

An extensible platform for seamless integration and management of applications for emotion sensing and interpretation

J. Alfredo Sánchez^{a,*}, Ximena Cortés^a, Oleg Starostenko^a, Ofelia Cervantes^a and Wanggen Wan^b

^a *Laboratory of Interactive and Cooperative Technologies, Universidad de las Américas Puebla, Santa Catarina Mártir S/N, Cholula, Puebla 72820, Mexico*

^b *Institute of Smart City, University of Shanghai, E605, No. 99 Shangda Road, Shanghai, China*

Abstract. The role of affect has become increasingly important in ambient intelligence applications. Developers require means for using multiple tools that support emotion detection and interpretation, which can be used jointly to provide meaningful system responses for the user in varied situations. We introduce *Vikara*, an extensible software platform that provides developers with uniform interfaces and services so their applications can access the results from existing (or newly implemented) tools for emotion sensing and interpretation. *Vikara* also provides components that make it possible for platform managers to visually monitor affective states as they are sensed by the various available tools. We have experimented with our platform via the development of applications for emotion detection by using (1) the well-known Facial Action Coding System and a Kinect sensor; and (2) a self-reporting interface for Android-based mobile devices. We report on the initial results of using our platform and discuss ambient intelligence scenarios in which we see potential applications, including user experience and usability evaluation, marketing, distance education and inter-personal communication.

Keywords: Affective computing applications, emotion detection, emotion interpretation, integration platform

1. Introduction

Ambient intelligence (AmI) aims to construct environments that respond to the presence and needs of users in meaningful ways. In order to accomplish this, AmI applications should take advantage of all available sources of information regarding the user's context, needs and expectations. Emotions are fundamental components of the users' context; however, incorporating them into AmI applications is not straightforward.

Detecting an emotion refers to the process of discovering user actions or reactions, or cues, that are associated with affect, whereas interpretation refers to the process of determining a specific emotion and its in-

tensity as a result of analyzing those cues. For brevity, in this paper we refer to detection and interpretation of emotions under the umbrella term of emotion analysis. There has been significant progress in the development of techniques for emotion analysis in varied contexts. Facial cues, body or brain sensors, speech variations, posture reactions, among others, have been used as the basis for relatively successful techniques for emotion analysis in real time. Other techniques rely on the users' self-report of their feelings regarding specific events or situations. Each of these approaches exhibits advantages and disadvantages, depending on the contexts or settings in which they are applied. There are also tools that help developers in the process of incorporating specific techniques for emotion analysis into their applications. However, programming libraries and development kits typically focus on the use of one specific approach, which relies on

* Corresponding author. E-mail: alfredo.sanchez@udlap.mx.

its own devices, representation models and data collection methods. This makes it difficult for programmers to combine emotion analysis techniques, particularly in the context of ambient intelligence applications. In AmI, the integration of diverse approaches can result in significant improvements in terms of accuracy or response time that may be critical for taking appropriate actions in a natural manner.

In this paper, we introduce *Vikara* (Sanskrit for *affection*), an extensible software platform that provides developers with uniform interfaces and services so their applications can access the results from existing (or newly implemented) tools for emotion analysis. *Vikara* also provides components that make it possible for platform managers to visually monitor affective states as they are sensed by the various available tools. Given that our platform is addressed to software developers who need to integrate affect-aware components, we demonstrate the validity of our approach by developing two applications that rely on differing techniques for emotion analysis: a Kinect-based emotion interpreter that relies on the well known Facial Action Coding System (FACS), and a self-reporting interface for Android-based mobile devices. We have successfully integrated these applications into our *Vikara* platform and we demonstrate how other developments can benefit from the availability of our open framework.

We report on the initial results of using our platform and discuss ambient intelligence scenarios in which we see potential applications, including user experience and usability evaluation, marketing, distance education and inter-personal communication.

2. Detecting and interpreting emotions

In this section, we provide some background for the main concepts related to our work as well as the challenges in the development of affect-aware applications. In Section 2.1, we start by clarifying the notions of emotion, mood and sentiment as used in this paper. We identify the set of basic emotions that we have been considering in our work, and we discuss their relevance for ambient intelligence. Then, in Section 2.2, we provide an overview of related work in the area of emotion analysis. In particular, we refer to work that has explored the analysis of emotions by relying on diverse modalities, as well as the challenges for their integration. These challenges have motivated our platform for integration of emotion analysis techniques, which is discussed in Section 3.

2.1. The objects of our affection

Terms related to affect and affective computing have been discussed extensively in the literature. Ambiguity may arise due to common use of related terms, such as emotions, moods and sentiments, in colloquial language. In this paper we adopt a distinction between these terms that is based on object-directedness, as discussed by Brave and Nass [2].

Thus, *emotion* refers to an instantaneous physiological, behavioral and cognitive reaction towards an event considered relevant to an individual. That event is the object towards which an emotion, such as fear or joy, is experienced. *Mood* is a more diffuse, longer lasting affective state that is not directed to a particular object. Depression is an example of mood that may have an impact on an individual's cognitive strategies over a longer term. Finally, a sentiment can be considered a state that results from attributes assigned to objects, which are grouped into categories or stereotypes. For example, a person may *hate* action movies, which may result from having watched movies of this kind that were too fast-paced or exceeded a tolerable level of noise, which are attributes assigned by the individual to all movies in the category.

Although there is a wide range of identifiable emotions, for the sake of simplicity and for demonstrating our integrative approach, we have adopted the view that suggests there is only a small set of innate *basic emotions* that transcend cultures [9]: fear, anger, disgust, surprise, happiness and sadness. Certainly, people do experience variations of these basic or *pure emotions*, but theorists posit that those variations are socially learned or are combinations of pure emotions.

Not surprisingly, emotions, moods and sentiments are closely inter-related, since recurring emotions may lead to certain moods, and these in turn may eventually lead to sentiments. In the case of user interfaces, for example, moods and sentiments may act as negative or positive filters for the user's perception and acceptance. In ambient intelligence, moods and sentiments can play an important role in determining appropriate system responses. However, although detecting moods and sentiments in automated ways may prove difficult, users may be able to state more directly whether they are in a certain mood or have a specific propensity or preference derived from sentiments. Contrastingly, people are generally less aware of their emotions at the time they are being experienced, hence the importance of assisting emotion analysis by using au-

tomated means. The instantaneous nature of emotions poses challenges for detection and interpretation that, if overcome, can have an important impact on highly responsive and adaptive ambient intelligence applications.

In addition to emotion detection, affective computing also covers systems that are capable of expressing what humans perceive as emotions (emotion synthesis), as well as environments that address effective mediation of affect between humans when technology is the main communication substrate [17]. In this paper, we focus mainly on the integration and management of emotion analysis applications. We posit that the coordinated use of multiple emotion analysis techniques can improve the accuracy and performance of emotion-aware applications, thus leading to potentially more adaptive responses from intelligent environments.

2.2. Support for developing emotion analysis software

Emotion analysis has been approached from very diverse perspectives. Calvo and D’Mello [3] conducted a survey in which they covered traditional techniques for emotion detection, which include physiological sensors, face analysis, and voice processing, as well as more recent methods, such as the use of affective words in text, and body language and posture. Their survey also examines thoroughly the theoretical models that underlie existing emotion detection techniques. Given the importance of multimodality for our integration approach, in what follows we discuss the main issues that have been associated to the use of varied sources for emotion analysis, as well as some of the salient works that have relied on alternative user cues for emotion detection.

2.2.1. Multimodal emotion analysis

One key contribution in Calvo and D’Mello’s survey is the review of multimodality in emotion detection, which is based on the fact that emotions simultaneously activate multiple physiological and behavioral response systems. Thus, for example, specific verbal, facial and postural expressions are activated by emotions such as surprise or anger. A method for emotion detection that analyzes all three signals should exhibit better performance than methods that are based only on recognizing facial cues or vocal expressions. However, they also notice that *systems that integrate information from different modalities have been widely advocated but rarely implemented* [8]. This concern

is shared by an earlier survey conducted by Zeng et al. [21], who discuss the challenges involved in the development of multimodal fusion analysis, including audiovisual, linguistic and paralinguistic, as well as multi-cue visual fusion based on facial expressions. Yet another survey by Haq and Jackson [13] covers recent efforts in emotion recognition using audio, visual and combined audio-visual modalities. This survey also emphasizes the importance of constructing suitable datasets and discusses fifteen audio, visual and audiovisual databases. All surveys point out the need for facilitating the integration of multiple approaches to emotion analysis, which is precisely the aim of our proposed platform.

Among the efforts that have involved emotion detection from multiple channels, Kapoor and Picard [14] relied on facial and postural analysis, as well as on activity logs to support an affect-aware learning environment. Chuang and Wu [4] coupled speech and text feature analysis and provide evidence of improved emotion recognition accuracy when these two modalities are used as compared to the individual approaches. Baenziger, Grandjean, and Scherer [1] studied human abilities to recognize emotion by means of an instrument that considers facial, voice and corporal expressions. They found that visual and auditory recognition appear to be separate abilities that are independent of personality dispositions. These cases illustrate that relying on more than one modality can improve emotion recognition, but also that adding modalities or integrating their results pose significant challenges, which we aim to address with our work.

Calvo and D’Mello [3] refer to three main methods for combining signals from different sensors in multimodal emotion analysis: *Data fusion*, *feature fusion*, and *decision fusion*. Data fusion can be applied when the signals have the same temporal resolution, such as physiological signals generated by the same sensing equipment. In general, data fusion is hard to accomplish due to the need of high precision synchronization of the sensors. Feature fusion refers to the combination of features from multiple signals. For instance, features such as the mean, median, maxima or minima from electrocardiograms and electromyograms can be combined along with the specific features for each device to support inference about emotions. Decision fusion works at a higher level by merging the output of emotion classifiers for each of the signals. This method proposes that affective states can be determined for each input and later integrated so as to infer user emotions based on the various sensors.

Table 1
A comparison of tools and libraries for emotion analysis

Project	Recognized Emotions	Maintains History	Open Source	Multiple Users	Classification Method	Provides API	Reference
eMotion	6 basic emotions	Yes	No	No	Facial, motion vectors and motion units	No	[20]
FaceReader	6 basic emotions	Yes	No	No	Facial, Active Appearance Model	Yes	[7]
SHORE	4 basic emotions (happiness, anger, sadness, surprise)	No	No	Yes	Facial	Yes	[12]
RealEyes	Boredom, anger, anxiety, stress, surprise, euphoria, joy	No	Yes	No	Facial, physiological, postural	No	[19]
Semaine API	Application-dependent	No	Yes	No	Multimodal	Yes	[18]

Applications that use the services of our proposed Vikara platform can incorporate emotion analysis components that are based on any of the available techniques, provided those components produce their results in a simple format that indicates mainly an inferred emotion and its intensity. Multi-modality can occur at the level of the emotion analysis component, which can apply data or feature fusion, but most importantly, Vikara is distinct in that applications can implement decision fusion multimodality by integrating inference results from various emotion analysis components.

2.2.2. Tools and libraries for emotion analysis

We also have surveyed some of the currently available software tools for emotion analysis, some of which potentially could provide input for applications that participate in the Vikara platform. A summary of our findings is presented in Table 1.

As observed in the table, users and developers of affect-aware applications have a wide range of tools and specialized libraries at their disposal. Most of them have focused on interpreting only 4–6 basic emotions based on a single signal. Thus, for example, eMotion [20] provides a database of facial natural expressions associated to emotions, which are used to experiment with several machine learning algorithms for emotion detection, including Bayesian networks, support vector machines and decision trees. An interface has been developed that presents an interpretation of the user’s affective state in terms of emotion components, which can be interpreted by an observer, for instance, by considering those emotions with higher weight. For example, the user in Fig. 1 can be consid-

ered to be angry, as this is the predominant detected emotion (55%).

FaceReader [7] is a proprietary tool that has reported increasing success rates when recognizing emotions from facial expressions by using the Active Appearance Model [5], which is based on a deformable face template placed over an image, returning the most likely face position. Over 10,000 manually annotated images are used to train the recognition component.

SHORE [12] is a C++ library that has reported useful levels for emotion recognition by focusing on only four basic affective states of the user: happy, surprised, angry and sad. A key distinction of this tool is its capability for recognition of emotion from video input with several simultaneous faces (see Fig. 2).

RealEyes [19] is a tool for emotion recognition that has been used to support usability studies. Its recognition capabilities have explored the use of signals from physiological sensors, voice and posture. This multimodality, however, can be considered to perform feature fusion, as the individual signals do not become accessible to the user or other applications.

Semaine API [18] is a framework intended to support the development of emotion-oriented applications. It uses a message-oriented middleware for all communication in the system. As a result, all communication is asynchronous, which decouples the various parts of the system.

The Semaine API shares some of the goals of our Vikara platform, particularly those related to integration of components and enabling multimodality. Reportedly, as opposed to Vikara, the Semaine API does not provide a way to maintain a history of emotions through user sessions, does not implement a compo-

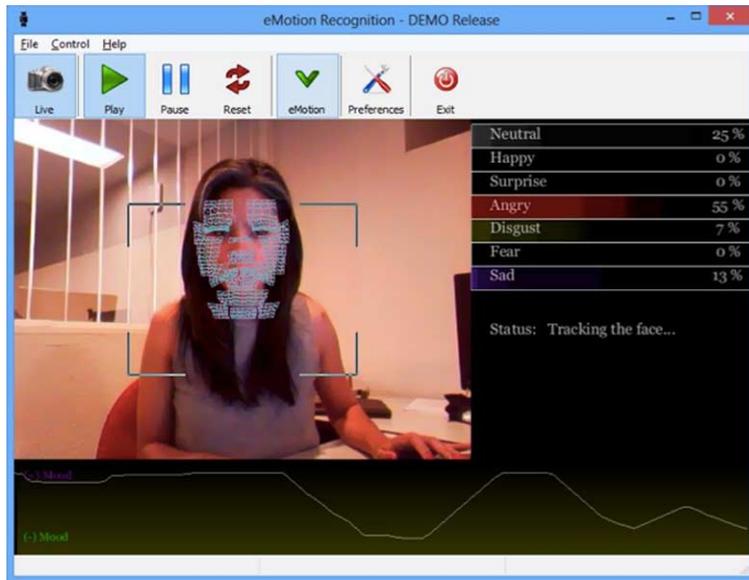


Fig. 1. Affective states in eMotion.

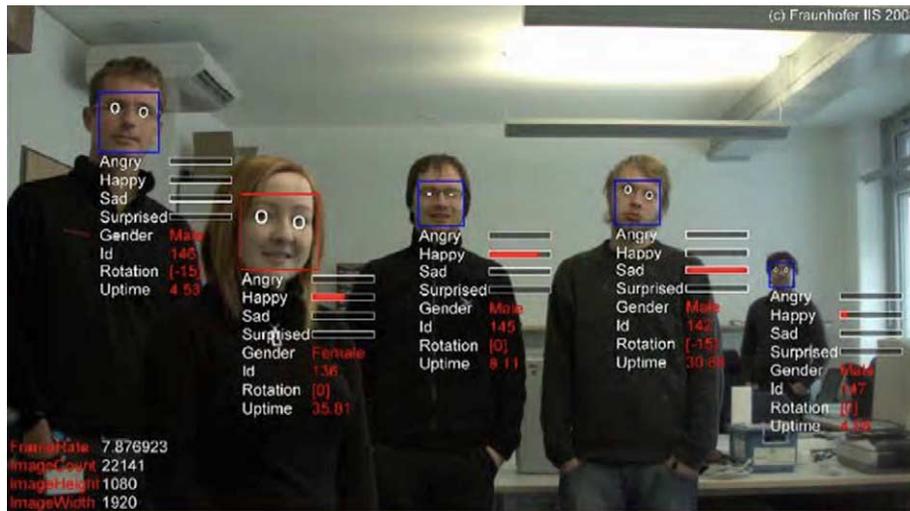


Fig. 2. Affective states of multiple subjects in SHORE. (Image used with permission.)

ment for visualizing sensed emotions from multiple sources as they occur (instead, it provides visualization means for an administrator to follow message exchanges), and has been tested only with very simple applications to explore its basic messaging facilities. Whereas Semaine API emphasizes the use or the development of simplified representation formats, Vikara relies on a simple representation format and emphasizes modularization so applications can take advantage of interpretation results from existing or newly developed emotion analysis components.

Given the short-lived nature of emotions, keeping track of what users have experienced becomes important for the purpose of analysis. In this respect, eMotion and FaceReader stand out as they allow observers to access previously experienced emotions. In this sense, these tools provide support for maintaining records of detected emotions as well as functionality for applications to make these records available for analysis by observers after user sessions have ended. In general, facial expressions are the basis for emotion analysis, though more recent versions of various

tools are exploring voice and posture analysis. Some tools can be used directly to support emotion detection in activities such as usability testing, whereas others, notably FaceReader, SHORE and Semaine provide programming interfaces. Most tools are proprietary and commercially available, though the Semaine API is available as open source.

In summary, we have identified six desirable features in tools and libraries for emotion analysis: Recognition of at least the main primary emotions, ability to maintain record of detected emotions, availability as open source, handling multiple users, use of diverse emotion detection and classification techniques, and development support via APIs. As noted in Table 1, various existing tools offer some of these features, but none includes all of them. We do not advocate the development of a monolithic tool that provides all desirable features. Instead, we aim to provide an integration platform that allows developers to build emotion-aware applications that take advantage of existing tools and libraries. In this way, existing and novel emotion analysis techniques can be implemented as components that are incorporated (via our platform) into applications depending on the requirements of each situation or problem domain.

3. The Vikara platform

In order to provide a design rationale for our integrative platform, we first discuss a typical scenario in which applications that involve emotion analysis are developed at present, particularly in the context of ambient intelligence. Then we present a conceptual overview of our integrative framework, as well as a detailed description of its prototypical implementation.

3.1. Context scenario

Consider a team that is working on applications that take advantage of emotion analysis in order to support the following activities: detecting lies during interrogation sessions, accompanying students of a tutoring system, monitoring alertness of pilots during long flights, and assessing user satisfaction during usability studies. In each case, tools and techniques for affect sensing may vary. For instance, detecting whether a pilot is somnolent or distracted may rely on analysis of facial cues in a video signal coming from a high-resolution

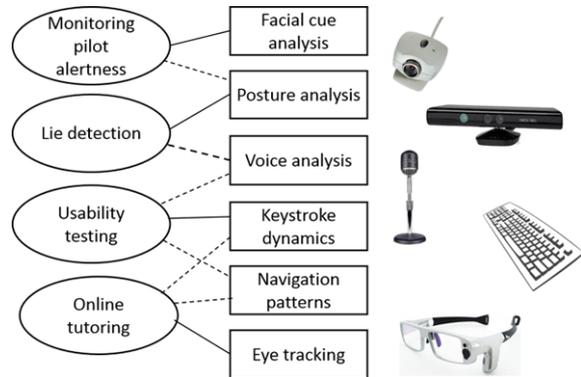


Fig. 3. Applications involving emotion detection and interpretation techniques and technologies.

camera, but developers may also want to take advantage of posture analysis techniques. Similarly, lie detection may be using posture analysis as obtained from a Kinect sensor, but the voice channel can enhance performance if properly used. Usability tests may take advantage of keystroke dynamics and video processing, but its results can be enriched with voice analysis and keystroke dynamics or navigation patterns considerations. Finally, tutoring aids may rely on eye tracking mechanisms, but could produce better results if navigation patterns or keystroke dynamics are taken into account. Developers may want to experiment with multiple techniques and technologies in order to improve various aspects of emotion analysis. Some applications may rely on existing tools and libraries; others may require ad-hoc development.

In the current scenario, each application incorporates emotion analysis mechanisms as required by developers to explore different approaches or to enhance existing components, thus typically replicating, adapting and oftentimes duplicating code and effort. Extending an application to incorporate an additional technique or algorithm (in order to implement decision fusion multimodality, as described in the previous section) requires significant code changes.

We have aimed to provide a platform to make it possible for developers to incorporate existing or newly created emotion analysis techniques into applications with only minimal changes to code, so multiple approaches can be explored and combined. Figure 3 illustrates this target scenario. In this figure, applications are represented by ovals, whereas techniques for affect sensing are represented by rectangles. Technologies that can be used for implementing some techniques, such as video cameras, microphones, Kinect

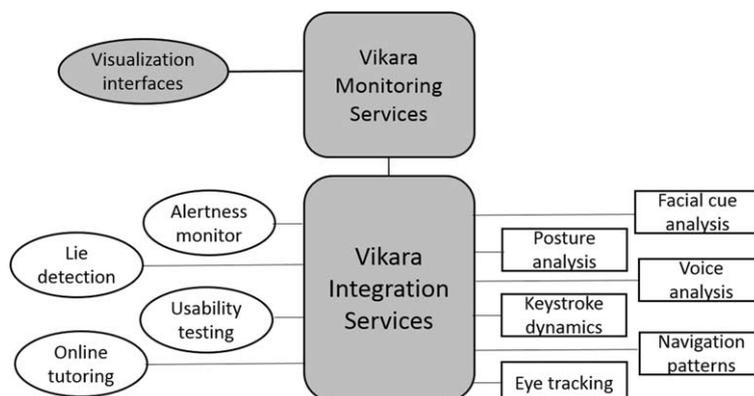


Fig. 4. Vikara's integration, monitoring and visualization components.

sensors, keyboards and eye-tracking glasses are represented on the right side of the figure. Solid lines indicate that an application relies on a given technique, whereas dashed lines indicate that the application may benefit from exploring or integrating a complementary technique.

In what follows, we discuss how the Vikara platform makes it possible for applications to integrate results produced by tools for affect sensing in a seamless manner.

3.2. Vikara services

As illustrated in Fig. 4, Vikara introduces two major components into the scenario presented in Fig. 3: Integration services and monitoring services. Integration services provide a uniform representation for affect expressions as reported by diverse tools that implement various techniques for affect sensing. Monitoring services use the persistent representation of emotions created by our integration services in order to offer a uniform view of the emotions being sensed.

In the Vikara platform, integration services act as a middleware component that allows developers to focus on the issues of the domain for which they are building affect-aware applications, while taking advantage of a common internal representation of emotions from potentially varied sources. For instance, instead of working on the details of emotion detection algorithms, a developer of ambient intelligence applications needs to determine the conditions under which specific events must be triggered (such as when specific lighting might be turned on in response to a depressive mood of the user). Conversely, developers who implement analysis techniques can concentrate on improving precision and effectiveness of their algorithms and providing

timely notifications, independently of the applications that will use their results.

Monitoring services can be helpful both in the development and the deployment stages, as they allow developers to access all available inputs from emotion detectors and interpreters. As noted in the figure, these inputs can be used to provide a graphical representation to show which emotions are sensed at a given instant and how emotions evolve in time for specific users or sessions. During development, this is helpful for debugging affect-aware applications. At deployment time, application managers and observers in general can rely on visualization components to explain system responses or to further support inferences regarding user behavior.

3.3. Vikara prototype

In order to demonstrate the viability and potential of the Vikara integrative platform, an operational prototype has been implemented. Integration services are implemented as web services based on three key controllers:

- *Sessions*, in charge of handling data associated with each use of a specific emotion analysis component. Data in this area includes a session identifier, an identifier for the corresponding emotion interpreter as well as for each of the emotions that are detected along with their timestamps.
- *Emotions*, in charge of keeping track of sensed emotions and their attributes, such as the interpreter that detected each of them, their intensity, and their time of occurrence. Given that multiple emotion sensing techniques can be used and they may vary in precision and interpretation, more

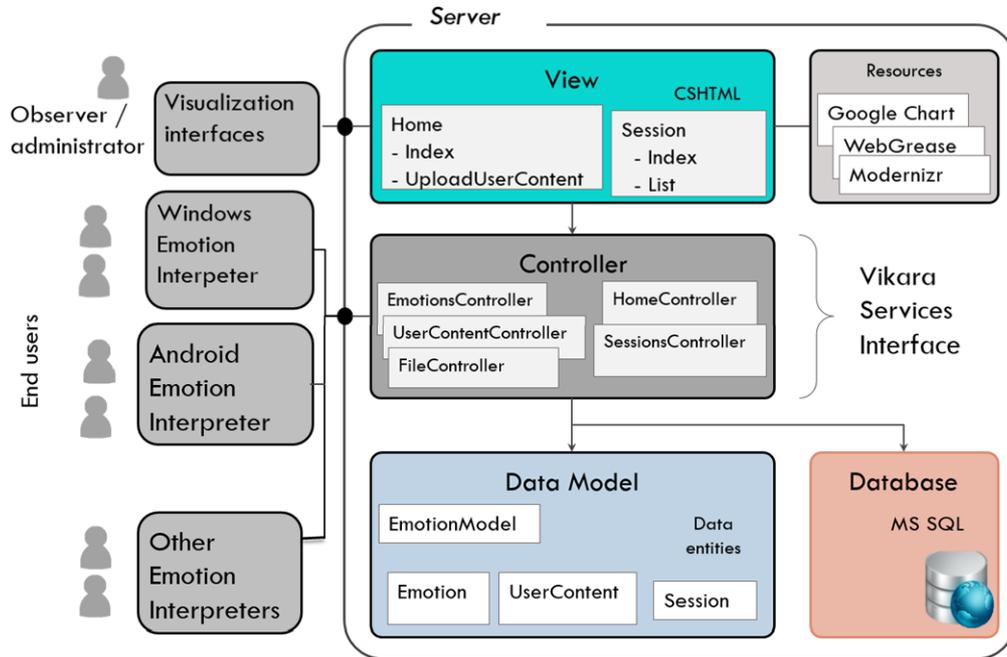


Fig. 5. Architecture of the Vikara prototype.

than one emotion can be recorded as occurring at a given time within a session. It will be up to applications to take these results from emotion sensors and to determine the best way to combine them or select those that may be used to respond appropriately, for example, in an ambient intelligence scenario.

- *User content*, in charge of acquiring contents used for detecting and interpreting emotions (e.g. video from a webcam) as well as storing contents that can be used by the monitoring and visualization components to facilitate session analysis.

3.3.1. Prototype architecture

Both applications and emotion analysis components are provided with a concise programming interface that allows them to easily start or terminate sessions, register or request emotion attributes for specific users and sessions, and uploading or unloading related contents.

The model-view-controller architecture of the prototype is illustrated in Fig. 5. The data model depicted at the bottom defines the attributes and relationships that will be stored in a database for later use by applications.

The main entities in the Emotion Model comprise users, emotions and sessions, which provide persistence needed for abstractions that appear at the Con-

troller level. Our data model for representing emotions generally follows the elements of the Emotion Markup Language (EML), proposed as a W3C Recommendation.¹ However, considering the diversity of existing tools for emotion analysis, and in order to facilitate interoperability among applications and emotion analysis components, only four attributes are defined as mandatory: an emotion identifier (*EmotionId*), an emotion name (*EmotionName*), an intensity level (*EmotionIntensity*) and the instant in which emotions are detected (*Timestamp*).

This provides sufficient flexibility for Vikara to handle any set of emotions sensed by detection tools, beyond the six primary emotions we have used for our initial explorations. Moreover, the data model covers the case in which more than one detection tool registers an emotion simultaneously. Applications can retrieve registered emotions using their identifiers or timestamps.

Applications and tools in Vikara operate in the context of sessions, which are given persistence by our data model through five attributes: *SessionId*, *StartTime*, *EndTime*, *UserContentId* and *DataProvider*.

¹Emotion Markup Language (Emotion ML 1.0). W3C Recommendation 22 May 2014. <http://www.w3.org/TR/emotionml/>. Last visited on Sept. 20, 2014.

Whereas the first three attributes identify and delimit the duration of a session, *UserContentId* refers to material that has been uploaded as part of a session (e.g., a video), and *DataProvider* identifies the emotion detection components associated to a session.

Figure 5 also shows that the View component of the Vikara architecture relies on readily available tools and libraries for providing alternative representations for the stored entities, such as charts and graphs (e.g. *Google Chart API*, *Web Grease* and *Modernizr*) needed by visualization interfaces.

Central to this architecture are the REST services provided by Vikara, which are represented by the controllers that can be accessed by applications and monitoring services. Communication is based on stateless HTTP requests, which result in very low network requirements, given the simple attributes defined for information exchange. An example of information transferred to Vikara by an emotion detection component via a POST method is shown in Fig. 6. This involves the case in which more than one emotion has been sensed with the same timestamp (disgust, with intensity 0.3; happiness, with intensity 0.5).

An application that requires emotions detected in a given session would issue a GET method. For the case of the emotions referred to in Fig. 6, with *SessionId* = 5031, the exchange would be:

```
http://localhost/Vikara/Sessions/
Resultset?sessionId=5031
```

which would generate the response:

```
HTTP/1.1 200 OK
[...]
var resultset = [
[...]
['Second', 'Anger', 'Disgust', 'Fear', '
Happiness', 'Sadness', 'Surprise'],
[0,0,50,0,0,50,0],
...]
```

followed by all the values for emotions detected at different times during the session.

3.3.2. Visualization interface

In order to demonstrate monitoring capabilities offered by Vikara, a client application has been implemented which provides support for basic administration tasks involving user contents and associated sessions and emotions. Key functionality of this component is illustrated in Fig. 7.

In the figure, emotions are visualized as they are detected and interpreted by an emotion analysis application (such as the ones described in the following section), which infers emotions based on facial cues. In this scenario, the user is watching a video, which is shown on the top left area. Facial expressions can be observed in the video shown on the bottom right area. As these expressions change, emotions are inferred by the application and reported, along with an intensity level, via the Vikara integration services. The visualization client listens to changes reported by the monitoring services and produces a chart that represents graphically the emotions occurring at a given instant. This is shown on the top right area. As noted earlier, if multiple emotion analysis techniques are used, more than one emotion can be reported at a given instant, and this can easily be observed graphically, as different colors indicate different emotions, whereas emotion intensity is related to bar lengths.

As the session progresses, emotion changes are displayed on a line chart (bottom left) that shows coincidences and discrepancies among emotion analysis components, as well as the effect of what the user is watching on her affective state. In this case, the chart shows that the user expresses some light disgust (red line, intensity 50) at the beginning of the video, but not much later on she expressed happiness (green line, intensity 100, when the subject in the video started dancing). This sort of analysis can be very helpful, for example, in scenarios such as user experience studies related to software or other products in which the observer is interested in how the user receives specific modules or subparts that may require adjustments or further development.

4. Using and evaluating an integrative platform

In order to validate and demonstrate the potential of the Vikara platform, we have proceeded as follows: (1) We have developed two applications for emotion detection; (2) we have had a programmer integrate applications into Vikara; and (3) We have made the platform as well as the test applications publicly available for experimentation by other researchers.

Our platform does not impose any requirements on the applications regarding their internal structure, hardware utilized or techniques for emotion recognition on which they rely. We demonstrate this flexibility by developing two very diverse applications: One that relies on facial cues analysis and takes ad-

```

{
  "UserContentId": 1005,
  "DataProvider": "Kinect Basic Emotions 1.5",
  "StartTime": "2013-05-22T21:39:56.310002Z",
  "EndTime": "2013-05-22T21:40:18.398895Z",
  "CapturedEmotions": [
    {"Name": "Anger", "Intensity": 0, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Disgust", "Intensity": 0.3, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Fear", "Intensity": 0, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Happiness", "Intensity": 0.5, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Sadness", "Intensity": 0, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Surprise", "Intensity": 0, "Timestamp": "2013-05-22T21:39:57.493424Z"},
    {"Name": "Anger", "Intensity": 0, "Timestamp": "2013-05-22T21:39:58.5598056Z"},
    {"Name": "Disgust", "Intensity": 0.3, "Timestamp": "2013-05-22T21:39:58.5598056Z"},
    {"Name": "Sadness", "Intensity": 0, "Timestamp": "2013-05-22T21:40:17.654629Z"},
    {"Name": "Surprise", "Intensity": 0, "Timestamp": "2013-05-22T21:40:17.654629Z"}
  ],
}

```

Fig. 6. Information exchanged for emotions in a session.

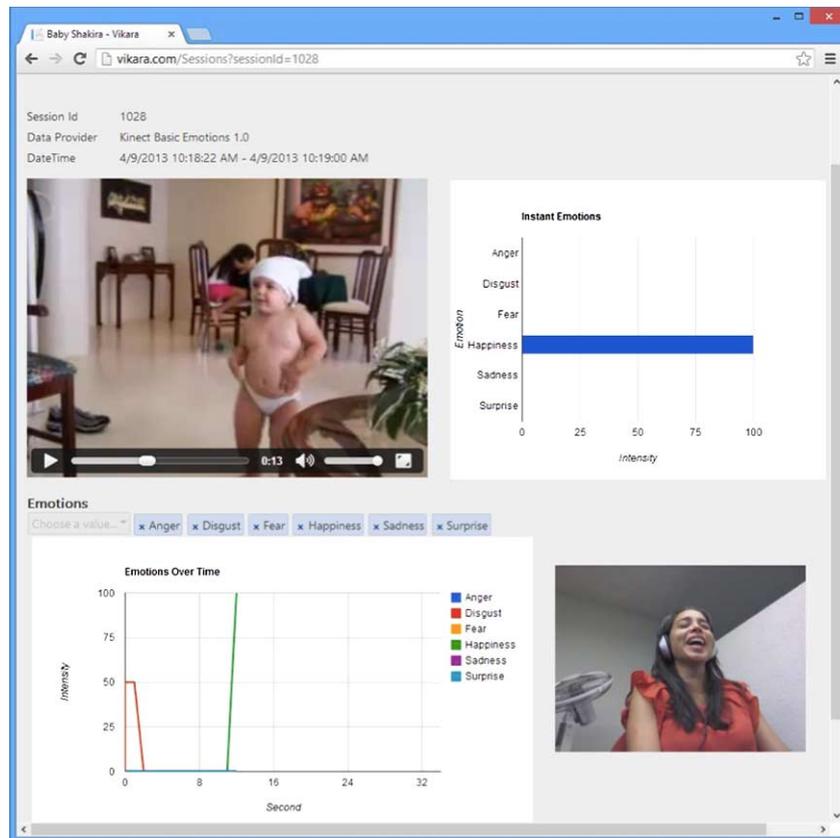


Fig. 7. Visualization of emotion analysis.

vantage of the Kinect sensor, and another that is designed to run on an Android mobile device and to allow for user self-report of emotions. Applications do have to comply with the simple messaging protocol defined by Vikara. This can be taken into consideration when the application is designed, or a wrapper can

be implemented for emotion recognition components so their results are produced in the format required by Vikara.

We describe the two test applications in the following sections. Then, we report results of their integration into Vikara.

Table 2

AUs values handled by the Kinect's Face Tracking SDK (adapted from [15])

AU	AU Values
Neutral face	All AUs 0
AU0 – Upper lip raiser	1 = showing teeth fully -1 = maximal possible pushed down lip
AU1 – Jaw lowerer	1 = fully open -1 = closed
AU2 – Lip stretcher	1 = fully stretched -.5 = rounded (pout) -1 = fully rounded (kissing mouth)
AU3 – Brow lowerer	-1 = raised almost all the way +1 = fully lowered
AU4 – Lip corner depressor	-1 = very happy smile +1 = very sad down
AU5 – Outer brow raiser	-1 = fully lowered (very sad face) 1 = raised as in deep surprise

4.1. A Kinect-based implementation of FACS

The Facial Action Coding System (FACS) developed by Ekman & Friesen [10] has been used with relative success both by psychologists and programmers as an aid for analyzing human expressions and inferring affective states. We decided to implement an emotion analysis component that takes advantage of existing functionality for obtaining and analyzing facial cues, available in the Kinect's Face Tracking software developer kit (SDK), and apply FACS for emotion interpretation. The Face Tracking SDK refers to simple animation units (AUs, generally equivalent to Action Units) with values ranging between -1 and $+1$, as summarized in Table 2.

Our emotion analysis component couples results from the Face Tracking SDK with a technique similar to the FACS Affective Interpretation Dictionary (FACSAID), which maps combinations of action units to corresponding emotions [11]. FACSAID is a project aimed at providing links between facial expressions and their psychological interpretations, particularly applying FACS. Their approach is described as a dictionary because users can look up meanings for a particular facial behavior, or look up the facial behaviors that imply a particular meaning or emotion. The main combinations of AUs used to interpret emotions in our prototypical implementation are summarized in Table 3.

Emotions inferred in this way are registered with the Vikara integration services for use by applications. In this case, our desktop application, which we have termed *Vikrti* (Sanskrit for *modification*), is oriented to

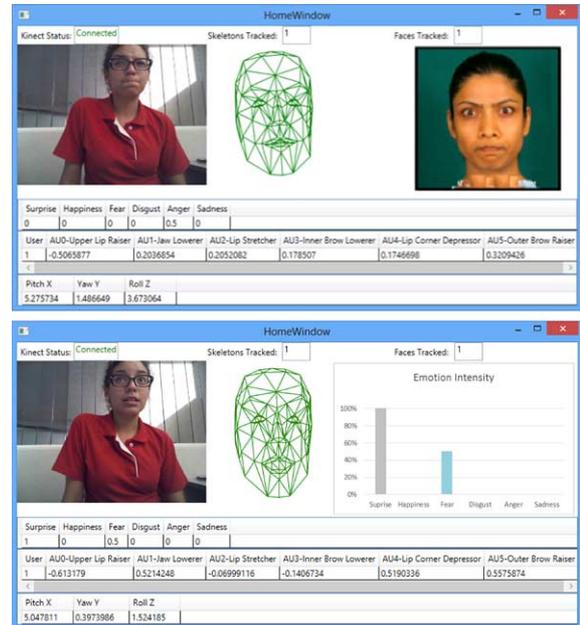


Fig. 8. User interface for an observer in the Vikrti emotion interpreter.

emotion recognition using the Kinect sensor, and is intended for use in a variety of settings. Vikrti provides a visual interface only for an observer, which is illustrated in Fig. 8.

A user whose emotions are being analyzed can be presented, for example, with tangible products or software interfaces for which user feedback is needed, along with the Kinect sensor for detecting facial cues, but does not actually need to be aware of the operation of Vikrti.

Figure 8 shows the observer's side of the scenario used to evaluate the performance of Vikrti, in which a session has been started and a subject is presented with a series of pictures of faces that have been previously classified as representing known emotions. The setting for the user whose emotions are being analyzed is illustrated in Fig. 9.

The observer has access to each picture being presented (right area) as well as to the video of the user, who is asked to produce the same emotion represented in the picture. The interface in Fig. 8 also shows a so-called skeleton of the subject's face, as recognized by the Kinect sensor (the Face Tracking SDK allows for multiple faces to be tracked, but we only are relying on one at the moment). What is more important for the purposes of our work, the interface shows numeric values obtained from Vikara integration services for basic emotions occurring during the session. In the up-

Table 3
AUs and values used for emotion interpretation

AU\Emotion	Happiness	Anger	Fear (1)	Fear (2)	Disgust	Surprise	Sadness
AU0 – Upper lip raiser					>.1		
AU1 – Jaw lowerer			>0			>.4	
AU2 – Lip stretcher	>0		<0				>0
AU3 – Inner brow lowerer		>.15		<0			>.35
AU4 – Lip corner depressor	<0		>0				>0
AU5 – Outer brow raiser				>0			>0

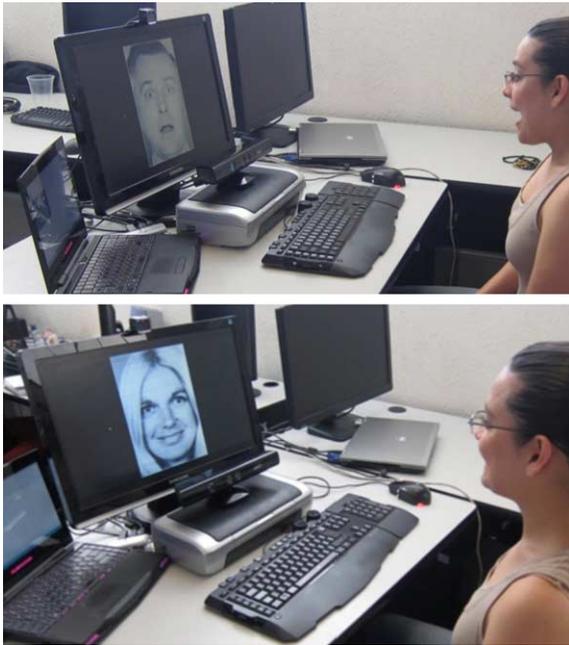


Fig. 9. Setting for a user of the Vikrti emotion interpreter.

per part of the figure, *anger* has been successfully recognized, as it is the only emotion in the list that has received a non-zero value. For further justification of this interpretation, the interface also shows the values determined by the emotion analysis component, which ultimately are provided by the Kinect's Face Tracking SDK functionality. In the lower part of the figure, it can also be noticed that *surprise* is properly recognized, but *fear* is also suggested as an alternative interpretation.

It is interesting to note that in spite of its straightforward approach, Vikrti shows that the Kinect-based emotion interpreter provides meaningful and potentially useful results, at least for pure emotions. Recognition of produced emotions such as surprise and fear was very good (above 75% of the time), whereas sadness, anger and happiness were recognized correctly between 40% and 60% of the time. Consistently with

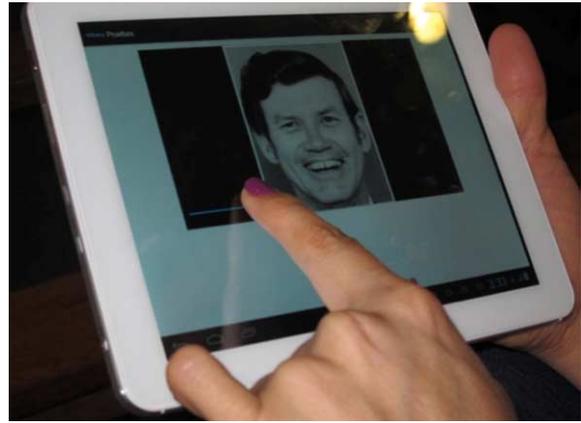


Fig. 10. Android-based self-reporting application.

results of more elaborate tools (such as FaceReader and SHORE), disgust is an emotion that is harder to recognize correctly (only 33% of the time). Complete results of experimentation with Vikrti are reported in [6].

4.2. An Android-based emotion interpreter

Vikara has been designed to facilitate the integration of applications that share a common representation of emotion interpretations, independently of the operating system platforms or programming languages in which they are implemented. In order to demonstrate this claim, we also developed a self-report application in which users can indicate emotions being experienced as well as their corresponding intensity levels by using an Android-based device such as a tablet or a smartphone. We refer to this application as *Avyatha* (Sanskrit for *subtle emotion*). Its user interface is illustrated in Fig. 10.

In this interface, the user selects the image that best represents his or her emotion at a given time, as well as an intensity level. This information is registered with Vikara's integration services and can be made available to monitoring services or to other applications.

As in the case of Vikarti, Avyatha can be useful in diverse settings, including intelligent environments. For instance, in the context of a guided tour through points of interest in a smart city, if a user expresses positive emotions regarding a specific site, the system can activate recommendations of relevant related sites, objects in the surroundings, or services for the visitor.

4.3. Integration and visualization

Both of our prototypical applications include emotion recognition components that present their results by following the format defined by Vikara, thus making their integration straightforward. In each case, we provided our programmer with the specification of the available services as well as the format handled for representing emotions by each application. We defined that integration would be completed if it was possible for Vikara's visualization service to display emotions at the time they were detected. In the case of the Kinect-based application, videos presented to users for studying their reactions were uploaded by taking advantage of Vikara's functionality to acquire user content, as described in Section 3.3. Applications were able to communicate with Vikara via methods available for defining sessions, registering emotions, and getting emotions in the JavaScript Object Notation (JSON) required by the platform.

In terms of our established goals, the integration of our applications was successful, as their emotion-related data were displayed in the visualization interface, as illustrated in Fig. 7. Integration proceeded smoothly, requiring about two hours for the self-report application, and close to four hours for the Kinect-based application (including video uploading times).

We consider our integration experiments provide a reasonable demonstration of the potential of our platform. We do acknowledge, however, that integration times will vary depending on programming skills and on whether additional mappings are required in order to convert emotion detection results from existing software components into the simple format required by Vikara. Regarding the construction of applications that take better advantage of the availability of multiple components for emotion analysis, we are in the process of exploring these concepts in the context of supporting usability studies.

4.4. Making Vikara publicly available

We have made the Vikara platform as well as our prototypical applications publicly available from our

lab's website (see <http://ict.udlap.mx/vikara>), so other developers can explore and experiment with our approach. Installing the main software components requires only an applications server and a database management system (we have used Microsoft's SQL Server, but this can be changed, for example, to MySQL). Participating applications may have their own specific requirements. In the case of the applications described in the previous sections, installation and use requires the Kinect sensor or an Android mobile device.

5. Discussion

Affective computing is becoming increasingly important for Ambient Intelligence (AmI), and a significant number of applications are incorporating (or should be incorporating) emotion analysis capabilities. Smart spaces, smart cities, intelligent buildings, ubiquitous computing, and in general ambient intelligence (AmI) applications pose challenges for researchers to make it possible to respond to user needs and preferences in timely, intuitive and adaptive manners. Although user models and user profiles can help in adapting system responses to personalities and general sentiments of the users, contextual and situational changes affect the user's expectations and emotions, thus requiring real-time adaptivity from AmI applications.

Whereas emotion analysis still requires significant work before it can be deployed in open AmI applications that respond effectively to user emotions that are expressed spontaneously, a lot can be learned from current work in more controlled environments, with less interference and more stable conditions in terms of variables such as user movement, lighting or number of users.

Thus, we posit that progress in emotion-oriented adaptivity in intelligent tutoring systems, for example, should contribute towards the incorporation of affective computing into smart lecture rooms or laboratories. Similarly, advances in applying emotion analysis in product evaluation for marketing purposes with one or several users, represent strides towards affect awareness in open shopping environments, with multiple, intermittent users. Also, applications of affective computing in training for specialized staff interpersonal communication, as well as for actors in performance arts, will contribute to generate a rich corpora of signals (facial and body expressions, audio and video recordings) that will be key in supporting improved inference of affective states.

In a similar vein, even though affect-aware applications for AmI are still being developed in relative isolation, it is critical to provide means for developers to facilitate experimentation with techniques for emotion analysis and for observers and behavioral scientists to obtain a global view of emotion analysis performance. Our work on the Vikara integrative platform is intended as a contribution in that direction.

We have started work in the area with the goal of incorporating emotion analysis considerations into processes for evaluating usability and user experience with interactive systems [16]. Our prototypical implementation of a component for emotion analysis based on FACSaid and the Kinect's Face Tracking SDK explores this line, in addition to its role for demonstrating how an application can be integrated by using the Vikara platform.

6. Conclusion

We have presented Vikara, a platform for integration and management of applications for emotion analysis. The main contribution of this work is an architecture that modularizes components for emotion analysis and separates them from, but makes them accessible to, the components that implement the logic of affect-aware applications. We accomplish this by introducing middleware that offers communication services to both applications and emotion analysis components through a simple protocol, as well as monitoring services that allow for observers to obtain a global view of emotion sensing and interpretation from diverse sources. We have demonstrated the feasibility of this architecture by developing an operational prototype of our integration platform as well as two initial applications that implement distinct approaches to emotion interpretation, and that have been successfully integrated into Vikara.

Pending work on the Vikara platform includes the development and integration of more affect-aware applications (such as those discussed in the previous section), as well as the integration of libraries (such as SHORE) to make them available to applications. Though we have demonstrated the validity and potential of our approach, definitive evidence of its practical usefulness and scope will be obtained as other developers and researchers experiment with the integration of new components for emotion analysis and are able to monitor their results and incorporate them, through Vikara, into applications that implement multimodal affect recognition.

Stress testing of the current platform is also needed, considering particularly the use of a relational database for providing persistence to sessions and emotion data, and the need to respond in real-time to a large number of user sessions. More AmI-specific functionality that could be considered for Vikara includes reporting, monitoring and visualizing user identification, as well as the history of application responses and corresponding emotional reactions from the users.

Acknowledgments

Support for this research has been provided by the CUDI-Conacyt Program for Advanced Applications, the National High Technology Research and Development Program of China (863 Program) under Grant No. 2013AA01A603 and the National Nature Science Foundation of China under Grant No. 61373084.

References

- [1] T. Baenziger, D. Grandjean and K.R. Scherer, Emotion recognition from expressions in face, voice, and body. The multimodal emotion recognition test (MERT), *Emotion* **9**, 691–704.
- [2] S. Brave and C. Nass, Emotion in human-computer interaction, in: *The Human-Computer Interaction Handbook*, J.A. Jacko and A. Sears, eds, L. Erlbaum Associates Inc., Hillsdale, NJ, 2002, pp. 81–96.
- [3] R.A. Calvo and S.K. D'Mello, Affect detection: an interdisciplinary review of models, methods, and their applications, *IEEE Transactions on Affective Computing* **1**(1) (2010), 18–37.
- [4] Z.J. Chuang and C.H. Wu, Multi-modal emotion recognition from speech and text, *Int'l J. Computational Linguistics and Chinese Language Processing* **9** (2004), 1–18.
- [5] T. Cootes and C. Taylor, Statistical models of appearance for computer vision, Technical report, University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering, 2000.
- [6] X. Cortés, Vikara: plataforma de desarrollo y gestión de aplicaciones para detección de emociones, BEng thesis, Department of Computing, Electronics and Mechatronics, Universidad de las Américas Puebla, May, 2013.
- [7] M.J. den Uyl and H. van Kuilenburg, FaceReader: an online facial expression recognition system, in: *Proc. of the 5th International Conference on Methods and Techniques in Behavioral Research*, 2005, pp. 589–590.
- [8] S. D'Mello and A. Graesser, Multimodal semi-automated affect detection from conversational cues, gross body language and facial features, *User Modeling and User-Adapted Interaction* **10** (2010), 147–187.
- [9] P. Ekman, An argument for basic emotions, *Cognition and Emotion* **6**(3/4) (1992), 169–200.

- [10] P. Ekman and W.V. Friesen, The facial action coding system: a technique for measurement of facial movement, 1978, Web document, available from: <http://face-and-emotion.com/dataface/facs/description.jsp>.
- [11] P. Ekman and E.L. Rosenberg, eds, *What the Face Reveals: Basic and Applied Studies of Spontaneous Expression Using the Facial Action Coding System (FACS)*, 2nd edn, Oxford University Press, USA, 2005.
- [12] E. Ernst, R. Tobias and C. Kueblbeck, A modular framework to detect and analyse faces for audience measurement systems, in: *Proc. of the 2nd Workshop on Pervasive Advertising (2009)*, 2009, pp. 75–87.
- [13] S. Haq and P.J.B. Johnson, Multimodal emotion recognition, in: *Machine Audition*, W. Wang, ed., IGI Global, pp. 398–423.
- [14] A. Kapoor and R.W. Picard, Multimodal affect recognition in learning environments, in: *Proc. 13th Ann. ACM Int'l Conf. Multimedia*, 2005, pp. 677–682.
- [15] Kinect, Kinect for Windows programming SDK 1.8. Online document, available from: <http://msdn.microsoft.com/en-us/library/jj130970.aspx>, Last accessed on Feb. 24, 2014.
- [16] Y.A. Méndez, C.A. Collazos, T. Granollers and J.A. Sánchez, Evaluating interactive systems from an emotional perspective, *Revista Científica Guillermo de Ockham* **12**(1), 43–49.
- [17] R.W. Picard, *Affective Computing*, The MIT Press, 1997.
- [18] M. Schröder, The SEMAINE API: towards a standards-based framework for building emotion-oriented systems, *Advances in Human-Computer Interaction* (2010), 21, Article ID 319406, DOI:10.1155/2010/319406.
- [19] R. Schultz, C. Peter, M. Blech, J. Voskamp and B. Urban, Towards detecting cognitive load and emotions in usability studies using the RealEYES framework, in: *Proc. of the 2nd International Conference on Usability and Internationalization*, Springer, Berlin, Heidelberg, 2007, pp. 412–421.
- [20] Y. Sun, N. Sebe, M.S. Lew and T. Gevers, Authentic emotion detection in real-time video, in: *Computer Vision in Human-Computer Interaction*, N. Sebe, M. Lew and T.S. Huang, eds, Vol. 3058, Springer, Berlin, Heidelberg, 2004, pp. 94–104.
- [21] Z. Zeng, M. Pantic, G.I. Roisman and T.S. Huang, A survey of affect recognition methods: audio, visual, and spontaneous expressions, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **31**(1) (2009), 39–58.