cambridge, MA, 2004.

Bill Gaver and Heather Martin. Alternatives: exploring information appliances through conceptual design proposals. In *Proceedings of the conference on Human factors in computing systems (CHI'00)*, pages 209–216, 2000.

Jonathan Grudin. Why CSCW applications fail: problems in the design and the evaluation of organizational interfaces. In *Proceedings of the Conference on Computer Supported Cooperative Work (CSCW'88)*, pages 85–93, New York, USA, 1988.

Jonathan Grudin. Groupware and social dynamics: Eight challenges for developers. *Communications of the ACM*, 37(1):93–105, 1994.

Marc Hassenzahl and Nina Sandweg. From mental effort to perceived usability: transforming experiences into summary assessments. In *Extended abstracts on Human factors in computing systems (CHI'04)*, pages 1283–1286, Vienna, Austria, 2004.

Marc Hassenzahl and Noam Tractinsky. User experience a research agenda. *Behaviour & Information Technology*, 25(2):91–97, 2006.

Stephen S. Intille, Emmanuel M. Tapia, John Rondoni, Jennifer Beaudin, Chuck Kukla, Sitij Agarwal, Ling Bao, and Kent Larson. Tools for studying behavior and technology in natural settings. In *Proceedings of the International Conference on Ubiquitous Computing (Ubicomp'03)*, volume 3869 of *Lecture Notes in Computer Science*, pages 157–174, Seattle, WA, 2003.

Stephen S. Intille. The goal: smart people, not smart homes. In *Proceedings of the International Conference on Smart Homes and Health Telematics*, pages 3–6, Belfast, United Kingdom, 2006.

Wendy Ju and Larry Leifer. The design of implicit interactions. *Design Issues, Special Issue on Design Research in Interaction Design*, to appear.

Ji-Ye Mao, Karel Vredenburg, Paul W. Smith, and Tom Carey. The state of user-centered design practice. *Communications of the ACM*, 48(3):105–109, 2005.

Bonnie A. Nardi, Allan Kuchinsky, Steve Whittaker, Robert Leichner, and Heinrich Schwarz. Video-as-data: Technical and social aspects of a collaborative multimedia application. *Computer Supported Cooperative Work*, 4(1):73–100, 1995.

William M. Newman. Better or just different? On the benefits of designing interactive systems in terms of critical parameters. In *Proceedings of the conference on Designing interactive systems: processes, practices, methods, and techniques*, pages 239–245, Amsterdam, The Netherlands, 1997.

Jakob Nielsen. Noncommand user interfaces. *Communications of the ACM*, 36(4):83–89, 1993.

Jakob Nielsen. *Usability Engineering*. Academic Press, Boston, MA, 1993.

Anton Nijholt, Thomas Rist, and Kees Tuijnenbreijer. Lost in ambient intelligence? In *Extended abstracts on Human factors in computing systems (CHI'04)*, pages 1725–1726, Vienna, Austria, 2004.

Donald A. Norman. *The Design of Everyday Things*. MIT Press, 1998.

Jay F. Nunamaker Jr., Robert O. Briggs, and Daniel D. Mittleman. Electronic meeting systems: Ten years of lessons learned. In David Coleman and Raman. Khanna, editors, *Groupware: Technology and Applications*. Prentice Hall, Englewood Cliffs, NJ, 1995.

Sharon L. Oviatt. *The Human-Computer Interaction Handbook: Fundamentals, Evolving Technologies and Emerging Applications*, chapter 14: Multimodal interfaces, pages 286–304. Lawrence Erlbaum Associates, 2003.

Maja Pantic, Alex Pentland, Anton Nijholt, and Thomas S. Huang. Human computing and machine understanding of human behavior: a survey. In *Proceedings of the International Conference on Multimodal Interfaces (ICMI'06)*, pages 239–248, Banff, Canada, 2006.

Marianne G. Petersen, Kim H. Madsen, and Arne Kjær. The usability of everyday, technology-emerging and fading opportunities. *ACM Transactions on Computer-Human Interaction*, 9(2):74–105, 2002.

Jonathon Phillips, Patrick J. Flynn, Todd Scruggs, Kevin W. Bowyer, and William Worek. Preliminary face recognition grand challenge results. In *Proceedings of the Conference on Automatic Face and Gesture Recognition 2006 (FGR'06)*, pages 15–24, Southampton, United Kingdom, 2006.

Jenny Preece, Yvonne Rogers, Helen Sharp, and David Benyon. *Human-Computer Interaction*. Addison-Wesley Longman Ltd., 1994.

Matthias Rauterberg. Quantitative measures for evaluating human-computer interfaces. In *Proceedings of the International Conference on Human-Computer Interaction*, pages 612–617, Orlando, Florida, 1993.

Leah M. Reeves, Jennifer Lai, James A. Larson, Sharon L. Oviatt, T. S. Balaji, Stéphanie Buisine, Penny Collings, Phil Cohen, Ben Kraal, Jean-Claude Martin, Michael McTear, TV Raman, Kay M. Stanney, Hui Su, and Qian Ying Wang. Guidelines for multimodal user interface design. *Communications of the ACM*, 47(1):57–59, 2004.

Dennis Reidsma, Herwin van Welbergen, Ronald Poppe, Pieter Bos, and Anton Nijholt. Towards bi-directional dancing interaction. In *International Conference on Entertainment Computing (ICEC'06)*, volume 4161 of *Lecture Notes in Computer Science*, pages 1–12, September 2006.

Ralph E. Rengger. *Human Aspects in Computing: Design and Use of Interactive Systems with Terminals*, chapter Indicators of Usability based on performance, pages 656–660. Elsevier, Amsterdam, The Netherlands, 1991.

Albrecht Schmidt, Matthias Kranz, and Paul Holleis. Interacting with the ubiquitous computer: towards embedding interaction. In *Proceedings of the joint conference on Smart objects and ambient intelligence (sOc-EUSAI'05)*, pages 147–152, Grenoble, France, 2005.

Norbert Streitz and Paddy Nixon. Introduction: The disappearing computer. *Communications of the ACM*, 48(3):32–35, 2005.

Peter Thomas and Robert D. Macredie. Introduction to the new usability. *ACM Transactions on Computer-Human Interaction*, 9(2):69–73, 2002.

Arthur H. van Bunningen, Ling Feng, and Peter M.G. Apers. Context for Ubiquitous Data Management. In *International Workshop on Ubiquitous Data Management (UDM'05)*, pages 17–24, Tokyo, Japan, 2005.

Mark Weiser. The computer of the 21st century. *Scientific American*, 265(3):66–75, 1991.

Steve Whittaker, Loren Terveen, and Bonnie A. Nardi. Let's stop pushing the envelope and start addressing it: a reference task agenda for HCI. *Human Computer Interaction*, 15(2-3):75–106, 2000.

Charlotte Wiberg. Usability and fun: An overview of relevant research in the hci community. In *Proceedings of the CHI Workshop on Innovative Approaches to Evaluating Affective Interfaces*, Portland, OR, 2005.

When Will a Human in the Loop Accelerate Learning?

Hema Raghavan
hema@cs.umass.edu
University of
Massachusetts
Amherst MA, USA

Omid Madani & Rosie Jones
{madani, jonesr}@yahoo-inc.com
Yahoo! Research
3333 Empire Avenue, Burbank
CA 91504, USA

Abstract

Supervised learning typically requires human effort to label a large number of training instances. Active learning strives to decrease the number of labeled training examples needed by actively engaging the learner and the human in an interactive process. Active learning has proven to be effective in many domains. With few training examples, past work has found that user prior knowledge on the importance of features can guide the learner to converge faster, that is, with lower labeling costs. In this paper we aim to understand the kinds of problems for which such human in the loop procedures are actually beneficial. In other words we ask whether there are some problems which significantly benefit from interactive learning and whether for some problems the user has no choice but to engage in the tedious process of labeling several examples. Towards this goal, we define a set of four difficulty measures, 2 each of instance and feature complexity, for linear classification problems. These measures can efficiently be computed for real world problems for which linear classifiers are effective such as text classification. We quantify the difficulty of 358 text classification problems and 9 corpora using our measures, illustrating the spectrum of problems that exist in text classification in addition to quantifying results that have only been qualitatively discussed in the text classification literature. We verify the intimate relationship (a high positive correlation) between feature complexity and instance complexity using our measures. We then use these measures to understand when machine learning with a human in the loop is likely to be useful.

1 Introduction

Many supervised learning problems require that a large amount of data be labeled. Labeling data is a tedious and costly process. A user who wants to train a personalized news filter, or their own customized spam filter is probably willing to invest some (but very little) time in training the system. In this paper we investigate whether there are some problems for which a human can guide the learner to an acceptable level of accuracy equivalent to what would be obtained using several randomly picked examples. If there exist such problems, then we hypothesize that there need to move beyond the traditional paradigm of instance based learning. We need a paradigm in which the learner asks the human "big bang for the buck" questions at each iteration. We want to bring forth for discussion what these other learning paradigms may be and whether there is any benefit to these alternate paradigms.

Active learning algorithms aim to intelligently choose instances from a pool of unlabeled data for an expert to label, with the hope of decreasing the number of labeled instances as compared to the case when these instances are picked randomly [Lewis and Catlett, 1994; Cohn *et al.*, 1990]. Figure 1 shows the performance obtained as a function of the number of training documents when the training documents are (a) actively and (b) randomly sampled. The active learner achieves the best possible accuracy with about 1/10th the training examples that random sampling needs.

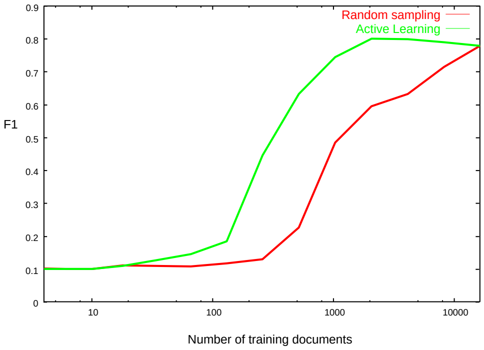


Figure 1: Active Learning

In supervised learning human input is not just restricted to labeling of examples. The human also picks a feature representation for these examples. The feature set is often large consisting of many redundant and irrelevant features and it is up to the learner to discern the good features from the bad. If the feature set is huge, then a large amount of training is needed for the learner to assign the correct weights and focus on the important features. However, in many applications like text classification, users may have some knowledge of the usefulness of features, and there has been increasing interest in harnessing user prior knowledge on features to bootstrap learning especially when the number of labeled examples are few[Raghavan *et al.*, 2005; Godbole *et al.*, 2004].

This paper aims to study the kinds of problems for which user feedback is likely to accelerate learning. We define a set of difficulty measures in Section 3 based on the number of instances and the number of features needed to achieve the maximum accuracy. The instance complexity measure intuitively captures the number of intelligently picked training examples needed to achieve the maximum achievable accuracy. A problem for which training on a few instances is sufficient to attain the maximum achievable accuracy is a low instance complexity problem. More generally speaking, you are more likely to see greater benefits from active learning for low instance complexity problems. Analogous to instance complexiy we define feature complexity which captures the minimum number of intelligently picked features needed to achieve the maximum achievable accuracy. A binary classification problem that needs only a few attributes to completely separate the positive class from the negative one is an example of a low feature complexity problem. Again, generally speaking, involving a user in intelligent feature selection is likely to see more benefits for low feature complexity problems.

In Section 4 we benchmark several text classification corpora for their difficulty (See Fig. 2). We find that our measures capture previously held beliefs about the difficulty of various text classification problems (See Fig. 3)[Bekkerman *et al.*, 2001; Joachims, 1997]. However, past work has typically considered only the Reuters-21578 and 20 Newsgroups corpora. By benchmarking 9 corpora and 358 problems, we place these two corpora and their underlying problems in perspective with respect to a broad range of text categorization problems.

The dual nature of complexity seems to imply that for low complexity problems, an intelligently picked feature is as good as an inteligently picked instance. We also know that labeling features is much faster than labeling instances and that users can pick the most predictive features fairly accurately [Raghavan *et al.*, 2005]. However, for more complex problems feature selection may be much more difficult for the user and instance feedback is the more reasonable alternative. Hence, we think a tandem approach of asking on instance feedback and feature feedback is most beneficial: if the problem is of low complexity, a few features that the user marks will quickly lead the classifier to convergence; if the problem is of high complexity, the user would not be able to recommend features but can provide feedback on instances instead. In section 6 we try to understand the implications of the dual nature of complexity. We simulate an active learning system that extends the traditional document feedback scenario to ask the users to mark features feedback aka the system in [Godbole *et al.*, 2004; Raghavan *et al.*, 2005]. We find that feature feedback accelerates active learning speed by an amount that is inversely related to the feature complexity of the problem. For low feature complexity problems, a few training documents combined with feature feedback can give a big improvement in speed. Many problems in text classification fall in the low to medium range of complexity and stand to gain from such a dual feedback (term feedback + document feedback) framework.

Our main contributions are:

- Proposing efficient procedures for quantifying inherent learning complexity in terms of instances as well as features. We make novel use of selective sampling (active learning) and feature ordering to achieve this. We use our measures to measure the correlation between instance complexity and feature complexity, highlighting the dual nature of complexity: a learning problem requiring a large number of instances to learn, requires large number of features, and vice versa.
- Benchmarking several text classification corpora and their underlying problems and using that to understand when and why active learning or intelligent feature selection are beneficial.
- Lastly, given the dual nature of complexity (from (1)) we explore the kinds of problems for which feature feedback in conjunction with document feedback can help an active learning algorithm converge faster than using document feedback alone.

We would like to emphasize that we are not measuring the degree of learnability of a categorization problem, i.e., we are neither trying to predict nor measure the best achievable performance, nor the complexity of the boundary between classes. We are instead trying to measure how quickly an active learner converges and obtaining insights into the underlying causes. To this end we use our complexity measures to characterize problems as easy or difficult for interactive learning. Our measures are defined in such a way that it gives greater emphasis to the early stage of learning, i.e., a problem for which there is significant improvement in accuracy at the early stage of learning gets a lower score (less complex) than a problem that achieves the same improvement at a later stage of learning.

We believe that our complexity measures and observations are applicable to other high dimensional domains such as Bioinformatics, NLP and vision where linear classifiers do well [Golub and et al, 1999; Punyakanok and Roth, 2005; Wu and Zhang, 2004]. Our measures can be used to select problems and domains that are particularly difficult for active learning research. We want to bring up questions about when machine learning with a human in the loop is likely to be useful. Often times what seems important information for a human to learn a concept, may or may not be as important for a learning algorithm. With these questions in mind we explored the benefits of feature feedback as an aid to document feedback for text classification. We then ask what other types of feedback can users provide, and what is it that the learner needs to know. We want to bridge the gap between machine learning and HCI. For example, in this paper we see a thorough examination of how and why a learner benefits from feature feedback. Then given that feature feedback is likely to be useful, the question remains as to how to ask the human for feature feedback in a way that is of low cognitive load. These and other questions that we want to bring up for discussion are highlighted in section 8.

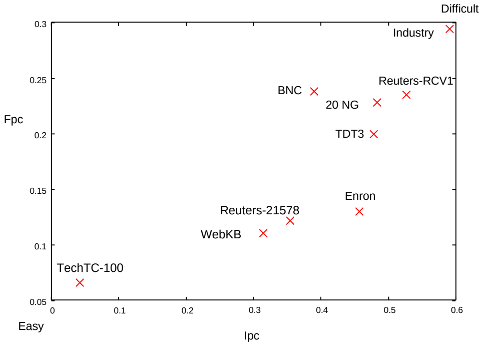


Figure 2: **Instance Complexity (I_{pc}) and Feature Complexity (F_{pc}). A higher value of complexity indicates a difficult problem. Notice how instance complexity and feature complexity are correlated.**

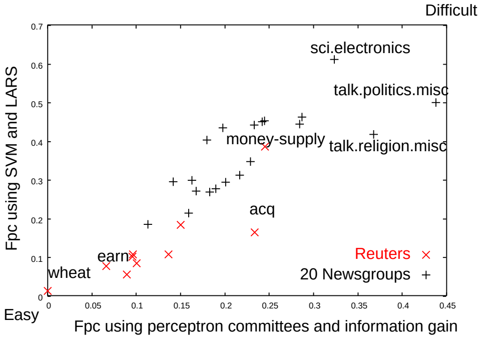


Figure 3: **Feature complexity (F_{pc}) scores of problems in the Reuters-21578 and 20 Newsgroups corpora computed using 2 different methods. Higher the complexity more difficult the problem.**

2 Data Sets

We consider 9 corpora and 358 binary classification problems as shown in Table 1. Most tasks are classification into topic-based categories, except for some tasks in two corpora: (1) the Topic Detection and Tracking corpus that contains classes based on events, for example *Hurricane George* and *Hurricane Mitch* are separate classes; and (2) the British National Corpus BNC corpus where the classes are based on genre. For all data sets we used unigram features. For some of them we further added n-grams of features if these n-grams improved performance[1].

Since we are studying convergence of a linear classifier, we considered only those problems for which there is ample training data to achieve an acceptable level of performance (of above 75% F1). Given that we have class skews as small as 0.003% for some of our problems, accuracy is not an appropriate measure of effectiveness and hence we chose $F1$ [Lewis *et al.*, 2004], the harmonic mean of precision and recall. The last column in Table 1 lists the average maximum $F1$ obtained using a linear classifier and bag-of-word features trained on 90% of the data and tested on the remaining.

3 Measures of complexity

We now describe 4 measures of complexity – 2 of instance complexity and 2 of feature complexity. Consider a learning algorithm which is supplied with the best possible training examples available for a task in a corpus. If only a few of these training instances are required for learning the task to high performance, we will say the task has low instance complexity. If a large number are required, we will say the task has high instance complexity. Our instantiation of these instance complexity measures attempt to capture how many of the best (most informative) instances for a given problem are needed in order to achieve performance close to that of a linear classifier trained with all features and ample training examples. In computing the instance complexity we use active learning methods which give us experimental upper bounds on complexity i.e., the tightness of the bound is dependent on the active learning method used.

Similarly, our feature complexity measures quantify how many of the most informative features are needed to achieve close to the maximum accuracy. Our feature complexity measures are also upper bounds on the true feature complexity, where the tightness of the bound is dependent on the feature selection method used.

3.1 Instance Complexity Measures

Active learning via selective sampling (*e.g.*, [Lewis and Catlett, 1994]) is a type of supervised learning where the learner is actively engaged in choosing the most informative examples for the expert to label, thereby lowering the manual labeling effort on the part of the expert.

Given a classification algorithm and a binary classification problem, there is some maximum achievable performance, often under 100% in practice (Table 1). In measuring the rate of learning we want to measure the minimum number of training examples we need in order to achieve the best performance for a given classifier. Note that simulating an oracle for a task with M examples would require training the classifier for every possible subset of training examples, that is, 2^M times. Using an active learning algorithm gives us an upper bound on the minimum number of examples needed for optimal accuracy, which requires training the classifier $O(M)$ times. How close this is to the true bound is dependent on the effectiveness of the ordering of instances by the active learning algorithm.

Our active learning algorithm begins with 2 randomly selected instances, one in the positive and one in the negative class. The active learner learns a classifier based on this information and then intelligently chooses the next instance from a pool of unlabeled examples for the expert to label. The classifier is retrained and the process continues.

We measure the performance, $F1_{2^t}(active)$ of the classifier after every 2^t iterations of active learning with t vary-

[1] Preprocessed sparse matrices for the freely available datasets are available at `http://url.hidden.for.review`

Corpus	Domain	M (# instances)	N (# features)	# topics	MaxF1
Reuters-21578	News-wire	9410	33378	10	0.874 (0.087)
Reuters-RCV1	News-wire	23149	47236	87	0.759(0.127)
Topic Detection Tracking(TDT)	News-wire and broadcast	67111	85436	10	0.918(0.001)
British National Corpus	News, journals etc.	2642	233288	15	0.774 (0.153)
Enron	E-mail folders	1971	711815	8	0.887(0.082)
20 Newsgroups	Newsgroup postings	19976	137728	20	0.851(0.007)
Industry Sector	Corporate web-pages	9565	69297	104	0.909(0.04)
TechTC-100	ODP hierarchy	149	18073	100	0.972(0.026)
WebKB	University websites	2101	28682	4	0.918(0.047)

Table 1: **For all corpora except TechTC-100 there is a one one-versus-all binary classification problem. The TechTC-100 dataset consists of a 100 binary classification problems with about 149 documents in each and an average of 18073 features in each.**

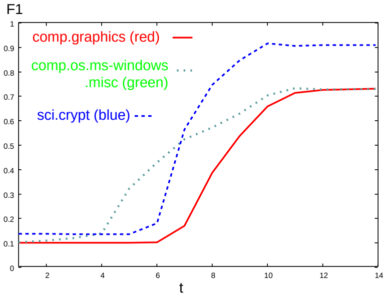


Figure 4: **Learning curves for a single classifier on 3 problems.**

ing as $1, 2, ..., \log_2 M$, where M is the total number of instances. We can measure active learning performance using the area under this resulting performance curve. Such a performance curve for three problems in the 20 Newsgroups data set is shown in Figure 4. For the red and green curves, the learner achieves the maximum attainable accuracy (0.70 F1) after seeing 2048 (2^{11}) examples. However, the rate of convergence of the green curve is initially higher than the red curve. Now consider the blue learning curve in Figure 4. The maximum accuracy achieved for this problem is much higher (0.90 F1) than for the other two problems, though the rate of active learning of the blue curve is more similar to the red curve than the green one, which is indicated by the profiles or shapes of the curves.

We measure the *active learning convergence profile* as the area under the normalized curve, given as:

$$p_{\mathrm{al}} = \frac{\sum_{t=2}^{log_2 M} F1_{2^t}(active)}{log_2 M \times F1_M(active)} \tag{1}$$

The value of p_{al} ranges between 0 and 1 and is independent of M, the total number of instances. Since performance is measured at exponentially increasing intervals, p_{al} implicitly gives a higher score to problems that converge more rapidly in the early stage of learning than later. Intuitively, a difference of 100s of instances (or features) is more significant when two problems require 10s or 100s of instances each, versus 1000s of instances. Similar to the Richter scale for earthquakes[2] of all our measures will be based on a logarithmic scale. Higher p_{al} implies faster convergence. The green, red and blue curves have p_{al} values of 0.61, 0.45 and 0.55 respectively.

We now describe our first two measures of complexity:

1. Instance profile complexity, I_{pc}: This measure is simply the complement of the active learning convergence profile, and is given as $I_{pc} = 1 - p_{\mathrm{al}}$. The higher the value, the more difficult or complex the problem. The active learning curve and hence the value of speed obtained is subject to the active learning algorithm and will be less than the ideal (theoretical best ordering of instances) case. Therefore, I_{pc} is an upper bound on the true complexity.

2. Instance complexity, C_i: In the process of dividing by $log_2 M$ for *speed* (see Eq. 1) we lose information about the total number of instances needed to reach the best possible performance. Consequently this information is lost in I_{pc}. For example, two active learning performance curves might look identical when normalized, but the actual number of instances needed to achieve maximum performance is different. We therefore define $C_{inst} = I_{pc} * i$ where i is the logarithm of the number of instances needed to achieve 95% of the best performance. i is an upper bound on the true value obtained had we known the ideal ordering.

3.2 Feature Complexity Measures

Our third and fourth measures attempt to capture the complexity of the problem in terms of the number of features needed to reach the best possible performance. We learn a ranking of the features in the order of decreasing discriminative ability for a given classification problem by using a large number of training documents and a feature selection criterion like information gain. We consider the performance of the classifier constructed using n top ranking features. We plot a feature learning curve by plotting performance at exponentially increasing intervals of n. The normalized area under this feature learning curve, *the feature learning convergence profile*, p_{fl} is computed by dividing by the best performance and the total number of features.

1. Feature profile complexity, F_{pc}: *Feature profile complexity* (F_{pc}) is then defined as $F_{pc} = 1 - p_{\mathrm{fl}}$. The computed value of F_{pc} is limited by the accuracy of the feature selection algorithm and is therefore an upper bound on the true feature complexity.

2. **Feature complexity**, C_f: Similar to C_i, we define $C_f = F_{pc} * f$, where f is the logarithm of the number of features in

[2] An earthquake of magnitude 6 is significantly more intense than one of magnitude 5

the feature learning curve needed to achieve 95% of the best performance.

3.3 Methods

The classifiers used in this paper are of the form $f(X) = w \cdot X + b$ where, X is a vector of features representing an instance, w is the vector of weights and b is a threshold. If $f(X) > 1$, the instance X is classified as positive, otherwise it is classified as negative.

We used a **committee of perceptrons** [Dasgupta *et al.*, 2005] for instance selection. We found a committee of 50 perceptrons to be effective and efficient.

We used **information gain**, a standard feature selection method that is computationally efficient [Brank *et al.*, 2002], to compute feature complexity.

For a given training and test set, we sweep through all values of b and use that b for which the F_1 is maximum, allowing us to compute a tighter upper bound

We also experiment with SVM uncertainty sampling [Lewis and Catlett, 1994] (**SVM**s are one of the most effective linear classifiers for text classification problems) for instance selection and **SVM LARS** [Keerthi, 2005], a new and effective forward selection technique for feature selection. However, both these algorithm have relatively high running time and we use it only in a limited way i.e., the p_{al} and p_{fl} values are computed by plotting the learning curves only up to 1024 features and instances respectively.

SVM LARS ignores highly correlated features in its feature selection whereas information gain does not. Therefore, the feature complexities computed by LARS may be more representative of the true feature complexity than complexity computed using information gain. Similarly uncertainty sampling using SVMs has generally proven to be empirically better than perceptron committees for instance selecion. Therefore the SVM based methods may give us a slightly tighter upper bound than perceptron and information gain, but the latter techniques are computationally more efficient and will give us a reasonable ranking of the problems and the difficulty of domains.

4 Difficulty of Corpora

We now benchmark the 9 corpora introduced in Section 2 as easy or difficult for active learning using our complexity measures. Table 2 shows the complexity of different data sets. By all measures the Tech100 data set ranks as the easiest, followed by WebKB and Reuters. BNC, Reuters-RCV1, 20 Newsgroups and the Industry sector corpora are difficult by both our instance complexity and feature complexity measures. This is better illustrated in the chart in Figure 2. This figure also indicates that instance complexity and feature complexity are highly correlated, which we will discuss in detail in the next section. Most corpora have problems of varying difficulty which is indicated by the standard deviation of the scores in Table 2. Even though the BNC corpus is small (less than 3k documents) it falls into the difficult end of the spectrum implying that genre classification is more difficult than subject based categorization.

The ranking of corpora using F_{pc} computed using SVM with LARS and Perceptron with information gain are also near identical as is illustrated by Figure 5 (We only show a subset of the problems to illustrate this, due to the slow running time of LARS). The ranking of individual problems in these two corpora using F_{pc} computed using these two methods also correlate fairly well (r=0.73[3]). The F_{pc} scores for individual problems in the Reuters-21578 and 20 Newsgroups corpus using both methods are illustrated in Figure 3. Our results also support previous results that say that 20-Newsgroups consists of problems that are more difficult than Reuters-21578 and that problems like *wheat* are much easier with lower feature complexity as compared to *acq*.

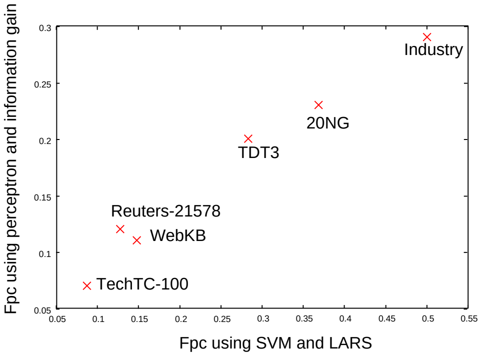


Figure 5: **Ranking using F_{pc} computed by two different methods results in a similar ranking of corpora.**

The Tech100 data set is a result of the efforts of Davidov et al [2004] to obtain a data set containing problems of varying difficulty in terms of maximum performance achievable. Yet we find all of the problems in this data set are of low complexity i.e., a few well chosen examples or features are sufficient to achieve the optimal accuracy.

The TDT corpus consists of English newswire documents (Eng News), the output of an automatic speech recognizer system for English broadcast sources (Eng ASR), machine translated newswire sources (MT News) and broadcast sources in Mandarin preprocessed through an ASR system and a machine translation system (MT ASR). We measured the difficulty of each of the subsections of this corpus. The C_f values for event based categorization are shown in the second column of Table 3.

The English sub-section of the corpus is easier than the machine translated one, which is more noisy. For example, topic 30036 is *Nobel Prizes Awarded*. The feature complexity of this problem in each subset is shown in the third column. The most important words in English Newswire and English ASR are (as expected) *Nobel*, *prize*, *Saramago* (person who won it) etc, making classificaton in Eng-News relatively easy. However, in MT News and MT ASR the most important keywords are *promises*, *Bell*, *prize* and *award*. The word *Nobel* is consistently translated to *promises Bell* in documents whose original source is *Mandarin*[4]. Names like *Saramago* which are

[3] r is Pearson's correlation coefficient, and r=1 denotes perfect correlation

[4] Nobel is a 3 character word in Mandarin, the first of two of which also correspond to the English word *promises* and the third of which corresponds to the English name *Bell*

Corpus	Instance Complexity Measures			Feature Complexity Measures		
	I_{pc}	i	C_i	F_{pc}	f	C_f
Tech100	0.04 (0.06)	3.24 (2.23)	0.20 (0.33)	0.07 (0.02)	1.89 (1.43)	0.14 (0.14)
WebKB	0.31 (0.13)	8.75 (0.50)	2.72 (1.04)	0.11 (0.04)	4.00 (2.16)	0.51 (0.47)
Reuters-21578	0.35 (0.13)	8.20 (1.03)	2.93 (1.24)	0.12 (0.07)	4.80 (2.04)	0.69 (0.56)
BNC	0.39 (0.16)	7.93 (1.91)	3.34 (1.73)	0.24 (0.11)	11.47 (3.83)	2.97 (1.60)
Enron	0.46 (0.09)	8.33 (0.87)	3.82 (0.94)	0.13 (0.06)	7.67 (4.42)	1.18 (0.70)
20NG	0.48 (0.04)	10.40 (0.68)	5.04 (0.71)	0.23 (0.08)	10.05 (1.39)	2.32 (0.95)
TDT3	0.48 (0.13)	9.30 (1.06)	4.55 (1.53)	0.20 (0.04)	6.50 (1.78)	1.34 (0.53)
Reuters-RCV1	0.53 (0.14)	10.67 (1.84)	5.81 (2.25)	0.23 (0.09)	7.69 (2.04)	1.81 (0.79)
Industry	0.59 (0.12)	10.34 (1.43)	6.20 (1.71)	0.29 (0.09)	5.97 (1.52)	1.77 (0.61)

Table 2: **Difficulty measures for different corpora. Higher the value, more complex the problem. Values in brackets indicate std. deviation. The complexity is computed using the perceptron algorithm & uncertainty sampling & info. gain for feature selection.**

highly discriminatory are out of vocabulary in MT, making the classification problem even harder. Additionally, a multi-source setting (newswire, broadcast and multiple languages) can be more difficult than considering each source alone, and even the sum of each, as the vocabulary across sources differs depending on the MT and ASR systems used.

Subset of TDT3	C_f by class type			
	Events	*Nobel Awarded*	Subject	*Legal & Criminal cases*
Eng News	0.65	0.27	2.03	2.56
Eng ASR	0.95	0.14	2.02	2.78
MT News	1.38	3.25	2.12	2.61
MT ASR	1.22	3.48	1.50	2.03
Whole corpus	6.50	1.60	2.78	3.30

Table 3: **Difficulty of the TDT corpus when broken down by source and by category type.**

So far we have considered categories based on events in the TDT corpus; therefore *Hurricane Mitch* and *Hurricane George* were different categories. The TDT corpus is also annotated by broader subjects like *natural disasters*, *elections* etc, the feature complexity of which is indicated in the fourth column of Table 3. The fifth column shows the C_f values for an example topic - *legal and criminal cases*. The important features for classifying by subject are words like *court*, *law* etc., which do not suffer from as many MT and ASR errors making the difficulty of subject based classification about the same in each source type , and in the whole corpus (see the 4rth column of Table 3).

5 Correlation of Instance Complexity and Feature Complexity

Figure 2 had illustrated that that I_{pc} and F_{pc} of corpora computed using perceptron and information gain are strongly correlated. The individual problems in the corpora are also strongly correlated ($r = 0.79$ ($p < 2.2e^{-16}$)). Additionally f and i are also strongly correlated ($r = 0.613$ ($p < 2.2e^{-16}$)) and therefore C_i and C_f are also strongly correlated ($r = 0.682$ ($p < 2.2e^{-16}$)).

I_{pc} computed using SVM uncertainty sampling and F_{pc} computed using SVM LARS have very high correlation ($r = 0.95$), higher than the value obtained using perceptron and information gain, probably because they have the same underlying SVM learning and SVM LARS does a better job of feature ordering for the SVM learner than information gain does for perceptron.

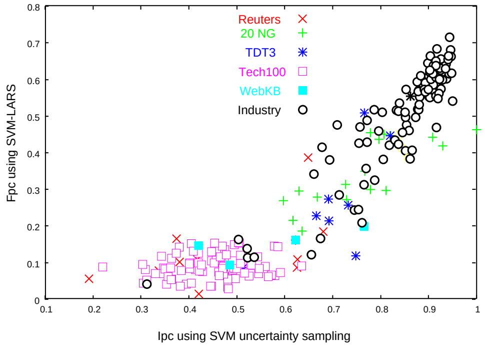


Figure 6: **Correlation between I_{pc} and F_{pc} using SVM and LARS. Correlation of instance complexity and feature complexity is independent of methods used to compute the two.**

We also experimented with random sampling for instance selection. The Table below shows the correlation coefficients for I_{pc} and F_{pc} for various combinations of classifiers, instance selection mechanisms and feature selection mechanisms for these 6 corpora.

classifier	Feature Sel.	Instance Sel.	r
SVM	LARS	Uncertainty	0.95
SVM	LARS	Random	0.88
Perceptron	Info. Gain	Uncertainty	0.81
Perceptron	Info. Gain	Random	0.79

Good instance and feature complexity measures should correlate well: a problem that requires a large number of instances, must require finding the right mix of weights for a relatively large number of features, and vice versa. We observe that our proposed measures satisfy this criterion.

6 When does interactive learning help?

In the previous section we saw that instance complexity and feature complexity are two sides of the same coin – a problem for which a few intelligently chosen instances can be used to build a good classifier is also one for which a few good features are very good predictors of class membership. Additionally, past work has shown that users can identify the most relevant features with reasonable accuracy[Raghavan *et al.*, 2005]. The same work found that labeling features is about 5 times faster than labeling documents. From these results we hypothesize that coupling intelligent feature selection with intelligent document selection should accelerate active learning. Asking users to come up with features apriori is quite difficult. Users found it difficult to determine the relevance of a feature, without having seen any relevant documents very difficult. Hence an interleaved approach of asking the users to mark relevant features in tandem with documents that they label is probably cognitively easier. Additionally, in some other work [5] we found that native English speakers could fairly easily point out machine translation errors of the kind discussed earlier where "Nobel" was consistently erroneously translated as "promises bell". That feedback significantly improved system performance.

[Raghavan *et al.*, 2005] built an active learning system for simultaneous document and feature feedback and showed that this dual feedback mechanism results in a much faster learning rate than traditional uncertainty sampling using a support vector machine as described in Section 3. In their InterActive Feature Selection algorithm, each time a document was picked by uncertainty sampling, the user was also asked to label 10 features. These features were obtained by ranking the features by their information gain scores on the current labeled set, where the labels asked on features were *relevant* (is the feature a discriminatory) or *non-relevant/don't know*. The labeled features were incorporated into the learning by scaling the value of that feature in all the instances. User feature feedback was simulated using an *oracle*, the details of which can be found in our paper. They found that actual users could emulate the oracle to an extent that resulted in as much improvement as can be achieved using the oracle.

Corpus	C_f	*speed*	
		Doc. f/b	Doc + term f/b
Tech100	0.14	0.556	**0.900**
WebKB	0.51	0.372	**0.714**
Reuters-21578	0.69	0.521	0.634
Enron	1.18	0.361	**0.651**
TDT3	1.34	0.244	0.296
Industry	1.77	0.109	**0.226**
RCV1	1.81	0.175	0.212
Newsgroups	2.32	0.187	**0.356**
BNC	2.97	0.322	0.411
Micro-average	1.37	0.280	**0.444**

Table 4: p_{al} of active learning with dual feedback. No.s in bold indicate statistically significant improvements computed using a two-tailed t-test at the 95% confidence interval

We measured speed of convergence of traditional uncertainty sampling (document feedback only) and that of the InterActive Feature selection system (document + term feedback) using the measure p_{al}. p_{al} is similar to the deficiency metric used in their paper and it measures the rate of convergence of active learning to its maximum achievable accuracy.

The values of p_{al} for both systems for all corpora are shown in Table 4. The two systems correspond to the "Act" and the "Oracle" systems in Figure 4 in their paper. To emphasize the early stage of learning like in that work, we computed p_{al} using only upto 50 labeled examples. In computing speed, we used the average F1 scores obtained after 30 different random initializations of the two initial training documents (one positive and one negative).

As per previous results, the p_{al} of document+term feedback is significantly higher than the speed of document feedback alone. We now explore the problems for which such a dual framework results in increased speed by plotting the difference in p_{al} versus feature complexity (C_f) (Figure 7). The improvement in speed due to the incorporation of term feedback in addition to document feedback is inversely related to feature complexity as seen in Figure 7 (r =-0.65). Table 4 shows a corpus-wise breakdown of the speed of active learning with and without feature feedback. p_{al} is improved by about 57% (last line of Table 4) on average.

The *faculty* class in WebKB shows significant improvement in speed(see Figure 7). For this problem, the keywords *faculty* and *professor* are sufficient to obtain 93% of the maximum achievable accuracy (90.05% F1). Both these terms appear for feature feedback within the first 5 iterations in all 30 trials. Similarly, for the Enron corpus, one of the folders is almost completely classified by the sender of the e-mail, *Wilson Shona* (there are some other folders that contain some e-mails by *Wilson Shona*). The algorithm recommends his e-mail id for feedback in the early iterations, resulting in significant improvements in performance. The *miscellaneous* category in the BNC corpus does not gain from term feedback whereas *arts/cultural material* does, because of discriminatory keywords like *opera, actor, theater* etc in the latter category that when marked relevant improve performance significantly. There are a couple of outliers like the RCV1 category *reserves* for which speed decreases by a large amount when term feedback is included. This may be because a fixed scaling factor of 10 for the selected features is used in the algorithm, which may not be appropriate for every problem. An interesting question is whether there are more robust methods for asking and taking feature feedback into account.

Using information gain has the drawback that highly correlated features get recommended for feedback, which tend not to add as much value. E.g., in the *Wilson Shona* category there are many correlated features corresponding to the header and signature of the sender. A forward selection algorithm such as LARS may be better for feature selection.

7 Related Work

In learning theory there has been work in determining the number of instances or queries needed to learn a concept, though the focus has often been on random samples and asymptotics [Angluin, 1992]. There have been studies on the relation between the number of features and the number of instances needed for learning (see e.g., [Hughes, 1968]), but they have assumed random samples and low dimensional spaces. Furthermore, we are concerned with computing empirical difficulty of actual problems. The classic "curse of di-

[5] which we will cite later for reasons of preserving anonymity

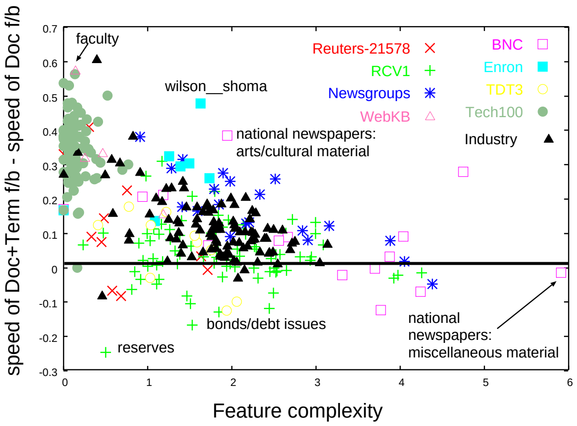


Figure 7: Difference in speed of Doc feedback and Doc+Term feedback as a function of C_f

mensionality” informally states that the higher the dimension of the problem, the harder the problem. However, our work goes after the inherent complexity of the problem. A large dimensional learning problem may be easy if only few features are required for learning it. We show here that actively picked examples reveal the complexity better, and we relate this to measures of feature complexity as well. Note that capturing the exact underlying complexity relates to maximum compression of a given string and is intractable. Thus the subject of this paper is exploring the utility of our approximate measures, which depends on the learning algorithm used as well as our chosen instance and feature selection techniques (and we report on some comparisons).

Ho and Basu [2002] defined a set of measures that captured the complexity of the geometry of the boundary for a few artificial and real binary classification problems of low dimensionality. In comparison, our work is in the domain of text classification, where a linear hyperplane is often effective making the geometry of the boundary less of an issue. For other domains where active learning is used [Tong and Chang, 2001] but where the classifier is not linear it is less clear whether our complexity measures can directly be used and we would be interested in exploring this question in the future. The difficulty in domains like text is that large amounts of training data may be needed in order to find the optimal hyperplane. Davidov et al [2004] developed a benchmark data set consisting of 100 text-classification problems with varying difficulty (accuracy ranging from 0.6 to 0.92). They also developed measures for predicting the difficulty of a problem, but this was in terms of its accuracy. Instead our focus is in understanding how many features or examples are needed to achieve the maximum accuracy.

Blum and Langley [1997] provide a good starting point to our work. They discuss the problem of selecting relevant examples and relevant features as two ways of gathering relevant information in a data set. They suggest using relevance as a measure of complexity. Their work is however theoretical and it is not clear how their measures may be used to quantify complexity for real world problems. They conclude their paper by stating the following *empirical challenge*: “... feature selection and example selection are tasks that seem to be intimately related and we need more studies designed to help understand and quantify this relationship. Much of the empirical work on example selection has dealt with low dimensional spaces, yet this approach clearly holds even greater potential for domains involving many irrelevant features.”

In this paper, we define a set of feature and instance complexity measures that can be used to quantify the difficulty of real world problems. We study the relationship between the measures in text classification, a domain with many irrelevant features. We find that instance complexity and feature complexity are highly positively correlated in Section 5, further corroborating the fact that our proposed measures indeed capture (approximate) the inherent feature and instance complexity of a problem.

There has been an increasing interest in techniques that use feature prior knowledge in addition to document labels in active learning and semisupervised settings [Dayanik *et al.*, 2006; Raghavan *et al.*, 2005; Wu and Srihari, 2004; Schapire *et al.*, 2002]. We have used our difficulty measures to better understand situations when such methods might work specially well. We have found that feature feedback accelerates active learning by an amount that is inversely related to the feature complexity of the problem. For low to mid range feature complexity problems, a few training documents combined with feature feedback can give a big improvement in accuracy with little labeled data. Many problems in our 9 corpora fall in a low to medium ($0 < C_f < 2$) range of complexity and stand to gain from such a dual feedback framework, automated email foldering being one such domain. Future work includes using these or similar measures to explain other observations, such as when other semi-supervised techniques may work well, as well as exploring methods for predicting the expected difficulty of a learning problem at the beginning stages of training (when few labeled data is available). This can inform the subsequent learning strategy taken.

8 Discussion for the Workshop

In this paper we systematically determined that text classification is a domain where a human in the loop is likely to accelerate learning. One is to use complexity measures to determine what other domains can benefit from interactive learning techniques. We also want to explore classification techniques that go beyond the instance based learning paradigm. We only touched the surface in breaking free from that paradigm by suggesting that learning can involve feedback on features. However, we thoroughly examined why feature feedback works and as a next step we want to determine how best to solicit feature feedback from users to accelerate learning, for example, the role of context in a users ability to determine useful features. We also wonder what other types of feedback humans may be able to provide – feedback on clusters perhaps? Are there corresponding notions of complexity, as in are there clusters for which if the learner knew some information about, learning would be accelerated? We ask how best to translate what the learner needs to know into a low cognitive load question for the human to answer. Conversely, from an HCI point of view certain pieces of information provided by a user seem like likely candidates for accelerating learning. But can classifiers built using current techniques absorb this information in a way that will actually enhance performance? Our ultimate goal is to design classifiers that can learn and adapt quickly with some feedback from users. We would like to bring up questions that sit at the bridge of machine learning and Human Computer Interaction at the workshop.

References

[Angluin, 1992] D. Angluin. Computational learning theory: survey and selected bibliography. In *STOC*, 1992.

[Bekkerman *et al.*, 2001] Ron Bekkerman, Ran El-Yaniv, Yoad Winter, and Naftali Tishby. On feature distributional clustering for text categorization. In *SIGIR*, 2001.

[Blum and Langley, 1997] A. Blum and P. Langley. Selection of relevant features and examples in machine learning. *Artificial Intelligence*, 97(1-2):245–271, 1997.

[Brank *et al.*, 2002] J. Brank, M Grobelnik, N Milic-Frayling, and D Mladenic. Feature selection using linear support vector machines. Technical report, Microsoft Research, 2002.

[Cohn *et al.*, 1990] D. Cohn, L.E. Atlas, and R. E. Ladner. Training connectionist networks with queries and selective sampling. *NIPS*, 1990.

[Dasgupta *et al.*, 2005] Sanjoy Dasgupta, Adam Tauman Kalai, and Claire Monteleoni. Analysis of perceptron-based active learning. In *COLT*, 2005.

[Davidov *et al.*, 2004] D. Davidov, E. Gabrilovich, and S. Markovitch. Parameterized generation of labeled datasets for TC based on a hierarchical directory. In *SIGIR*, 2004.

[Dayanik *et al.*, 2006] A. Dayanik, D. D. Lewis, D. Madigan, V. Menkov, and A. Genkin. Constructing informative prior distributions from domain knowledge in TC. In *SIGIR*, 2006.

[Godbole *et al.*, 2004] S Godbole, A Harpale, S Sarawagi, and S Chakrabarti. Document classification through interactive supervision of document and term labels. In *PKDD 04*, pages 185–196, 2004.

[Golub and et al, 1999] T. Golub and D. Slonim et al. Molecular classification of. cancer: Class discovery and. class prediction by gene expression and monitoring. *Science*, 286(15):531–537, 1999.

[Ho and Basu, 2002] T. K. Ho and M. Basu. Complexity measures of supervised classification problems. *IEEE Trans. Pattern Anal. Mach. Intell.*, 24(3):289–300, 2002.

[Hughes, 1968] G. F Hughes. On the mean accuracy of statistical pattern recognizers. *IEEE Trans. on Info. Theory*, 14:55–63, 1968.

[Joachims, 1997] Thorsten Joachims. A probabilistic analysis of the Rocchio algorithm with TFIDF for TC. In *ICML*, 1997.

[Keerthi, 2005] S. Keerthi. Generalized LARS as an effective feature selection tool for TC with SVMs. In *ICML*, 2005.

[Lewis and Catlett, 1994] D. D. Lewis and J. Catlett. Heterogeneous uncertainty sampling for supervised learning. In *ICML*, 1994.

[Lewis *et al.*, 2004] D. D. Lewis, Y. Yang, T. G. Rose, and F. Li. Rcv1: A new benchmark collection for text categorization research. *JMLR*, 5:361–397, 2004.

[Punyakanok and Roth, 2005] V. Punyakanok and D. Roth. Inference with classifiers: The phrase identification problem. *Computational Linguistics*, 2005. to appear.

[Raghavan *et al.*, 2005] H. Raghavan, O. Madani, and R. Jones. Interactive feature selection. In *IJCAI 05*, 2005.

[Schapire *et al.*, 2002] R. Schapire, M. Rochery, M. Rahim, and N. Gupta. Incorporating prior knowledge into boosting. In *ICML*, 2002.

[Tong and Chang, 2001] Simon Tong and Edward Chang. Support vector machine active learning for image retrieval. In *MULTIMEDIA '01: Proceedings of the ninth ACM international conference on Multimedia*, pages 107–118, New York, NY, USA, 2001. ACM Press.

[Wu and Srihari, 2004] X. Wu and R. Srihari. Incorporating prior knowledge with weighted margin support vector machines. In *Proceedings of KDD*, 2004.

[Wu and Zhang, 2004] Yimin Wu and Aidong Zhang. Feature selection for classifying high-dimensional numerical data. In *CVPR (2)*, pages 251–258, 2004.

Capturing and Disseminating Shareable Learning Experiences

Vive Kumar
Institute of Information Sciences and Technology, Massey University
63 Wallace St, Room 3D09, Post box 756, Wellington, New Zealand
V.S.Kumar@Massey.AC.NZ

Abstract

Educational technology innovations play a major role in engaging students to perform online educational tasks, across individual, institutional, and national boundaries. This research advances this conception of engagement by capturing and disseminating online learning experiences of students in an effort to intrinsically motivate them to share their best practices in learning. Students are encouraged to record and share their learning experiences using our ontology-oriented theory-centric software tool. In doing so, students not only observe the products of their learning but also the process of how they learnt. These unique and computationally formal recordings of learning experiences not only allow educators to observe how learners learn, but also provide opportunities for learners to reflect on their understanding of meta-cognitive processes that they employed or neglected in their learning. Further, these recordings feed our software system to autonomously analyse students' learning behaviour and to actively promote self- and co-regulation among learners. This paper presents the need for such a system, the architecture of the system, and concludes with key experimental observations from the prototypes and future exploration possibilities.

1 Introduction

Research is ongoing in a number of frontiers to understand how learning infrastructure, cognitive tools, and social experiences sustain learning. Over the last two decades, a number of technological solutions have been employed in online learning environments. Computer-Aided Instruction (CAI) brought the content of learning to an interactive electronic format. Computer-Mediated Communication (CMC) techniques enhanced human-computer interaction between the educational system and the learners. Computer-Supported Collaborative Learning (CSCL) and Computer-Supported Cooperative Work (CSCW) explored the social nature of learning. Intelligent Tutoring Systems (ITS) enabled educational systems to deliver informed and pedagogically sound instructions. Courseware Management Systems (CMS) offered features aimed at optimal online delivery of courses. In general, from a student's perspective, these systems facilitate him/her to engage in different types of learning interactions leading to different types of learning experiences. The difference between 'learning interaction' and 'learning experience' will be discussed below.

Many online learning environments allow learners to record their learning interactions. Interactions are task specific and they include encoded online/offline discussions, user manipulations within software applications, video recordings, eye-tracking, and so on. For instance, IHelp [Brooks et al., 2006] explicitly asks permission from students to record their collaborative interactions with other students, while gStudy [Winne et al., 2005] implicitly records trainee interactions with a troubleshooting system. Many online environments are capable of recording learner interactions so that researchers can recode these interactions, interpret sequences of interactions, analyse characteristics of interpretations, and offer explanations as to plausible reasons that motivated these patterns of interactions. For example, gChat [Winne et al., 2005] records interactions when learners are engaged in specific collaboration tasks.

In most of these online environments, interactions are recorded primarily for the consumption of researchers. That is, the job of the online environment is exactly like that of a video camera – to record data, and leave the processing of the recorded data to researchers. Researchers identify a set of tags to encode the interactions with. For instance, a group of utterances spoken by an online helper could be tagged as "helper provides clues". Normally, researchers map these tags, posthoc, offline, onto observed interactions.

Alternatively, researchers identify tags up front and embed the tags into the online system. In such systems, online interactions can be tagged as and when they happen. However, such real-time tagging tends to be limiting in the sense that, in most cases, it is difficult to envision and track all possible combinations of interaction sequences of learners. That is, some sequences of observed interactions may not belong to any of the predefined tags and hence require new tags.

In many cases, researchers independently hand-code these tags offline, to compensate for what the system could not automatically tag, as well as to provide unbiased, third-party confirmation for the encoding.

Moreover, it is quite impractical to record learner interactions at microscopic levels of granularity. For example, in gStudy [Winne et al., 2005], learner interactions are recorded at the level of macroscopic events (such as, highlighting a sentence, linking a word with a video clip, and so on) rather than at the level of keystrokes and pixel-capture of mouse movements. Still, automatic tagging is becoming popular among researchers who prefer to run tightly-controlled real-time experiments using rather powerful computers for participants to interact with.

Another generation of online learning environments offer to open-up recorded interactions to students so that students themselves can inspect [Collins et al., 1997] their interactions with the system. Students may open-up these interactions for the inspection of their colleagues. Many inspectable systems presented interactions without exposing the tags while a few systems attempted to even reveal tags along with their associated interactions to students in a visualisable format [Zapata-Riviera, 1999].

Yet another generation of online learning environments attempt to dynamically track learner interactions, dynamically encode interactions, dynamically interpret sequences of encoded interactions, and dynamically offer explanations and feedback (as in MI-EDNA [Shakya, 2005] and EPSILON [Soller, 2004]). Human-Computer Interactions Institute[1] at Carnagie Mellon University is a front-runner in the design and development of such *model-tracing* systems. However, all these dynamisms are constrained to work only within well-defined task domains. That is, the interpretation of a sequence of learner interactions in one system could be very different from an interpretation of the same set of interactions by another system, assuming first of all that both systems have valid 1-to-1 mappings for each other's tags. LORNET[2] attempts to overcome such interoperability issues in mapping tags across learning objects and systems.

Scalability of tag-encoding mechanism is one of the key challenges of the Semantic Web community. To scale the tag encoding mechanism one can resort to ontologies. Ontology provides a common and formal interpretation of the vocabulary. The primary goal of these scalability efforts is to capture learning interactions and interpret learning interactions in a consummate machine shareable form. That is, learning interactions and their interpretations are made intelligible not only to humans (researchers and students) but also to software systems.

We are currently working on a novel approach to online learning that captures what we call *machine-shareable learner experience* (MSLE).

2 Shareable Learning Experience

So far, our discussion revolved mostly around learning interactions. We will now introduce the notion of Machine Shareable Learning Experiences.

In online learning, interactions of a student can be defined as his/her observed actions[3] within the scope of a well-defined task. For example, the following sequence of actions of a student, {'browse to end of page', 'highlight phrase', 'search for phrase', 'seek help', 'search for phrase', 'failed link', 'seek help', 'failed link', 'search for phrase', and 'linked phrases'}, forms a sample interaction, with respect to the task of 'summarising'.

Learning experiences, on the other hand, can be defined as formal recordings of learning processes and learning products of learners, within the scope of a well-defined learning task. A machine shareable learning experience encases five components. First, it contains ontologised observed interactions. Second, it contains ontologised representation of the environment in which learning happened. Third, it contains ontologised learning strategies reflected on by the student and peers. Fourth, it contains ontologised formal reviews of learning products and learning processes by the instructor. Fifth, it contains ontologised mixed-initiative review of learning products and processes by the theory-centric software system.

2.1 Ontologising Interactions

In a simplified sense, ontology provides an extendable and shareable framework to capture a common vocabulary, common enough for both humans and machines to have the same interpretation of an encoding. Ontology includes machine-interpretable definitions of basic concepts in the domain and the relations that exist among them [Murray, 1999]. The power of ontologies rests with the ability to represent knowledge explicitly (as concepts, properties, and constraints); to encode semantics (as relations, meta-data, and inference); and to allow for a shared understanding of the represented formal knowledge within and in-between humans and

[1] http://www.hcii.cmu.edu/

[2] http://www.lornet.org/

[3] Both online actions and encoded offline actions

the machines. Ontology is a powerful representation scheme that can describe and model a vast range of complex systems using concepts and relations. Ontology has been extensively employed in the domain of online learning environments in the past decade. The uses of ontology for course authoring, knowledge engineering, and domain instantiation have been some of the popular areas of research [Aroyo, 2002].

For example, the ontology of Zimmerman's SRL model [Shakya, 2005; Zimmerman, 2002] comprises of three top-level concepts – forethought, performance, and self-reflection. A number of properties are associated with these concepts, properties such as 'hasGoal', 'hasMotivation', 'hasPlannedStages', 'hasStrategicPlan', and so on. The forethought phase has a sub-concept called optimal-initial-phase that inherits some of the properties of the parent, forethought. Importantly, optimal-initial-phase also defines "necessary & sufficient" conditions so that a sequence of interactions can be classified and instantiated under optimal-initial-phase. That is, by computing the "necessary & sufficient" conditions, one can assert whether a student has successfully undergone the optimal-initial-phase. For instance, the "necessary & sufficient" conditions for optimal-initial-phase are 'hasGoal=True', 'hasMotivation=True', and 'hasStrategicPlan=True'. Learner interactions can be parsed and analysed to deduce whether a student has set himself/herself a goal to pursue, whether he/she indicated to have enough motivation to achieve the goal, and whether he/she has specified a strategic plan to define the process that leads to the goal. Thus, by enabling the system to formally interpret interactions in terms of ontological components of SRL, one can contend that the notion of SRL is intelligible to the system.

The ontological representation of observed interactions refers to instantiation of the observed interaction data into an ontological form. Manual instantiation is tedious, cumbersome, and error-prone. Alternatively, observed interactions (aka, trace data), preferably recorded in XML format, can be automatically or semi-automatically instantiated. For example, MI-EDNA meta-tags lesson contents of 'Java Programming' in a semi-automatic manner using DocBook[4] and instantiates the content ontology using XSLT style sheets. On the other hand, learner interactions were logged in an XML file at real-time, and in parallel, were directly fed into the ontology instantiator [Shakya, 2005].

Once ontologised, the encoded interactions are shareable with other humans as well as with other systems. That is, if we share Zimmerman's SRL ontology with a fellow researcher, we can expect that researcher to precisely understand how we recognize phase-specific activities from learner interactions. Similarly, if we share the instantiated ontology with another ontological system, we can be assured that that system will interpret sequences in exactly the same way our system would. Thus, ontological representation enables shareability of learning experiences while preserving the underlying semantics.

2.2 Ontologising Learning Environment

The environment of learning refers to a mapping[5] between a) what has been learnt with respect to the ontologised model of SRL and b) the software/hardware set up in which learning happened. For example, if a student learnt to program in Java, the IDE (Integrated Development Environment), task model, concept map, applicable programming strategies, and other relevant Java programming resources form the software/hardware setup [Rao et al., 2006]. The task model includes expected and observed activity sequences of students, at task level. The concept map is a collection of concepts associated with the task and undirected links of informal relations among the concepts. Programming strategies, among others, include coding conventions, code engineering practices, and debugging styles of programmers. Rao et al., outline the design of a Programming Style ontology that present concepts associated with programming (e.g., commenting, debugging, compiling, and coding), relations among these concepts (e.g., 'less number of compiles', 'coding speed'), and constraints on concepts and relations.

The definition of the environment purposefully excludes information pertaining to students, instructors, peers, and the theory-centric mixed-initiative system, which are complex enough to warrant their own individual components.

2.3 Ontologising Strategy Reviews by Learner

The learning strategies reviewed/reflected on by a student refers to a mapping between a) what has been learnt with respect to the ontologised model of SRL, b) associated/observed metacognitive activity sequences, and c) explicit commentaries from students that explain theory-specific and content-specific linkages between what has been learnt and the sequences. The types of information recorded by this component comprise of students' comments on 1) abstraction of sequences to strategies, 2) strategies missed out by students, 3) identification of popular strategies among groups of students, 4) mapping between strategies, learning objectives, and grades/performance, and 5) mapping between strategies and students' background knowledge. Students' background knowledge is simply a learner

[4] www.docbook.org

[5] Throughout this paper, the term mapping indicates establishment of either a generic relation (e.g., a datatype property in an ontology created using Protégé) or a causal relation (e.g., a directed probabilistic relation in a Bayesian Belief Network created using JavaBayes), or both, between two entities.

model, conceived to be an inspectable computational model of students' task knowledge, cognitive load capacity, current workload, motivational estimates, learning habits, and inclination to collaborate. The learner model is designed to be an ontology, thus imposing an ontological representation of student reviews as well.

2.3 Ontologising Instructor Review

A formal review of products and processes of student learning by the instructor addresses a mapping between a) what has been learnt with respect to the ontologised model of SRL, b) the associated/observed metacognitive activity sequences of students, and c) instructor commentaries on theory-specific and content-specific linkages between what has been learnt and the sequences.

2.4 Ontologising System Review

A mixed-initiative review of products and processes of student learning by the theory-centric software system, not so surprisingly, is very similar to that of the instructor review. The key difference, however, is that the software system performs its review in a self-sufficient manner, preferably in consultation with the instructor (aka, human-in-the-loop) when the consultation is necessary in its computational decision-making process.

2.5 Summary

The raw recording of a learning session, its ontologised learning environment, an ontologised review/reflection by the student(s), a formal review by the instructor(s), and a mixed-initiative review by the software system, all put together, create a consummate shareable learning experience. It is consummate on account of

- a semantics-preserving transformation of raw student-system interactions (of a single learning session) to ontology and/or causal network
- a portfolio of overlapping reviews by student, instructor, and the software system
- a grounding of mappings in self-regulation
- an inspectable ontology of the learner model
- a mixed-initiative software system that reactively and proactively preserves and shares the recorded learning experience for the consumption of humans and other software systems

Having set forth the conceptual foundation, the next section describes the architecture of the system.

3 The Architecture

The implementation efforts of this research have been funded by two projects – Learning Kit[6] and LORNET[7]. The Learning Kit project is designed to support students in reflecting their self-regulated learning (SRL) strategies. It aims to develop a study tool called gStudy, which is a cross-platform software tool for researching the underlying processes of learning – importantly in reading, writing, collaboration, and programming. gStudy allows learner interactions to be captured and analyzed to recognize tactics and strategies employed by students during their learning process to reach their learning objectives [9].

Our system, named MII↔RA, has been developed as an extension to gStudy [Winne, 2005] to systematically capture semantically-rich learning experiences. For example, MII↔RA captures real-time learner interactions when students are engaged in reading activities from within gStudy. The raw data consists of interactions including browsing, highlighting, compiling code, text chatting, indexing, concept mapping, note taking, reviewing, collaborating, and so on. Similarly, interactions of programmers in an IDE [Rao et al., 2006] or interactions of students engaged in a writing assignment can also be captured at real-time and recorded.

Further, raw data from other applications that students use can also be processed offline. Presently, we are importing data from two additional applications – one, a student ePortfolio application, and the other, an online collaboration application. This section describes the design of the system in terms of the technical, the functional, and the mixed-initiative architectures.

3.1 Technical Architecture

The technical design of the system consists of four main components: the underlying ontology, the ontology instantiator, the inference engine, and the interface. The technical architecture of the system is presented in Figure 1.

These raw interaction data observed in the system are instantiated/recorded in domain-specific ontologies.

Presently, the system contains 5 ontologies – content, interaction, learner, time, and strategies. These ontologies, put together, serve as the connector for the rest of the modules in the architecture, as an area of information exchange for the other three components.

Further, the ontologies also coordinate the actions arising from the other three components.

The raw data instantiation into the ontological structure can be fully- or semi-automated; that is, the instances

[6] http://www.learningkit.sfu.ca

[7] http://www.lornet.org

can either populate the ontology without any human intervention or with minimal human manipulation. Presently, the system employs fully-automated instantiation to transfer raw data from real-time learner interactions into the ontologies. The learner interaction data, observed from within gStudy, is first captured in an XML format. An XML parser is then used to browse the XMLized data and create the corresponding ontology instances in OWL format. The XML parser instantiates ontology concepts and establishes ontology relations between instances that have just been created in the ontology based on the constraints and the restrictions predefined in the ontology.

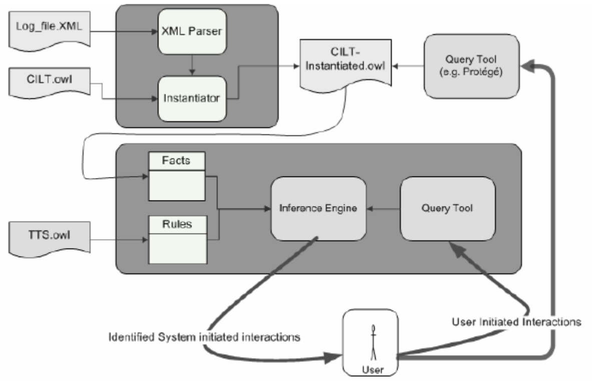


Figure 1 : Technical Architecture

The system, at present, uses three types of inference engines: one based on Description Logic, the second on Production Rules, and the third on Bayesian. The inference engines provide a gateway for the system to reason with the ontological knowledge.

The ontological knowledge contains the interaction data of students and patterns of SRL. In addition, the ontological knowledge also contains rules that interpret patterns in learner interactions. Further, the ontology is translated into a Bayesian Belief Network to exact probabilistic inferences.

Presently, MII↔RA uses the instantiated learner interaction to recognize patterns of tactics and strategies enacted by the learners. The production rules match tactics and strategies to specific phases and variables of the SRL model [Shakya, 2005].

The interface of the system enables students to query the system and extract information about their learning styles and their learning patterns, particularly in comparison with the styles and patterns of their classmates.

3.2 Functional Architecture

The functional architecture of the system describes the flow of information across the components within the system. The driving force behind the system's functionality is the theory of SRL.

Students evolve their learning strategies, mostly without being conscious about it, as they progress through their learning process. We contend that technology in educational applications can enhance how effectively learners interact with learning environments based on guidelines provided by theories of Educational Psychology. Self-Regulated Learning (SRL) is a theory that concerns how learners develop learning skills and how they develop expertise in using learning skills effectively. It comprises of a set of strategies employed by students to regulate their own learning processes. It arises from two key observations. First, learners' goals for learning take precedence over goals set by teachers, authors of curricula, and developers of learning objects. Second, learners are in charge of how they learn. They choose which study tactics and learning/problem-solving strategies to use as they strive to achieve their goals. Normally, learners set unsuitable goals, have a limited repertoire of learning skills, do not use learning skills they have, and frequently need extensive help to manage learning and collaborative tasks [Winne, 1997; Winne & Hadwin, 1998].

The system is designed to recognise strategies students employ over a period. Self-regulation transforms mental abilities to academic skills, which involves selective use of specific processes that must be personally adapted to each learning task. Such cognitive transformations are explicitly captured in the ontology and are shared with other key modules in the system for adaptation. SRL processes include: a) setting proximal goals; b) adapting strategies to attain goals; c) monitoring performance for signs of progress; d) restructuring contexts to make them compatible with goals; e) managing time; f) self-evaluating one's methods; g) attributing causation to results; and h) adapting future methods. Self-regulating students focus on how they activate, alter, and sustain specific learning practices in social contexts as well as solitary contexts, and this process is functionally captured within the data flow of the system.

Presently, MII↔RA represents and reasons with two SRL models - Zimmerman's three phase model [Zimmerman, 2002] and Winne and Hadwin's four phase model [Winne and Hadwin, 1998; Winne, 2001]. Tracing a student's individual study habits with respect to a specific model of self-regulation is one of the primary goals of our research.

As a precursor to MII↔RA, a software system called MI-EDNA [Shakya, 2005] has been developed in the domain of Reading. For instance, in helping students to adapt strategies to attain reading goals, MI-EDNA provides interfaces for them to set their goals, to monitor their progress, to identify tactics and strategies they use in attaining specific goals, to compare their strategies with the strategies employed with their peers, to explore strategies advocated by the instructor, and to compare their performances after they adopted a new set of strategies.

The functional architecture in Figure 2 depicts the flow of functionality of the system in the domain of Programming called MICE. In MICE, the flow starts with programmer interactions in an IDE. The current software system uses BlueJ as the IDE. These interactions trigger events to instantiate appropriate elements in the ontologies. Changes in ontologies trigger execution of rules.

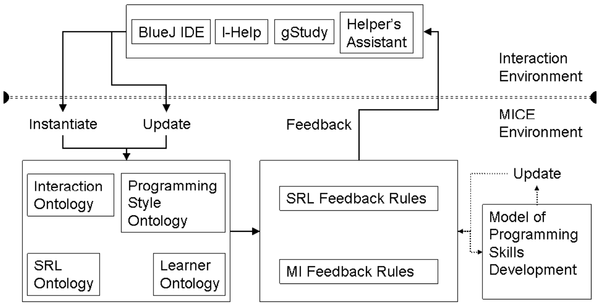


Figure 2 : Functional Architecture

Specifically, the purpose of MICE rules are threefold:

- rules are used to computationally recognize programming style components;
- rules are used to identify opportunities for system-initiated interaction such as programmers spending too much time debugging a piece of code or programmers consistently failing to construct task models;
- rules are used to engage programmers in mixed-initiative dialogues with MICE. For example, the MICE system and the programmer can engage in a well-defined, role-playing conversational model when situation warrants it.

The feedback of MICE are marshaled at real-time to the BlueJ IDE as well as to external systems including IHelp [Brooks et al., 2006] and gStudy [Winne et al., 2005]. In return, events observed at external systems can be recorded in the interaction ontology. For example, gStudy events related to links-creation, highlighting, browsing, and searching can be recorded in the interaction ontology.

3.3 Mixed-Initiative Architecture

Mixed-Initiative Interactions attempt to model an interaction strategy where conversants (students or the system) contribute appropriate information, when it is best suited, towards mutually negotiated goals [Hearst, 1997]. At any one time, one conversant might have the initiative, controlling the interactions, while the others contribute to the interactions as required. Mixed-Initiative interactions are driven by conversants' relative knowledge, preferences, and task toward common, partially shared, and individual goals.

Humans naturally engage in Mixed-Initiative Interactions. The natural flow of communication between a learner and an instructor involves interaction initiatives originating from both sides. Our research aims to capture the process of this natural phenomenon and induce similar interactions between a learner and the system. However, we restrict the scope of interactions to the domain of SRL.

Mixed-initiative systems [Hearst, 1999] exhibit various degrees of involvement pertaining to the initiatives taken by the user or the system. One of the key elements for successful mixed-initiation is the ability of the system to recognize opportunities for initiatives based on well-founded theoretical principles.

The system passively observes learner interactions, recognizes opportunities for initiatives, and actively but non-intrusively initiates interactions. The opportunities for initiatives are recognized based on the principles of SRL regulated by the scaffolding/fading techniques that serve as the basis of the interaction model.

Students can query and understand the educational theories and learning practices behind the guidance/feedback/initiatives provided by the system.

The production rules match the strategies to specific phases and variables observed in the SRL models. As production rules trace learner interactions as an overlay of the SRL model, the path traced so far (and possibly the projection of the traced path) help interpret the learner interactions with respect to the model. Presently, the system initiates interaction with the students based on the following:

- Guidance to learners on navigation of content (content scaffold)
- Methods they use to study/solve problems (SRL scaffold)
- How much they have learned (knowledge of results)
- How learner's peers study and what they score on tests (normative scaffold)
- Supporting learners based on the context of interaction (context scaffold)

Three separate prototypes were built to test the exigencies our conceptual framework in three different domains – Reading, Writing, and Programming. The next section outlines some of the key results of experimentation with the prototypes.

4. Key Observations and Conclusion

This research was inspired by the need to 1) be able to record the processes of learning at various levels of granularity and associate sub-sequences these processes to levels of abstraction of appropriate educational theories, and 2) semantically encode and reuse best-practices of human-in-the-loop approach [Kumar, 2002] in assisting online helpers. There are some preliminary yet satisfactory results to report.

Kumar et al. [2005] contend that current software engineering practices emphasise on external constraints that programmers deal with more often than internal metacognitive abilities that they could nurture to greater productivity. In an effort to show that by observing programming practices from a theoretical viewpoint one could infer the importance of nurturing metacognitive skills involved in programming, Samin [2004] experimentally showed the treatment group that used a coding environment enhanced with metacognition support performed significantly better (mean 54.39 out of 100) than the control group that used the same coding environment without metacognition support (mean 43.41 out of 100). Samin further detected various stages of programming – Warm-up (T (1, 40) = -3.071, p<0.05), Thinking (T (1, 40) = -2.096, p<0.05), and Coding. That is, less than 50% of the programming process (for both control and treatment groups) was spent in coding. The rest of the time was spent in applying learning operations of searching, monitoring, assembling, rehearsing, and translating to build metacognitive products. This research presents an excellent opportunity *for automatic analysis of programmer behaviour and reasoning about mixed-initiative interaction while coding*[4].

Shakya [2004] analysed data obtained from an experiment and showed that MI-EDNA was able to recognize and count the occurrences of learning tasks tackled by students, learning strategies employed by students within each learning task, mappings between sub-sequences of learner interactions and phases of SRL, and system-initiated feedback/scaffolding prospects. That is, the system is in a position to explicitly and consistently related student interactions to pieces of a cognitive theory. In line with the goals of the workshop, this research presents an excellent opportunity *to exploit the need for anticipatory user interfaces to unobtrusively sense the user's behavioural cues, to learn and to adapt automatically to the particular user behavioural patterns and the context in which the user acts*[8].

Presently, we are devising experiments to observe and analyse the impact of mixed-initiative/human-in-the-loop approach in the task domains of Programming and Writing.

Our most important and conclusive work lies in subjecting MSLEs to *mixed-initiative interpretation* (MII) based on the SRL model and in return subjecting the SRL model to *rationale-assessment* (RA) using Bayesian learning. Mixed-Initiative Interpretation implies the ability of a system to consistently interpret interactions across multiple task domains at multiple levels of abstraction. This is made possible with the introduction of task-independent theory-centric encoding of interactions. Rationale-Assessment implies the ability of a system to assess the validity of the theoretical-basis based on interpretations that it has made so far across different task domains and episodes. MII and RA feed on each other. MII is dependent on the theoretical basis acceptable to RA and RA is dependent on instances of MII that have been interpreted so far. The validity can be asserted by 'learning' causal links and values in a Bayesian representation of Zimmerman's SRL model. The MII↔RA cycles are self-sustainable and, minimally, online learning environments that encompass the MI-RA framework can be termed sustainable.

Our research creates opportunities to improve the quality of online learning. The core strength of our system resides with its ability to enforce a tight integration between the learning practices and education theories. With the deployment of mixed-initiative interactions, based on Self-Regulated Learning principles, a learner-conducive communication occurs between the software and the learner. Inferring the instantiated ontology yields theory-oriented explanations for mixed-initiative interactions. These experiences in products and processes of learning can be shared across individuals and institutions.

Acknowledgments

This research was supported by SSHRC[9]'s Learning Kit project and NSERC[10]'s LORNET project. A number of student researchers contributed to this work, notably, Behzad Samin, Jurika Shakya, Shilpi Rao, David Brokenshire, Pat Lougheed, Mayo Jordanov, Liam Doherty, Max Whitney, and Brittney Bogyo.

References

[Aroyo et al., 2002]. Lora Aroyo, Darina Dicheva, Alexandra Christea. “Ontological Support for Web Courseware Authoring”. In: ITS02 Intelligent Tutoring Systems. Volume LNCS 2363., Springer, 270–280, 2002.

[8] Partially quoted from the description of the AI for Human Computing workshop proposal at IJCAI 2007

[9] Social Sciences and Humanities Research Council of Canada

[10] Natural Sciences and Engineering Research Council of Canada

[Brooks et al., 2006]. Brooks, C., Panesar, R., Greer, J., "Awareness and Collaboration in the iHelp Courses Content Management System". In Proceedings of the 1st European Conference on Technology Enhanced Learning, Crete, Greece, 2006 (forthcoming).

[Collins et al., 1997] Jason A. Collins, Jim E. Greer, Vive S. Kumar, Gordon I. McCalla, Paul Meagher, and Ray Tkatch. “Inspectable User Models for Just-in-time Workplace Training”. International Conference on User Modelling, 327-337, 1997.

[Greer et al., 2000] Greer, J.E., McCalla, G.I., Cooke, J.E., Collins,J.A., Kumar, V.S., Bishop, A.S., Vassileva, J.I. “Integrating Cognitive Tools for Peer Help: the Intelligent IntraNet Peer Help-Desk Project”. In S. Lajoie (Ed.) *Computers as Cognitive Tools: The Next Generation*, Lawrence Erlbaum , 2000, 69-96.

[Hearst, 1999] Marty Hearst. “Trends and Controversies - Mixed-Initiative Interaction”. IEEE Intelligent Systems, 14–24, 1999.

[Kumar, 2002] Vive Kumar. “Embedding Human Reasoning in Soft Computing”, International Conference on Hybrid Intelligent Systems (HIS’02), Santiago, Chile, pp 625 - 633, 2002.

[Kumar, 2004] Vive Kumar, “An Instrument for Providing Formative Feedback to Novice Programmers”. Annual Meeting of American Educational Research Association (AERA), Division I – Education in the professions, Paper session – Relationship between teaching and learning (13.032), San Diego, CA, USA, pp.71, 2004.

[Kumar et al., 2005] Vive Kumar, Philip Winne, Allyson Hadwin, John Nesbit, Dianne Jamieson-Noel, Tom Calvert, Behzad Samin. Effects of self-regulated learning in programming, IEEE International Conference on Advanced Learning Technologies (ICALT 2005), Kaohsiung, Taiwan, 5-8 July, 383 - 387, 2005.

[Murray, 1999] T. Murray. “Authoring Intelligent Tutoring Systems: An analysis of the state of the art”. International Journal of Artificial Intelligence in Education, 10:98 – 129, 1999.

[Rao et al., 2006] Shilpi Rao, Vive Kumar, Marek Hatala, Dragan Gasevic. “MICE: Capturing Programming Styles in a Mixed-Initiative Coding Environment”. I2LOR – Conference on Intelligent Interactive Learning Object Repositories, 2006, (forthcoming).

[Samin, 2004] Behzad Samin. “Effects of Self-Regulated Learning in Programming”. MSc Thesis, Simon Fraser University, Canada, 2004.

[Shakya, 2005] Jurika Shakya, Knowledge Engineering and Knowledge Dissemination in Mixed-Initiative Ontological Framework, MSc Thesis, Simon Fraser University, Canada, 2005.

[Soller, 2004] Amy Soller. “Computational Modeling and Analysis of Knowledge Sharing in Collabrative Distance Learning”. User Modelling and User-Adapted Interaction, 14, 351-381, 2004

[Winne, 2001] Philip Winne. “Self-Regulated Learning Viewed from Models of Information Processing”. In B. Zimmerman and D Schunk (Eds), Self-regulated learning and academic achievement: Theoretical perspectives. 2 Edn. Hillsdale, NJ: Erlbaum, 153–189, 2001.

[Winne and Hadwin, 1998] Philip Winne and Allyson Hadwin. “Studying as Self-Regulated Learning”. In. D. Hacker, J. Dunlosky & A. Graesser (Eds.). Metacognition in educational theory and practice, 277–304, 1998.

[Winne, 1997] Philip Winne. “Experimenting to Boot-Strap Self-Regulated Learning”. In: Journal of Educational Psychology. Vol. 89. American Psychological Association, 397–410, 1997.

[Winne et al., 2005] Philip Winne, John Nesbit, Vive Kumar, Allyson Hadwin, Susanne Lajoie, Roger Azevedo, Nancy Perry. “Supporting Self-Regulated Learning with gStudy Software: The Learning Kit Project”. In International Journal of Technology, Instruction, Cognition and Learning, 3, 105-113, 2005.

[Zapata-Riviera et al., 1999] Diego Zapata-Rivera, Eric Neufeld, and Jim Greer. “Visualization of Bayesian Belief Networks” In IEEE Visualization, Late Breaking Hot Topics Proceeding, pages 85 – 88, 1999.

[Zimmerman, 2002] Barry Zimmerman. “Becoming a Self-Regulated Learner: An Overview”. Theory into Practice. Volume 41, 2002.

SmartWeb Handheld — Multimodal Interaction with Ontological Knowledge Bases and Semantic Web Services

Daniel Sonntag, Ralf Engel, Gerd Herzog, Alexander Pfalzgraf,
Norbert Pfleger, Massimo Romanelli, Norbert Reithinger
German Research Center for Artificial Intelligence
66123 Saarbrücken, Germany
firstname.lastname@dfki.de

Abstract

SMARTWEB aims to provide intuitive multimodal access to a rich selection of Web-based information services. We report on the current prototype with a smartphone client interface to the Semantic Web. An advanced ontology-based representation of facts and media structures serves as central description for rich media content. Underlying content is accessed through conventional web service middleware to connect the ontological knowledge base and an intelligent web service composition module for external web services, which is able to translate between ordinary XML-based data structures and explicit semantic representations for user queries and system responses. The presentation module renders the media content and the results generated from the services and provides a detailed description of the content and its layout to the fusion module. The user is then able to employ multiple modalities, like speech and gestures, to interact with the presented multimedia material in a multimodal way.

1 Introduction

The development of a context-aware, multimodal mobile interface to the Semantic Web [Fensel *et al.*, 2003], i.e., ontologies and web services, is a very interesting task since it combines many state-of-the-art technologies such as ontology development, distributed dialog systems, standardized interface descriptions (EMMA[1], SSML[2], RDF[3], OWL-S[4], WSDL[5], SOAP[6], MPEG7[7]), and composition of web services. In this contribution we describe the intermediate steps in the dialog system development process for the project SMARTWEB [Wahlster, 2004], which was started in 2004 by partners from industry and academia.

In our main scenario, the user carries a smartphone PDA and poses closed and open domain multimodal questions in the context of football games and a visit to a Football Worldcup stadium. Many challenging task such as interaction design for mobile devices with restricted computing power have to be addressed: the user should be able to use the PDA as a question answering (QA) system, using speech and gestures to ask for information about players or games stored in ontologies, or other up-to-date information like weather forecast information accessible through web services, Semantic Web pages (Web pages wrapped by semantic agents), or the Internet.

The partners of the SMARTWEB project share experience from earlier dialog system projects [Wahlster, 2000; 2003; Reithinger *et al.*, 2005b]. We followed guidelines for multimodal interaction, as explained in [Oviatt, 1999] for example, in the development process of our first demonstrator system [Reithinger *et al.*, 2005a] which contains the following assets: *multimodality*, more modalities allow for more natural communication, *encapsulation*, we encapsulate the multimodal dialog interface proper from the application, *standards*, adopting to standards opens the door to scalability, since we can re-use ours as well as other's resources, and *representation*. A shared representation and a common ontological knowledge base ease the data flow among components and avoids costly transformation processes. In addition, semantic structures are our basis for representing dialog phenomena such as multimodal references and user queries. The same ontological query structures are input to the knowledge retrieval and web service composition process.

In the following we demonstrate the strength of Semantic Web technology for information gathering dialog systems, especially the integation of multiple dialog components, and show how knowledge retrieval from ontologies and web services can be combined with advanced dialogical interaction, i.e., system-initiative callbacks, which present a strong advancement to traditional QA systems. Traditional QA realizes like a traditional NLP dialog system a (recognize) - analyze - react - generate - (synthesize) pipeline [Allen *et al.*, 2000]. Once a query is being started, the information is pipelined until the end, which means that the user-system interaction is reduced to user and result messages. The types of dialogical phenomena we address and support include reference resolution, system-initiated clarification requests and

[1] http://www.w3.org/TR/emma

[2] http://www.w3.org/TR/speech-synthesis

[3] http://www.w3.org/TR/rdf-primer

[4] http://www.w3.org/Submission/OWL-S

[5] http://www.w3.org/TR/wsdl

[6] http://www.w3.org/TR/soap

[7] http://www.chiariglione.org/mpeg

pointing gesture interpretation among others. Support for underspecified questions and enumeration question types additionally shows advanced QA functionality in a multimodal setting. One of the main contributions is the ontology-based integration of verbal and non-verbal system input (fusion) and output (system reaction).

The paper is organized as follows: we begin with an example interaction sequence, in section 3, we explain the dialog system architecture. In section 4, the ontological knowledge representation and web service access is described. Section 5 then gives a description of the underlying language parsing and discourse processing steps, and their integration. Conclusions about the success of the system so far and future plans are outlined in section 6.

2 Multimodal interaction sequence example

The following interaction sequence is typical for the SMARTWEB dialog system.

(1) **U:** “When was Germany world champion?”

(2) **S:** “In the following 4 years: 1954 (in Switzerland), 1974 (in Germany), 1990 (in Italy), 2003 (in USA)”

(3) **U:** “And Brazil?”

(4) **S:** “In the following 5 years: 1958 (in Sweden), 1962 (in Chile), 1970 (in Mexico), 1994 (in USA), 2002 (in Japan)” + [*team picture, MPEG-7 annotated*]

(5) **U:** Pointing gesture on player *Aldair* + “How many goals did this player score?”

(6) **S:** “Aldair scored none in the championship 2002.”

(7) **U:** “What can I do in my spare time on Saturday?”

(8) **S:** “Where?”

(9) **U:** “In Berlin.”

(10) **S:** *The cinema program, festivals, and concerts in Berlin are listed.*

The first and second enumeration questions are answered by deductive reasoning within the ontological knowledge base modeled in OWL [Krotzsch *et al.*, 2006] representing the static but very rich implicit knowledge that can be retrieved. The second example beginning with (7) evokes a dynamically composed web service lookup. It is important to note that the query representation is the same for all the access methods to the Semantic Web (cf. section 5.1) and is defined by foundational and domain-specific ontologies. In case that the GPS co-cordinates were accessible from the mobile device, the clarification question would have been omitted.

3 Architecture approach

A flexible dialog system platform is required in order to allow for true multi-session operation with multiple concurrent users of the server-side system as well as to support audio transfer and other data connections between the mobile device and a remote dialog server. This types of systems have been developed, like the Galaxy Communicator [Cheyer and Martin, 2001] (cf. also [Seneff *et al.*, 1999; Thorisson *et al.*, 2004; Herzog *et al.*, 2004; Bontcheva *et al.*, 2004]), and commercial platforms from major vendors like VoiceGenie, Kirusa, IBM, and Microsoft use X+V1, HTML+SALT2, or derivatives for speech-based interaction on mobile devices. For our purposes these platforms are too limited. To implement new interaction metaphors and to use Semantic Web based data structures for both dialog system internal and external communication, we developed a platform designed for Semantic Web data structures for NLP components and backend knowledge server communication. The basic architecture is shown in figure 1.

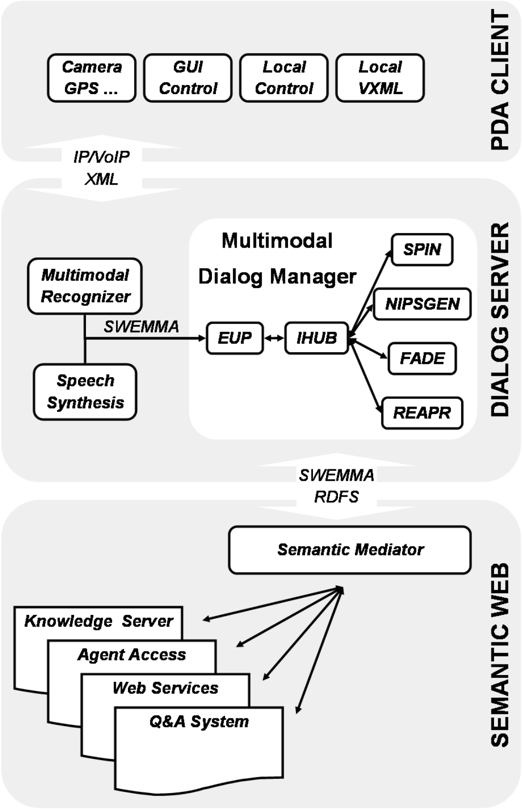


Figure 1: SMARTWEB handheld architecture.

It consists of three basic processing blocks: the PDA client, the dialog server which comprises the dialog manager, and the Semantic Web access system.

On the PDA client, a local Java-based control unit takes care of all I/O, and is connected to the GUI-controller. The local VoiceXML-based dialog system resists on the PDA for interaction during link downtimes.

The dialog server system platform instantiates one dialog server for each call and connects the multimodal recognizer for speech and gesture recognition. The dialog system instantiates and sends the requests to the *Semantic Mediator*, which provides the umbrella for all different access methods to the Semantic Web we use. It consists of an open domain QA system, a Semantic Web service composer, Semantic Web pages (wrapped by semantic agents), and a knowledge server.

The dialog system consist of different, self-contained processing components. To integrate them we developed a Java-based hub-and-spoke architecture [Reithinger and Sonntag, 2005]. The most important processing modules in the dialog system connected in the IHUB are: a speech interpretation component (SPIN), a modality fusion and discourse component (FADE), a system reaction and presentation component (REAPR), and a natural language generation module (NIPSGEN), all discussed in section 5. An EMMA Unpacker/Packer (EUP) component provides the communication with the dialogue server and Semantic Web subsystem external to the multimodal dialog manager and communicates with the other modules of the dialog server, the multimodal recognizer, and the speech synthesis system.

Processing a user turn, the normal data flows through $SPIN \rightarrow FADE \rightarrow REAPR \rightarrow SemanticMediator \rightarrow REAPR \rightarrow NIPSGEN$. However, the data flow is often more complicated when, for example, misinterpretations and clarifications are involved.

4 Ontology representation and web services

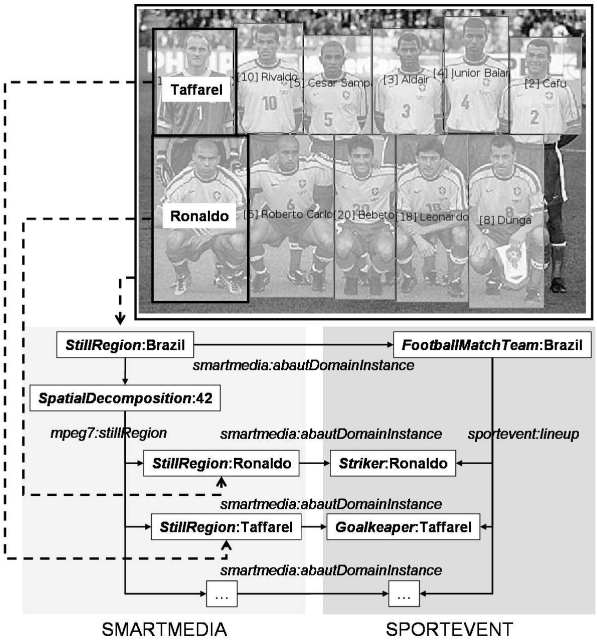


Figure 2: A SMARTMEDIA instance representing the decomposition of the Brazil 1998 world cup football team image.

The ontological infrastructure of SMARTWEB, the SWIntO (SMARTWEB **Int**egrated **O**ntology), is based on an upper model ontology realized by merging well chosen concepts from two established foundational ontologies, DOLCE [Gangemi *et al.*, 2002] and SUMO [Niles and Pease, 2001], in a unique one: the SMARTWEB foundational ontology SMARTSUMO [Cimiano *et al.*, 2004]. Domain specific knowledge (sportevent, navigation) is defined in dedicated ontologies modeled as sub-ontologies of the SMARTSUMO. The SWIntO integrates question answering specific knowledge of a discourse ontology (DISCONTO) and representation of multimodal information of a media ontology (SMARTMEDIA). The data exchange is RDF-based.

We realized a discourse ontology (DISCONTO) with particular attention to the modeling of discourse interactions in QA scenarios. The DISCONTO provides concepts for dialogical interaction with the user as well as more technical request-response concepts for data exchange with the Semantic Web subsystem including answer status which is important in interactive systems. In particular DISCONTO comprises concepts for multimodal dialog management, a dialog act taxonomy, lexical rules for syntactic-semantic mapping, HCI concepts (e.g. pattern language for interaction design [Sonntag, 2005]), and concepts for questions, question focus, semantic answer types [Hovy *et al.*, March 2001], and multimodal results [Sonntag and Romanelli, 2006].

Information exchange between the components of the server-side dialog system is based on the W3C EMMA standard that is used to realize containers for the ontological instances representing, e.g., multimodal input interpretations. SWEMMA is our extension to the EMMA standard which introduces additional *Result* structures in order to represent components output. On the ontological level we modeled an RDF/S-representation of EMMA/SWEMMA.

The SMARTMEDIA is an MPEG7-based media ontology and an extension to [Hunter, 2001; Benitez *et al.*, 2002] that we use to represent output result, offering functionality for multimedia decomposition in space, time and frequency (mpeg7:SegmentDecomposition), file format and coding parameters (mpeg7:MediaFormat), and a link to the Upper Model Ontology (smartmedia:aboutDomainInstance). In order to close the semantic gap between the different levels of media representations, the *smartmedia:aboutDomainInstance* property has been located in the top level class *smartmedia:Segment*. The link to the upper model ontology is inherited to all segments of a media instance decomposition to guarantee deep semantic representations for the *smartmedia* instances referencing the specific media object and for making up segment decompositions.

Figure 2 shows an example of this procedure applied to an image of the Brazilian football team in the final match of the World Cup 1998, as introduced in the interaction example. In the example an instance of the class *mpeg7:StillRegion*, representing the complete image, is decomposed into different *mpeg7:StillRegion* instances representing the segments of the image which show individual players.

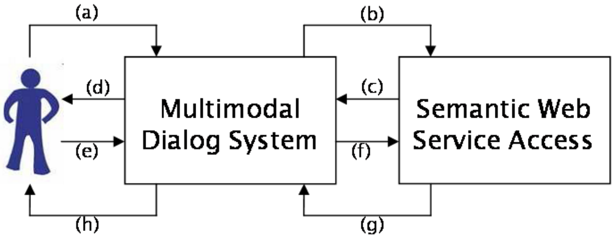
The *mpeg7:StillRegion* instance representing the entire picture is then linked to a *sportevent:MatchTeam* instance, and each segment of the picture is linked to a *sportevent:FieldFootballPlayer* instance or sub-instance. These representations offer a framework for gesture and speech fusion when users interact with Semantic Web results such as MPEG7-annotated images, maps with points-of interest, or other interactive graphical media obtained from the ontological knowledge base or multimedia web services.

4.1 Multimodal access to web services

To connect to web services we developed a semantic representation formalism based on OWL-S and a service composition component able to interpret an ontological user query. We extended the OWL-S ontologies to flexibly compose and invoke web services on the fly, gaining sophisticated representation of information gathering services fundamental to SMARTWEB.

Sophisticated data representation is the key for developing a composition engine that exploits the semantics of web service annotation and query representation. The composition engine follows a plan-based approach as explained, e.g., in [Ghallab *et al.*, 2004]. It infers the initial and goal state from the semantic representation of the user query, whereas the set of semantic web services is considered as planning operators. The output gained from automatic web service invocation is represented in terms of instances of the SMARTWEB domain ontologies and enriched by additional media instances, if available. Media objects are represented in terms of the SMARTMEDIA ontology (see above) and are annotated automatically during service execution. This enables the dialog manager for multimodal interaction with web service results.

A key feature of the service composition engine is to detect underspecified user queries, i.e., the lack of required web service input parameters. In these cases the composition engine is able to formulate a clarification request as specified within the discourse ontology (DISCONTO). This points out the missing pieces of information to be forwarded to the dialog manager. Then the composition engine expects a clarification reponse enabling it to replan on the refined ontological user query.



(a) User query: What can I do in my spare time on Saturday?
(b) Ontological user query is sent to web services.
(c) Clarification request (asking for a city) is sent back.
(d) Verbalized clarification request: Where?
(e) User clarification response: In Berlin.
(f) Completed ontological query is sent to web services.
(g) Ontological result of service execution is sent to dialog.
(h) Generated results are multimodally presented to the user.

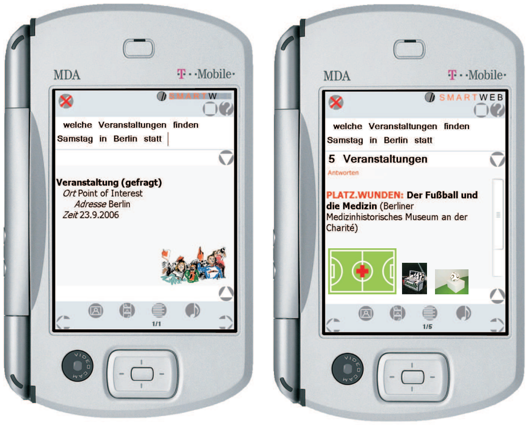
Figure 3: Data flow for the processing of a clarification request as in the example (7-10) ”What can I do in my spare time on Saturday?”.

According to the interaction example (7-10) the composition engine searches for a web service demanding for activity event types and gets its description. Normally, the context module incorporated in the dialog manager would complete the query with the venue obtained from a GPS receiver attached to the handheld device. In case of no GPS signal, for instance indoors, the composition engine asks for the missing parameter (cf. figure 3), which makes the composition engine more robust and thus more suitable for interactive scenarios.

In the interaction example (7-10) the composition planner considers the *T-Info EventService* appropriate for answering the query. This service requires both date and location for looking up events. While the date is already mentioned in the initial user query, the location is being asked from the user by clarification request. After the location information (dialogue step (9) in the example:*In Berlin*) is obtained from the user, the composition engine invokes in turn two T-Info (DTAG) web services[8] offered by Deutsche Telekom AG (see also [Ankolekar *et al.*, 2006]): first the *T-Info EventService* as already mentioned above, and then the *T-Info MapService* for calculating an interactive map showing the venue as point-of-interest. Text-based event details, additional image material, and the location map are semantically represented (the map in MPEG7) and returned to the dialog engine.

5 Semantic parsing and discourse processing

Semantic parsing and other discourse processing steps are reflected on the interaction device as advanced user perceptual feedback functionality. The following screenshot illustrates the two most important processing steps for system-user interaction, the feedback on the natural language understanding step and the presentation of multimodal results. The semantic parser produces a semantic query (illustrated on the left in figure 4), which is presented to the user in nested attribute-value form. The web service results (illustrated on the right in figure 4) for the interaction example (7-10) are presented in a multimodal way, combining text, image, and speech: *5 Veranstaltungen* (five events).



welche Veranstaltungen finden Samstag in Berlin statt

Veranstaltung (gefragt)
Ort Point of Interest
Adresse Berlin
Zeit 23.9.2006

welche Veranstaltungen finden Samstag in Berlin statt

5 Veranstaltungen

PLATZ.WUNDEN: Der Fußball und die Medizin (Berliner Medizinhistorisches Museum an der Charité)

Figure 4: Semantic query (illustrated on the left) and web service results (illustrated on the right).

[8] http://services.t-info.de/soap.index.jsp

5.1 Language understanding with SPIN and text generation with NIPSGEN

The parsing module is based on the semantic parser SPIN [Engel, 2005]. A syntactic analysis of the input utterance is not performed, but the ontology instances are created directly from word level. The typical advantages of a semantic parsing approach are that processing is faster and more robust against speech recognition errors and disfluencies produced by the user and the rules are easier to write and maintain. Also, multilingual dialog systems are easier to realize as a syntactic analysis is not required for each supported language. A disadvantage is that the complexity of the possible utterances is somewhat limited, but this is acceptable for most dialog systems.

One outstanding feature of the parser is the possibility for order-independent matching, i.e., the order of elements in the input stream is ignored if order-independent matching is active. This simplifies the processing of free-word order languages like German and increases the robustness. Order-independent matching can have an huge impact on performance as parsing in general becomes an NP-complete task [Huynh, 1983]. To ensure fast processing notwithstanding, several off-line optimizations, like rule ordering, have been implemented which increase the performance for rule sets that are typical for dialog systems. The average processing time is about 50ms per utterance, which ensures direct feedback to user inputs.

The knowledge base of the parser consists of 544 rules and 2250 lexicon entries currently. To give an impression how the rules look like, four rules are provided as examples to process the utterance *When was Brazil world champion*. The first one transforms the word *Brazil* to the ontology instance `Country`:

```
Brazil → Country(name:BRAZIL)
```

The second one transforms countries to teams as each country can stand for a team in our domain:

```
$C=Country() → Team(origin:$C)
```

The third one processes *when* generating an instance of the type `TimePoint` which is marked as questioned:

```
when →
TimePoint(variable:QEVariable(focus:text))
```

The fourth rule processes the verbal phrase <*TimePoint*> *was* <*Team*> *world champion*

```
$TP=TimePoint() was $TM=Team() world
   champion →
QEPattern(patternArg:Tournament(
              winner:$TM, happensAt:$TP))
```

The text generation module uses the same SPIN parser that is used in the language understanding module together with a TAG grammar which is modelled similar to the XTAG grammar[9]. The input of the generation module are instances of SWIntO representing the search results. Then these results are verbalized in different ways, e.g., as heading, as row of a table or as text which is synthesized. A processing option indicates the current purpose.

The input is transformed to an utterance in four steps:

1. An intermediate representation is built up on a phrase level. The required rules are domain dependent.
2. A set of domain independent rules transforms the intermediate representation to a derivation tree for the TAG-grammar.
3. The actual syntax tree is constructed using the derivation tree. After the tree has been built up, the features of the tree nodes are unified.
4. The correct inflections for all lexical leafs are looked up in the lexicon. Traversing the lexical leafs from left to right produces the result text.

In the SMARTWEB system currently 179 domain dependent generation rules and 38 domain independent rules are used.

5.2 Multimodal discourse processing with FADE

An important aspect of SMARTWEB is its context-aware processing strategy. All recognized user actions are processed with respect to their situational and discourse context. A user is thus not required to pose separate and unconnected questions. In fact, she might refer directly to the situation, e.g., *“How do I get to Berlin from here?”*, where *here* is resolved to GPS information, or to previous contributions (as in the elliptical expression *“And in 2002?”* in the context of a previously posed question *“Who won the Fifa World Cup in 1990?”*). The interpretation of user contributions with respect to their discourse context is performed by a component called *Fusion and Discourse Engine*—FADE [Pfleger, 2005][10]. The task of FADE is to integrate the verbal and nonverbal user contributions into a coherent multimodal representation to be enriched by contextual information, e.g., resolution of referring and elliptical expressions.

The basic architecture of FADE consists of two interweaved processing layers: (1) a production rule system—PATE—that is responsible for the reactive interpretation of perceived monomodal events, and (2) a discourse modeler—DiM—that is responsible for maintaining a coherent representation of the ongoing discourse and for the resolution of referring and elliptical expressions.

In the following two subsections we will briefly discuss some context-related phenomena that can be resolved by FADE.

Resolution of referring expressions

A key feature of the SMARTWEB system is that the system is capable of dealing with a broad range of referring expressions as they occur in natural dialogs. This means the user can employ deictic references that are accompanied by a pointing gesture (such as in *“How often did this team [pointing gesture] win the World Cup?”*) but also—if the context provides enough disambiguating information—without any accompanying gestures (e.g., if the previous question is uttered in the context of a previous request like *“When was Germany World Cup champion for the last time?”*).

[9] http://www.cis.upenn.edu/∼xtag/

[10] The situational context is maintained by another component called *SitCom* that is not discussed in this paper.

Moreover, the user is also able to utter time deictic references as in *"What's the weather going to be like tomorrow?"* or *"What's the weather going to be like next Saturday?"*.

Another feature supported by FADE is the resolution of *cross modal* spatial references, i.e., a spoken reference to visually displayed information. The user can refer, for example, to an object that is currently displayed on the screen. If a picture of the German football team is displayed, the system is able to resolve references like *"this team"* even when the team has not yet been mentioned verbally. MPEG7-annotated images (see section 4) even permit spatial references to objects displayed within pictures, e.g., as in *"What's the name of the guy to the right of Ronaldo?"* or *"What's the name of the third player in the top row?"*.

Resolution of elliptical expression

Humans tend to keep their contributions as short and efficient as possible. This is in particular the case for follow-up questions or answers to questions. Here, people often make use of elliptical expressions, e.g., when they ask a follow-up question *"And the day after tomorrow?"* in the context of a previous question *"What's the weather going to be like tomorrow?"*. But even for normal question-answer pairs people tend to omit everything that has already been conveyed by the question (User: *"Berlin"* in the context of a clarification question of the system like *"Where do you want to start?"*; see section 4.1).

Elliptical expressions are processed in SMARTWEB as follows: First, SPIN generates an ontological query that contains a semantic representation of the elliptical expression, e.g., in case of the aforementioned example "Berlin". This analysis would only comprise an ontological instance representing the city Berlin. FADE in turn, then tries to integrate the elliptical expression with the previous system utterance, if this was a question. Otherwise it tries to integrate the elliptical expression with the previous user request. If the resolution succeeded, the resulting interpretation either describes the answer to the previous clarification question, or it describes a new question.

5.3 Reaction and presentation planning for the Semantic Web

Integral part of dialog management is the reaction and presentation module (REAPR). It manages the dialogical interaction for the supported dialog phenomena such as flexible turn-taking, incremental processing, and multimodal fusion of system output. REAPR is based on a finite-state-automaton and information space (IS). Our new approach differs from other IS approaches (e.g. [Matheson *et al.*, 2000]) by generating IS features from the ontological instances generated during dialog processing [Sonntag, 2006]. [11]

Since the dialog ontology is a model for multimodal interaction, multimodal MPEG7 result representations, multimodal result presentations, dialog state, and (agent) communication with the backend knowlege servers, large information spaces can be extracted from the ontological instances describing the system and user turns in terms of special dialog acts - to ensure accurate dialog management capabilities. REAPR decides, for example, if a semantic query is accepted for transfer to the Semantic Mediator. The IS approach to dialog modeling comprises, apart from dialog moves and update strategies, a description of informational components (e.g. common ground) and their formal representations. Since in REAPR the formal dialog specification consists of ontological structures as Semantic Web data structures, a formal well-defined complement to previous formal logic-based operators and Discourse Representation Structures (DRS) is provided. However, the ontological structures resemble typed feature structures (TFS) [Carpenter, 1992] we use for illustration further down. During interaction, many message transfer processes take place, mainly for query recognition and query processing, all of which are based on Semantic Web ontological structures, and REAPR is involved in many of them. Here we give an example of ontological representations of user pointing gestures (dialog step (5) in the interaction example) which are obtained from the PDA and transformed into ontology-structures to be used by the input fusion module. The following figure shows the ontological representation of a pointing gesture as TFS.

$$\begin{bmatrix} \text{\textsc{PointingGesture}} & \\ \text{timePoint:} & 1151576316802 \\ \text{coordinate:} & \begin{bmatrix} \text{\textsc{CartesianCoordinate}} & \\ \text{xAxis:} & 195 \\ \text{yAxis:} & 55 \end{bmatrix} \\ \text{objectReference:} & \begin{bmatrix} \text{\textsc{FieldMatchFootballPlayer}} & \\ \text{label:} & \mathit{Aldair} \\ \text{number:} & 3 \\ \text{inMatchTeam:} & \begin{bmatrix} \text{\textsc{MatchTeam}} \\ \ldots \end{bmatrix} \\ \text{hasUpperRole:} & \begin{bmatrix} \text{\textsc{UpperRole}} \\ \ldots \end{bmatrix} \end{bmatrix} \end{bmatrix}$$

It is important to mention that dialog reaction behaviour within SMARTWEB is governed by the general QA scenario, which means that almost all dialog and system moves relate to questions, follow-up questions, clarifications, or answers. As these dialog moves can be regarded as adjacency pairs, the dialog behaves according to some finite state grammar for QA, which makes up the automaton part (FSA) in REAPR. The finite state approach enhances robustness and portability and allows to demonstrate dialog management capabilities even before more complex IS states are available to be integrated into the reaction and presentation decision process. The dialog component integration process is described in the next section.

5.4 Dialog component integration

In this section we will focus on issues of interest pertaining to the system integration. In the first instance dialog component integration is an integration on a conceptual level. All dialog manager components communicate via ontology instances. This assumes the representation of all relevant concepts in the foundational and domain ontologies – which is

[11] The IS state is traditionally divided into global and local variables which make up the knowledge state at a given time point. Ontological structures that change over time vastly enhance the representation capabilities of dialog management structures, or other structures like queries from which relevant features can also be extracted.

hard to provide at the beginning of the integration. In our experience, using ontologies in information gathering dialog systems for knowledge retrieval from ontologies and web services in combination with advanced dialogical interaction is an iterative ontology engineering process, which requires very disciplined ontology updates, since changes and extensions must be incorporated into all relevant components. The additional modeling effort pays off when regarding the strength of this Semantic Web technology for larger scale projects.

We first built up an initial discourse ontology for request-response concepts for data exchange with the Semantic Web sub-system. In addition, an ontological dialog act taxonomy has been specified, to be used by the semantic parsing and discourse processing modules. A great challenge is the mapping between semantic queries and the ontology instances in the knowledge base. In our system, the discourse (understanding) specific concepts have been linked up with the foundational ontology and, e.g., the sportevent ontology, and the semantic parser only builds up interpretations with SWIntO concepts. Although this limits the space of possible interpretations according to the expressivity of the foundational and domain ontologies, the robustness of the system is increased. We completely circumvent the problem of concept and relation similarity matching between conventional syntactic/semantic parsers and backend retrieval systems.

Regarding web services we transform the output from the web services, in particular maps with points of interest, into instances of the SMARTWEB domain ontologies for the same reasons of semantic integration. As already noted, ontological representations offer a framework for gesture and speech fusion when users interact with Semantic Web results such as MPEG7-annotated images and maps. Challenges in multimodal fusion and reaction planning can be addressed by using more structured representations of the displayed content, especially for pointing gestures, which contain references to player instances after integration. We extended this to pointing gesture representations on multiple levels in the course of development, to include representations of the interaction context, the modalities and display patterns used, and so on.

The primary aim is to generate structured input spaces for more context-relevant reaction planning to ensure naturalness in system-user interactions to a large degree. Currently, we experiment with the MDA's camera input indicating whether the user is looking at the device, to combine it with other indicators to a measure of user focus. The challenge of integrating and fusing multiple input modalities can be reduced by ontological representations, which exist at well-defined time-points, and are also accessible to other components such as the semantic parser, or the reaction and presentation module.

6 Conclusions

We presented a mobile system for multimodal interaction with an ontological knowledge base and web services in a dialog-based QA scenario. The interface and content representations are based on W3C standards such as EMMA and RDF. The world knowledge shared in all knowledge-intensive components is based on the existing ontologies SUMO and DOLCE, for which we added additional concepts for QA and multimodal interaction in a discourse ontology branch.

We presented the development of the second demonstrator of the SMARTWEB system which was successfully demonstrated in the context of the Football World Cup 2006 in Germany. The SWIntO ontology now comprises 2308 concept classes, 1036 slots and 90522 instances.[12] For inference and retrieval the ontology constitutes 78385 data instances after deductions.[13] The answer times are in a 1 to 15 seconds time frame for about 90% of all questions. In general, questions without images and videos as answers can be processed much faster. The web service composer addresses 25 external services from traveling (navigation, train connections, maps, hotels), event information, points of interest (POIs), product information (books, movies), webcam images, and weather information.

The SMARTWEB architecture supports advanced QA functionalities such as flexible control flow to allow for clarification questions of web services when needed, long- and short-term memory provided by distributed dialog management in the fusion and discourse module and in the reaction and presentation module, as well as semantic interpretations provided by the speech interpretation module. This can be naturally combined with dialog system strategies for error recoveries, clarifications with the user, and multimodal interactions. Support for inferential, i.e., deductive reasoning, complements the requirements for advanced QA in terms of information- and knowledge retrieval. Integrated approaches as presented here rely on ontological structures and deeper understanding of questions, not at least to provide a foundation for result provenance explanation and justification. Our future plans on the final six month agenda include dialog management adaptations via machine learning and collaborative filtering of redundant results in our multi-user enviroment, and incremental presentation of results.

7 Acknowledgments

The research presented here is sponsored by the German Ministry of Research and Technology (BMBF) under grant 01IMD01A (SmartWeb). We thank our student assistants and the project partners. The responsibility for this papers lies with the authors.

References

[Allen *et al.*, 2000] James Allen, Donna Byron, Myroslava Dzikovska, George Ferguson, Lucian Galescu, and Amanda Stent. An Architecture for a Generic Dialogue Shell. *Natural Language Engineering*, 6(3):1–16, 2000.

[Ankolekar *et al.*, 2006] Anupriya Ankolekar, Pascal Hitzler, Holger Lewen, Daniel Oberle, and Rudi Studer. Integrating semantic web services for mobile access. In *Proceedings of 3rd European Semantic Web Conference (ESWC 2006)*, 2006.

[12] The SWIntO can be downloaded at the SMARTWEB homepage for research purposes.

[13] The original data instance set was 175293 instances, but evoked processing times up to two minutes for single questions by what interactivity was no longer guaranteed.

[Benitez *et al.*, 2002] Ana B. Benitez, Hawley Rising, Corinne Jorgensen, Ricardo Leonardi, Alesandro Bugatti, Koiti Hasida, Rajiv Mehrotra, A. Murat Tekalp, Ahmet Ekin, and Toby Walker. Semantics of Multimedia in MPEG-7. In *IEEE International Conference on Image Processing (ICIP)*, 2002.

[Bontcheva *et al.*, 2004] Kalina Bontcheva, Valentin Tablan, Diana Maynard, and Hamish Cunningham. Evolving GATE to Meet New Challenges in Language Engineering. *Natural Language Engineering*, 10, 2004. Special issue on Software Architecture for Language Engineering.

[Carpenter, 1992] B. Carpenter. The logic of typed feature structures, 1992.

[Cheyer and Martin, 2001] Adam J. Cheyer and David L. Martin. The Open Agent Architecture. *Autonomous Agents and Multi-Agent Systems*, 4(1–2):143–148, 2001.

[Cimiano *et al.*, 2004] Philipp Cimiano, Andreas Eberhart, Pascal Hitzler, Daniel Oberle, Steffen Staab, and Rudi Studer. The smartweb foundational ontology. Technical report, (AIFB), University of Karlsruhe, Karlsruhe, Germany, 2004. SmartWeb Project.

[Engel, 2005] Ralf Engel. Robust and efficient semantic parsing of free word order languages in spoken dialogue systems. In *Proceedings of 9th Conference on Speech Communication and technology*, Lisboa, 2005.

[Fensel *et al.*, 2003] Dieter Fensel, James A. Hendler, Henry Lieberman, and Wolfgang Wahlster, editors. *Spinning the Semantic Web: Bringing the World Wide Web to Its Full Potential*. MIT Press, 2003.

[Gangemi *et al.*, 2002] Aldo Gangemi, Nicola Guarino, Claudio Masolo, Alessandro Oltramari, and Luc Schcneider. Sweetening Ontologies with DOLCE. In *In 13th International Conference on Knowledge Engineering and Knowledge Management (EKAW02)*, volume 2473 of Lecture Notes in Computer Science, page 166 ff, Sigünza, Spain, Oct. 1–4 2002.

[Ghallab *et al.*, 2004] Malik Ghallab, Dana Nau, and Paolo Traverso. *Automated planning*. Elsevier Kaufmann, Amsterdam, 2004.

[Herzog *et al.*, 2004] Gerd Herzog, Alassane Ndiaye, Stefan Merten, Heinz Kirchmann, Tilman Becker, and Peter Poller. Large-scale Software Integration for Spoken Language and Multimodal Dialog Systems. *Natural Language Engineering*, 10, 2004. Special issue on Software Architecture for Language Engineering.

[Hovy *et al.*, March 2001] Eduard Hovy, Laurie Gerber, Ulf Hermjakob, Chin-Yew Lin, and Deepak Ravichandran. Towards semantic-based answer pinpointing. In *Proceedings of Human Language Technologies Conference, San Diego CA*, pages 339–345, March 2001.

[Hunter, 2001] Jane Hunter. Adding Multimedia to the Semantic Web - Building an MPEG-7 Ontology. In *Proceedings of the International Semantic Web Working Symposium (SWWS)*, 2001.

[Huynh, 1983] Dung T. Huynh. Communicative grammars: The complexity of uniform word problems. *Information and Control*, 57(1):21–39, 1983.

[Krotzsch *et al.*, 2006] Markus Krotzsch, Pascal Hitzler, Denny Vrandecic, and Michael Sintek. How to reason with OWL in a logic programming system. In *Proceedings of RuleML'06*, 2006.

[Matheson *et al.*, 2000] C. Matheson, M. Poesio, and D. Traum. Modelling grounding and discourse obligations using update rules. In *Proceedings of NAACL 2000*, May 2000.

[Niles and Pease, 2001] Ian Niles and Adam Pease. Towards a Standard Upper Ontology. In Chris Welty and Barry Smith, editors, *Proceedings of the 2nd International Conference on Formal Ontology in Information Systems (FOIS-2001)*, Ogunquit, Maine, October 17–19 2001.

[Oviatt, 1999] Sharon Oviatt. Ten myths of multimodal interaction. *Communications of the ACM*, 42(11):74–81, 1999.

[Pfleger, 2005] Norbert Pfleger. Fade - an integrated approach to multimodal fusion and discourse processing. In *Proceedings of the Dotoral Spotlight at ICMI 2005*, Trento, Italy, 2005.

[Reithinger and Sonntag, 2005] Norbert Reithinger and Daniel Sonntag. An integration framework for a mobile multimodal dialogue system accessing the semantic web. In *Proc. of Interspeech'05*, Lisbon, Portugal, 2005.

[Reithinger *et al.*, 2005a] Norbert Reithinger, Simon Bergweiler, Ralf Engel, Gerd Herzog, Norbert Pfeger, Massimo Romanelli, and Daniel Sonntag. A Look Under the Hood Design and Development of the First SmartWeb System Demonstrator. In *Proceedings of 7th International Conference on Multimodal Interfaces (ICMI 2005)*, Trento, Italy, October 04-06 2005.

[Reithinger *et al.*, 2005b] Norbert Reithinger, Dirk Fedeler, Ashwani Kumar, Christoph Lauer, Elsa Pecourt, and Laurent Romary. MIAMM - A Multimodal Dialogue System Using Haptics. In Jan van Kuppevelt, Laila Dybkjaer, and Niels Ole Bernsen, editors, *Advances in Natural Multimodal Dialogue Systems*. Springer, 2005.

[Seneff *et al.*, 1999] Stephanie Seneff, Raymond Lau, and Joseph Polifroni. Organization, Communication, and Control in the Galaxy-II Conversational System. In *Proc. of Eurospeech'99*, pages 1271–1274, Budapest, Hungary, 1999.

[Sonntag and Romanelli, 2006] Daniel Sonntag and Massimo Romanelli. A multimodal result ontology for integrated semantic web dialogue applications. In *Proceedings of the 5th Conference on Language Resources and Evaluation (LREC 2006)*, Genova, Italy, May 24–26 2006.

[Sonntag, 2005] Daniel Sonntag. Towards interaction ontologies for mobile devices accessing the semantic web - pattern languages for open domain information providing multimodal dialogue systems. In *Proceedings of the workshop on Artificial Intelligence in Mobile Systems (AIMS). 2005 at MobileHCI*, Salzburg, 2005.

[Sonntag, 2006] Daniel Sonntag. Towards combining finite-state, ontologies, and data driven approaches to dialogue management for multimodal question answering. In *Proceedings of the 5th Slovenian First International Language Technology Conference (IS-LTC 2006)*, 2006.

[Thorisson *et al.*, 2004] Kristinn R. Thorisson, Christopher Pennock, Thos List, and John DiPirro. Artificial intelligence in computer graphics: A constructionist approach. *Computer Graphics*, pages 26–30, February 2004.

[Wahlster, 2000] Wolfgang Wahlster, editor. *VERBMOBIL: Foundations of Speech-to-Speech Translation*. Springer, 2000.

[Wahlster, 2003] Wolfgang Wahlster. SmartKom: Symmetric Multimodality in an Adaptive and Reusable Dialogue Shell. In R. Krahl and D. Günther, editors, *Proc. of the Human Computer Interaction Status Conference 2003*, pages 47–62, Berlin, Germany, 2003. DLR.

[Wahlster, 2004] Wolfgang Wahlster. SmartWeb: Mobile Applications of the Semantic Web. In Peter Dadam and Manfred Reichert, editors, *GI Jahrestagung 2004*, pages 26–27. Springer, 2004.