IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 2, 2022, accepted March 23, 2022. Date of publication xxxx 00, 0000, date of current version xxxx 00, 0000.

*Digital Object Identifier 10.1109/ACCESS.2022.3163523*

# Toward Accountable and Explainable Artificial Intelligence Part Two: The Framework Implementation

**JORDAN VICE AND MASOOD MEHMOOD KHAN, (Member, IEEE)**
Faculty of Science and Engineering, Curtin University, Perth, WA 6102, Australia
Corresponding author: Masood Mehmood Khan (masood.khan@curtin.edu.au)

**ABSTRACT** This paper builds upon the theoretical foundations of the Accountable eXplainable Artificial Intelligence (AXAI) capability framework presented in part one of this paper. We demonstrate incorporation of the AXAI capability in the real time Affective State Assessment Module (ASAM) of a robotic system. We show that adhering to the eXtreme Programming (XP) practices would help in understanding user behavior and systematic incorporation of the AXAI capability in Machine Learning (ML) systems. We further show that a collaborative software design and development process (SDDP) would facilitate identification of ethical, technical, functional, and domain-specific system requirements. Meeting these requirements would increase user confidence in ML and AI systems. Our results show that the ASAM can synthesize discrete and continuous models of affective state expressions for classifying them in real-time. The ASAM continuously shares important inputs, processed data and the output information with users via a graphical user interface (GUI). Thus, the GUI presents reasons behind system decisions and disseminates information about local reasoning, data handling and decision-making. Through this demonstrated work, we expect to move toward enhancing AI systems' acceptability, utility and establishing a chain of responsibility if a system fails. We hope this work will initiate further investigations on developing the AXAI capability and use of a suitable SDDP for incorporating them in AI systems.

**INDEX TERMS** Artificial intelligence, explainable artificial intelligence, affective computing, system design, classifier design, interactive graphical user interface, human–computer interface.

## I. INTRODUCTION

Several researchers have reported that Artificial Intelligence (AI) experts and software engineers lead the artificial intelligence (AI) system design processes [1]–[3]. Practitioners, usually less involved in the process, find the existing eXplainable Artificial Intelligence (XAI) frameworks as algorithm-driven, lacking domain-specific considerations and offering frail explanations [4]. Recent works [5]–[7] report several gaps in the prevailing XAI capabilities and practitioners' needs. Such gaps can be filled by embedding explainability and transparency in Machine Learning (ML) and AI systems. Presenting statistical and probabilistic data alone can not help practitioners in understanding how domain-specific requirements were met [4], [5]. Particularly, the limited amount of explanations given by AI and ML systems do not suffice and comply with regulatory and industrial standards [8], [9]. Furthermore, an agreed and proven method of determining non-explainability of AI systems is not yet available to let practitioners assess XAI capabilities [10].

It is proposed that four system features: the quality of inputs and interactions between them, the method of combining the input information, the quality of the training data and, trustworthiness of the system decisions would suffice incorporating the XAI in ML and AI systems [11]. However, real-life use of these features in incorporating XAI is not common. Our proposed AXAI capability framework extends the generic XAI capability to enable ML and AI systems share their decisions and adequately explain the underlying reasoning processes. The existing XAI methods would neither separate nor quantify measures of comprehensibility, accuracy and accountability. Thus, incorporating and assessing explainability in AI systems remain difficult. The AXAI framework facilitates explaining reasons behind

The associate editor coordinating the review of this manuscript and approving it for publication was Jolanta Mizera-Pietraszko.

system decisions using elements of comprehensibility, predictive accuracy and system accountability. It measures comprehensibility as the readiness of a human to apply the acquired knowledge. The system accuracy is measured in terms of the ratio of the test and training data, training data size and the observed number of false-positive inferences. Finally, the AXAI framework measures accountability in terms of the inspectability of the input cues, the processed data and the output information, for addressing any legal and ethical issues. As such, the AXAI framework facilitates separation, measurement and delineation of elements embedded in each; comprehensibility, predictive accuracy and accountability, in a three-dimensional space.

In order to move toward building Accountable eXplainable Artificial Intelligence (AXAI) capable AI systems, we introduced the theoretical foundations of the AXAI capability framework in an accompanying paper entitled "Toward Accountable and Explainable Artificial Intelligence Part one: Theory and Examples." The proposed AXAI framework provides a systematic approach of delineating AI systems in a three-dimensional (3D) space consisting of three mutually perpendicular axes viz., accuracy, comprehensibility and accountability.

This paper demonstrates incorporation of the AXAI capability in the real time Affective State Assessment Module (ASAM) of a robotic system. We explain how eXtreme Programming (XP) practices align with the process of incorporating the AXAI capability in AI systems. While developing an AXAI-capable AI system, each stage of the software design process was laterally aligned with elements of the design process (analysis, design, implementation and testing). We show iterative development of various modules for adding the desired AXAI elements into the ASAM [12]. As evident in the following sections, this work makes several important contributions given below:

1) It demonstrates selection and use of an appropriate Software Design and Development Process (SDDP) for adding the AXAI capability in an AI system.
2) It shows how agile software design practices could be exploited for embedding AXAI capability in an AI system.
3) It also shows how the main features of extreme programming methods would help in ascertaining user requirements and incorporating user feedback.
4) It presents a novel approach for collaboratively soliciting user input for developing appropriate use cases.
5) It introduces the architectural and functional details of a portable, real-time affective state assessment module that can be attached to a robotic system.
6) It demonstrates how displaying visual and vocal cues would facilitate dissemination of explainability information in terms of system's predictive accuracy, transparency, and accountability.
7) It presents the process of systematic, localized and iterative development of an affective computing system.
8) It presents a representative set of conceptual models and use cases for progressive enhancement of an affective state assessment system.
9) Finally, it demonstrates a process of determining user motivation, user behavior and user-perceived value of the system features.

This paper is organized as follows. After introducing reasons for integrating AXAI capabilities into AI systems in this section, we discuss limitations of the existing automated Affective State Assessment (ASA) systems in Section II. We then introduce the alignment between XP practices and the process of embedding the AXAI capability in Section III. The iterative analysis and assessment of user requirements is shown in Section IV. Section V provides an overview of the ASAM, discussing hardware and software elements and how it is considered an AXAI-capable AI system. Section VI discusses and summarizes the findings from this work before the conclusion in Section VII, where we discuss future work and the importance of integrating AXAI-capable systems moving forward.

## II. BACKGROUND

Expressing affective states is an integral part of human life. Social robots therefore need to understand and express affective states. To interact with humans, robotic systems are now equipped with peripheral devices necessary for automated affective state assessment (ASA) [13]–[15]. Humans express internal thoughts, feelings, and emotions through an array of voluntary and involuntary cues. These cues also help humans in observing and synthesizing others' expressions and assessing affective experiences [16]. To imitate the human model of ASA, algorithm-based learning and classification approaches rely on one or more of visual, vocal, psychophysiological and neural cues [13], [17]–[19]. Emerging sophisticated ASA systems use different combinations of affective state models, learning techniques and classification methods [20]–[22] as their capabilities and accuracies progressively improve [23]–[26]. Modern ASA capabilities include the dynamic assessment of affect-arousal and, multimodal and contextual assessment of affective states [18], [26]–[28]. Since ASA systems rely on algorithm-based analysis, they inherit algorithmic biases while making inferences. Like other AI systems, ASA systems lack the ability to explain, are usually opaque, and do not help in establishing a chain of responsibility when system accountability is of concern. A typical model of implementing algorithm-based ASA is shown in Fig. 1 depicting how biases and errors travel through the decision-making process, while ASA systems progressively filter affect-expressing cues down the pyramid and reach a level of reduced transparency. Figure 1 uses a normalized measure of affective state strength [29] as $-1 \leq E_A \leq 1$ where $E_A$ represents the expression of an affective state that ranges from -1 to $+1$, showing

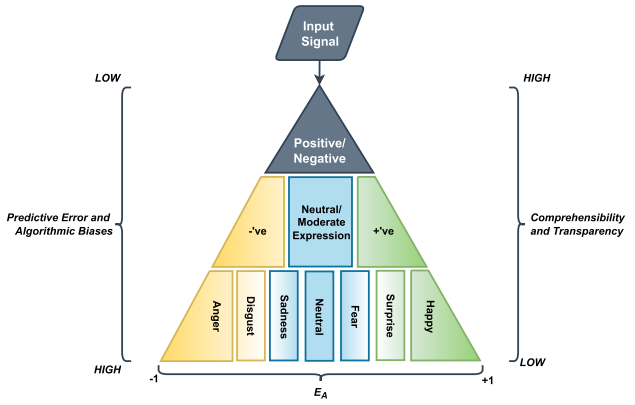


**FIGURE 1. The Algorithmic implementation of a typical affective state assessment (ASA) system showing the relationship between errors, biases and system transparency. The pyramid in this figure represents the overall system and provides insight into the interrelations between algorithmic biases, transparency and decision errors in ASA systems.**

how expressions and affective states change from 'negatively strong' to 'positively strong'. As obvious in Fig. 1, ASA systems offer either no or very little explainability to users and do not provide quantifiable measures of accuracy and accountability.

As shown in Fig. 2, a typical AI system would simply present results or inferences to users. As elements of comprehensibility, predictive accuracy and accountability are not available to users, the system behaves as a black box. The AXAI capability framework proposed and detailed in part one of this paper was designed to overcome these limitations. In order to open up the black box, the AXAI framework ensures that elements of comprehensibility, predictive accuracy and accountability are presented to users through a user-centred interface. As shown in Fig. 2, the black box of a typical AI system is transformed into a user-friendly graphical user interface showing elements of predictive accuracy, comprehensibility and accountability. Thus providing the AXAI-relevant information and explaining local ASA reasoning at each stage would eliminate the black box. Incorporating such a capability brings other advantages as well. For example, designers of an AXAI capable system, while adhering to XP practices, would readily notice any algorithmic biases and would have an opportunity to devise a suitable solution. The major benefits XP offers include an ease of software revision and update, ease of responding to changing requirements, inclusion of system users in the SDDP, and a formal ongoing mechanism for receiving feedback [30]. If designers fail to notice the problem during the initial design stages, users would discover them while inspecting the quantifiable measures provided by adding the AXAI capability in an ASA system. An illustrated comparison of the conventional ASA system with an AXAI-capable system in Fig. 2 shows how the two systems differ. The following sections describe the process of incorporating AXAI capabilities and shows how quantifiable measures of explainability, accuracy, and accountability can be added to an ASA system.

## III. THE ASAM DESIGN PROCESS FOR INCORPORATING THE AXAI CAPABILITY

User requirements keep changing during various stages of the SDDP viz., requirement analysis, design, implementation, and testing [31]. Faced with challenges pertaining to changing requirements, software developers in the mid 1990's started proposing ideas that later emerged as agile software design approaches. One of these approaches, XP, serves as an effective SDDP method [12]. Focusing on shorter processes and iterative progress, XP practices result in a flexible yet formal approach for dealing with a high rate of change in software requirements [30], [31]. This agile method emphasizes on the following four key points that brought fundamental changes to the software design and development process [31]:

1) Value individuals and interactions over processes and tools;
2) Value working software over comprehensive documentation;
3) Value customer collaboration over contract negotiation; and
4) Value responding to changes over following a plan.

We elaborate on the utility and suitability of these key points in the context of incorporating the AXAI capability in the ASAM and other AI systems' design and development.

1) Valuing individuals and interactions over processes and tools frees up AI system developers from such formal structures that ignore human engagement and rely on the built-in safeguards. This facilitates AXAI incorporation while avoiding an algorithm-centric view that would rely on 'developers' intuition of what constitutes a good explanation [7]. This results in more palatable, realistic explanations, by focusing on users who lack technical understanding of AI systems and engaging both users and AI system developers throughout the SDDP. Consequently, users are able to provide and developers are able to incorporate useful, domain-specific explanations. Continued interactions between users and system developers would help in embedding practitioner-oriented AXAI capabilities in AI systems.
2) Relying on working software ahead of comprehensive documentation would help in iteratively analysing, building and testing AI systems. A '*release-iteration*' cycle would help in providing immediate feedback and insight into the accuracy, comprehensibility and accountability aspects of the system and would reveal its ultimate strengths and weaknesses. This key point, combined with the first key point would let developers and users realize the domain-relevant utility and applicability of AI systems.
3) Valuing collaboration with customers over contract negotiations can help in avoiding unrealistic assumptions. This would assist in avoiding project changes that require ongoing adjustments and modifications throughout the SDDP. The ASAM SDDP benefitted from the collaborative work to iteratively build, test and improve the AXAI capabilities of the system.

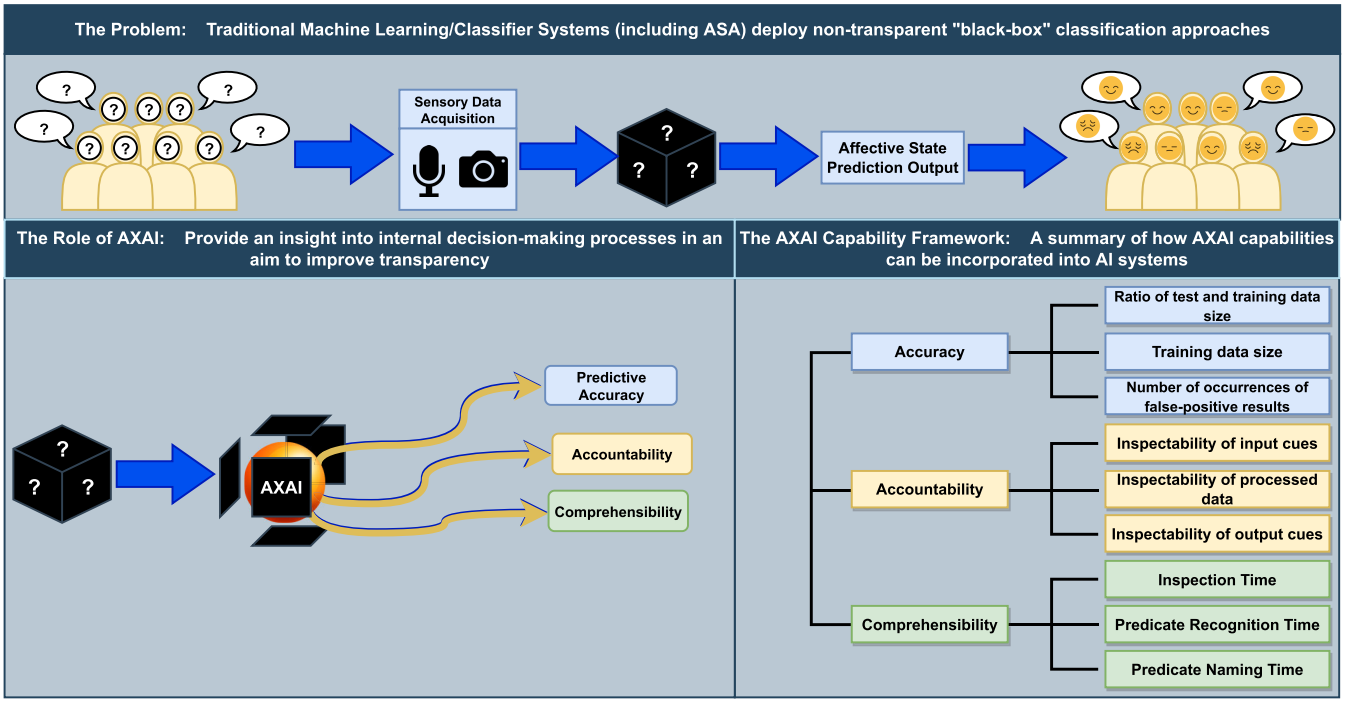


**FIGURE 2.** Top – General Graphical description of the black-box nature of machine learning systems. Information on decisions and reasons behind inferences is not available to users. Bottom left – An AXAI-capable ASA system transforms the black-box into a graphical user interface and shows elements of accountability, comprehensibility and predictive accuracy. Bottom right – Gallois-Lattices structure summarizing the AXAI-capability elements of the AXAI framework.

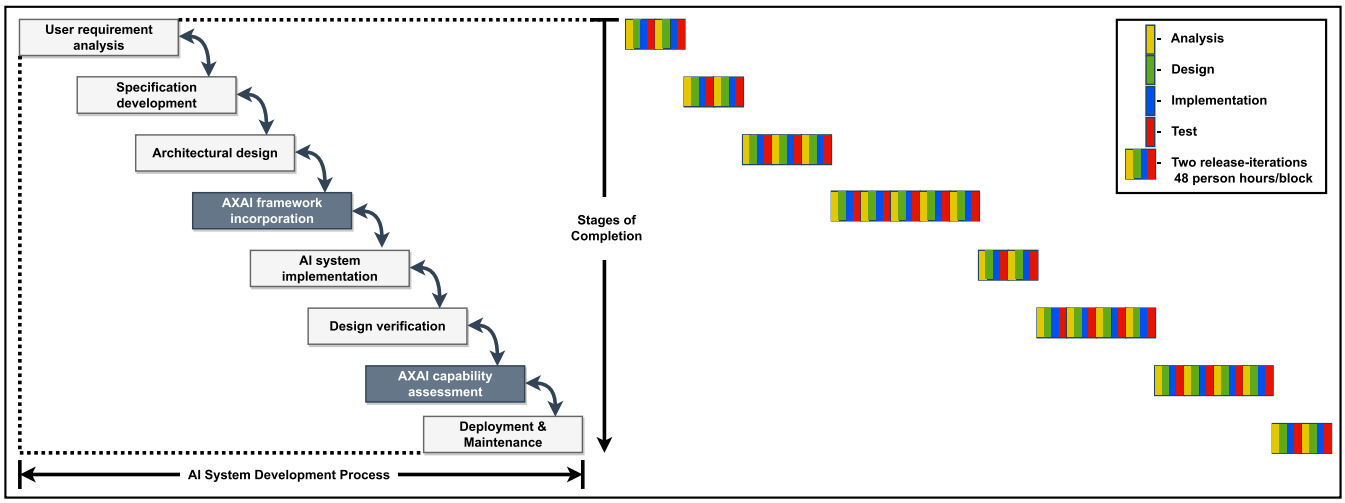


**FIGURE 3.** Mapping of the conventional AI system design and development process to the extreme programming practices. Each block of yellow, green, blue and red boxes represents a set of two release-iterations [12].

4) A logical implication of these key points was to respond to changes rather than follow a pre-planned scheme without abandoning the overall plan. Collaboratively and iteratively aligning a high-level plan with continuously emerging short-term strategies enabled the inclusion of changes that resulted from some new realizations that occurred between iterations.

Figure 3 explains how traditional AI system design processes could be laterally aligned to XP practices for incorporating AXAI capabilities in the ASAM. The conventional SDDP shown on the left-hand side of Fig. 3, illustrates the top-down iterative SDDP. This type of iterative process exposes built-in problems inherent in AI system development. Any flaws in system specifications would be noticed only during either the system implementation or the design verification stages. In order to overcome such problems, XP practices were adopted so that analysis, design, implementation and testing of the partially completed system could

be performed at each stage. Analysis, design, implementation and testing tasks are respectively shown in yellow, green, blue and red colours in Fig. 3. Each set of blocks represents four colours (two release-iterations). The duration of each block was approximately forty-eight person-hours.

The twenty-four sets, each having the four colour '*release-iterations*' blocks indicate the total person-hours ($24 \times 48 = 1152$ hours or twenty-nine forty-hour weeks) invested in completing the ASAM. As shown on the right hand side of Fig. 3, each of the user requirement analysis and specification development stages needed four iterations to complete. The architectural design stage took six release-iterations. The AI framework incorporation stage required ten release-iterations and the AXAI capability assessment stage required eight *release-iterations*. Having iterated at each stage, the development and maintenance stage required four iterations.

The term '*release*' includes a continuously prioritized set of AXAI capability and domain-related requirements that must be included in the AI system as it evolves from one partially completed stage to the next [33]. The term '*analysis*' in this discussion includes *stories* consisting of use cases suitable at each stage and for each '*release-iteration*' block [12]. As users would not be able to perceive all use cases at once, multiple release-iteration blocks at each stage allow them to progressively discover all important use cases.

Overall, employing XP practices resulted in a parsimonious and faster SDDP. The partially completed system weaknesses became readily visible at each stage. Problem rectification was therefore local, immediate and relevant to the SDDP stage. All major technical problems in implementing the system and any lack of domain-specific communication of explanations were locally discovered at each stage. Using the adopted SDDP, user perspective became visible from the developers' eye-level and developers' thoughts became clear to users. Each '*release-iteration*' block provided an opportunity to combine developer and user ideas, resulting in shared and mutually agreed goals. Extreme programming practices also allowed adhering to the aforementioned four key points and provided a structured approach for collaboratively working on the ASAM.

## IV. ITERATIVE ANALYSIS AND ASSESSMENT OF USER REQUIREMENTS

Some domain-specific user requirement analyses methods for AI systems were reported in the recent literature. For example, [34] examines user requirements in the context of data warehouse design, [35] attempts to determine requirements for interface design of a mobile location-based fair guide and [36] presents software requirement analysis and specification development processes for an intelligent system capable of monitoring and controlling smart phone user addiction. However, methods for ASA system-specific requirement analyses are not available in the literature. Hence, typical ASA systems use some *ad hoc* approach of dealing with user behaviour, expectations, and requirements.

We used both formal and user-driven requirement analysis approaches for incorporating AXAI capabilities into the ASAM. Formal approaches like those proposed in [37] were used to analyse the system level requirements. The domain-specific requirement analyses for the ASAM were based on user-driven methods such as the one proposed in [38]. We carried out the dynamic requirement analyses throughout the collaborative SDDP. Based on the two approaches, efforts were made to understand and model the user behaviour, system-user interaction requirements and application-specific implications of incorporating AXAI, as suggested in [38]–[40]. The modular structure of the ASAM, functional descriptions of various modules, and elements of the AXAI capability framework are visualised in Fig. 4.

During the process of SDDP, system developers and domain experts can progressively update and exchange information using a ''*hub-and-spoke*'' model. The ''*hub-and-spoke*'' model facilitates integration of loosely connected system requirements in a dynamic and ever changing SDDP [40]. We initially developed a high-level functional description of the ASAM that helped in performing user requirement analyses and developing specifications for each sub-module of the ASAM. The ''*hub-and-spoke*'' model was used to discover the link structure for determining the user requirements. For assessing the ASAM's functional behaviour vis-à-vis complying with XP practices, we developed use cases and tested various scenarios as elicited by agile design methodologies [39]. Use cases help in iteratively attaining the desired level of precision and refinement at each stage of system development. Figure 5 shows the two ''*Hub-and-spoke*'' models used for analysing system-level requirements and dynamic user interactions.

The following **six** examples show our SDDP, related analytical methods and demonstrate how use cases were envisaged and used during the developmental stages of the ASAM.

The internal management of affective state models, comparing the input data with the affective state models and labelling input data as one of the stored models are important functions of an ASA system. The system needs to maintain a list of all available affective state models that would be compared with an unknown incoming data sample before labelling it. During the process of comparing and matching, each affective state model, found different from the incoming data sample is removed from the list of available (matchable) models. Through this process, all differing models are removed from the list and an appropriately matching model is discovered. Through this process, an incoming data sample is assigned an expression label. Once a label is assigned, the list of available models is updated and all internally stored models are added to the list of matchable models before a new sample is received, compared, and labelled. A formal description of the '*compare-match-label*' process requires the following use cases:

1) The system should be able to maintain a list of affective state models. It must be able to remove and add the required number of models from the list.

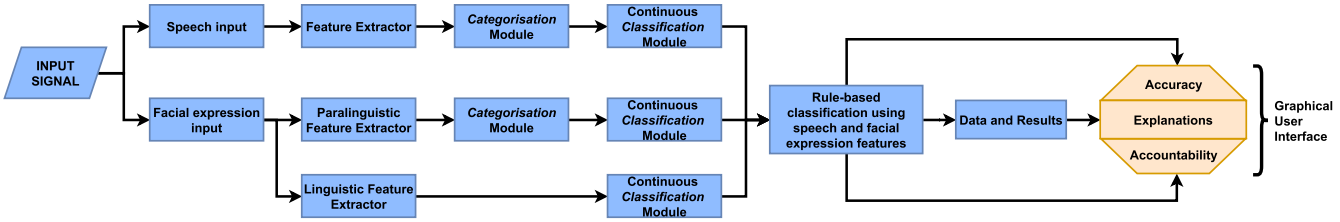


**FIGURE 4.** Functional modules and their interconnections within the ASAM. Notice how elements of accuracy, comprehensibility and accountability are embedded into the graphical user interface. This representation is expanded upon through Fig. 13.

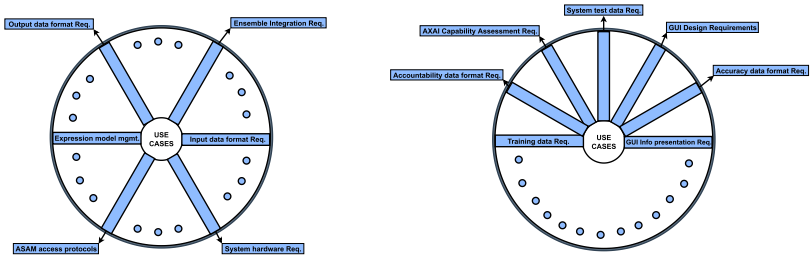


**FIGURE 5.** Parts of the ASAM's Hub-and Spoke model [34]. Left: System-level requirements. Right: The AXAI-specific requirements. Dots between spokes show additional spokes not shown in this figure. Use cases were used to describe the ASAM's functional features under varying conditions and stakeholders' requirements.

2) The system should be able to attach and record the label of a model that closely matches the incoming sample.
3) Once the '*compare-match-label*' process is complete, the system should be able to repeat the process.

The above listed use cases were analysed to produce a conceptual diagram for supporting the collaborative efforts and iteratively improving the system. A use case named MatchModel is shown in Fig. 6. It explains how an incoming sample **i** is assigned to an existing model **M** if a match **m** is found for labelling it as an expression **e**. For the ASAM to work effectively, models $M_j$, $M_k$, …, $M_n$ should not be available for matching with the sample **i** if the model $M_i$ matches the sample **i**. The bounded rationality phenomenon is widely used in combination with concepts borrowed from the fields of artificial intelligence and game theory [41]. Using the bounded rationality would help in framing all interactions between human users and the AI system as it would make the system navigation and human-system interactions extendable [41]. In the ASAM, interactions were planned such that each GUI field at level zero would provide the underlying construct either for comprehensibility, accountability, accuracy or a combination of these three feature vectors (see part one of this paper for more details). Each expandable field would attract more interaction to let users reach a deeper level of the GUI while seeking more information on explanations, accountability or accuracy [41]. Implementing this approach would require adequate descriptions for the use cases. We present a use case in Fig. 7 to elaborate on our implementation of the GUI.

An important aspect of the proposed AXAI capability was to let users explore partial functionality of the ASA system. This would allow users to inspect a particular

MatchModel $\triangleq$ **var** $M$:Model, $i$: sample;
Pre-condition : $M \in$ Model $\wedge$ $i \in$ sample
$\wedge$ IsMatcableTo (IsIncluded$^{-1}(M)$, $i$)
$\wedge$ IsAssignableTo ($M$, $i$)
Post-condition :$\exists e$: Expression $\bullet$ $e \notin$ Expression
$\wedge$ Expression$'$= Expression $\cup$ $\{e\}$
$\wedge$ Belongs$'$= Belongs $\cup$ $\{<e, M>\}$
$\wedge$ Matches$'$= Matches $\cup$ $\{<i, e, >\}$
$\wedge$ IsAssignableTo$'$= IsAssignableTo – $\cup_{i \in sample}\{< m, i >\}\{e\}$

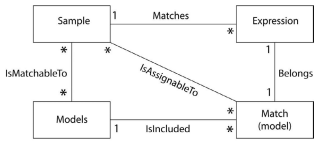


**FIGURE 6.** Left – Formal description of the MatchModel use case using the standard set notation. Right – Conceptual model of the MatchModel use case.

**Use case**: GUI display
**Actor:** The ASAM user requiring inspection of the input samples, processed data and the ASAM's inferences with output information
**Scope:** GUI providing the AXAI information
**Level:** GUI access
**Success Scenarios:**
1. The ASAM user needs to determine the AXAI capability of the ASAM by inspecting the input, processed and output information.
2. All fields of the GUI continuously update in real-time for providing dynamic data and analyses of the speech and visual cues.
3. The ASAM user has opportunities to examine all required/desired fields.
4. The ASAM user is able to inform the system that information at this level of the GUI has been inspected by clicking one of the several clickable options available at this level.
5. The ASAM GUI is updated such that each field becomes clickable after an interval of X seconds.

**Extensions:**
1a. The ASAM GUI module realizes that the system is at a deeper level of the GUI interface.
1a.1. The ASAM GUI module automatically reaches the GUI level zero after X seconds.
1a.2. The ASAM GUI module remains at level zero as long as the higher-level display option is inactivate and is not clicked.
1a.3. After arriving at a deeper level (level >0), the GUI module keeps displaying the relevant fields for Y seconds.
2a. At level zero, the ASAM GUI automatically updates all fields after every Z seconds.
2a.1. The ASAM's GUI module fields update function is inactive as long as the information at a deeper level (level>0) is being displayed.
2a.2. After arriving at a deeper level (level >0), the GUI module keeps displaying the relevant fields for Y seconds.

**FIGURE 7.** Formal description of the GUI display use case based on the standard set notation.

(or selected) input cue, relevant processed data and the inference/decision made by the ASAM. This would require sharing the input, processed and output data with users [11]. In some cases, users may also like to assess and ascertain AXAI capabilities on the basis of partial information and part-functionality. The use case illustrated in Fig. 8, shows how the facial expression data alone could be used for iterative assessment of part-functionality of the ASAM during

**Use case:** Partial information
**Goal:** The user needs to examine ASAM's decision using partial information.
**Actor:** User
**Scope:** Facial expression classification
**Level:** Partial functionality
**Precondition:** Appropriate visual data/information are available
**Trigger:** User opts to inspect the facial expression/ visual data/information above level zero.
**Precaution:** If the system is still analyzing data/information and a decision has yet to be made, the displayed information is able to reflect that.
**Success:** The visual input is correct. The facial expression data/ information is being analyzed. Input, processed, and output data fields are updated in real-time. The actor receives real-time sequence of facial expression images.

**Success Scenarios:**
1. The actor opts to inspect the facial expression data, analyses and relevant ASAM output.
2. The ASAM determines probabilities of input information assignment to various facial expression models.
3. The arrays of fields and data are regularly updated and presented to the actor.
4. The path of moving between various levels of data/information is pre-determined for actors.

Any update in arrays of fields and data at level zero cause updating of data/information at all levels.

**Extensions:**
1a. The actor is able to include the speech data/information in the review/s.
1a.1. The ASAM has provisions for inspecting visual data/information at multiple levels.
1a.2. The ASAM has provisions for inspecting speech data/information at multiple levels.
2a. The actor is able to freeze any frame of an input cue for examining the current decision.
3a. The actor can review data/information at each level for a prescribed and/or actor-determined amount of time.

**FIGURE 8.** Formal description of the partial information use case.

**Use case:** Comprehensibility
**Goal:** This use case ensures a user understands how data/information would lead the ASAM to make consistent inferences and label the sample data as belonging to one of the seven expressions of affective states.
**Actor:** The ASAM
**Scope:** Enable presentation, comprehension and future use of data/information, decisions and inferences in classifying and labeling the input data as expression of affective states. Implicit and explicit display of reasons behind input data classification and expression labeling
**Level:** Internal functionality
**Precondition:** Appropriate format of speech and visual data/information are fed to the ASAM
**Trigger:** Input speech and facial expression data/information are fed and received in real time. The user attempts to compare or mimic the ASAM capabilities after careful examination of the input speech and facial expression data/information, decisions and inferences.
**Precaution:** As long as the system is receiving, analyzing and producing data/information, decision and inference will keep changing. The displayed data fields and corresponding information must reflect that through the GUI.
**Success:** The actor receives appropriate speech and facial input data in real-time. The speech and facial input data are fed and received at a desired rate. All incoming data/ information can be analyzed. The input, processed and output data fields and the analytical information are periodically and regularly updated in real-time through charts and graphs. The system produces consistent information for similar cases.

**Success Scenarios:**
1. The actor is able to show and continuously update the input images.
2. The actor is able to plot, show and continuously update the paralinguistic (acoustic) feed as a waveform.
3. The actor is able to plot, show and continuously update the linguistic feed and the transcribed text in a legible form.
4. The actor is able to show and continuously update the inspectable facial expression data, analyzed information and the relevant ASAM output.
5. The actor is able to show, plot and continuously update the probabilities of the incoming facial expression information's' belonging to each of the seven expressions of affective states.
6. The actor is able to show, plot and continuously update the probabilities of the incoming paralinguistic information's' belonging to each of the seven expressions of affective states.
7. The actor is able to show, plot and continuously update the probabilities of the incoming linguistic feed's belonging to each of the seven expressions of affective states.

**Extensions:**
5a. The actor is able to show, plot and continuously update the dominance factor using the incoming facial expression information.
6a. The actor is able to show, plot and continuously update the dominance factor using the incoming paralinguistic information.
7a. The actor is able to show, plot and continuously update the dominance factor using the incoming linguistic feed.

**FIGURE 9.** Formal Description of the comprehensibility use case.

**Use case:** Accountability
**Goal:** This use case enables providing ample data and information for as long as a user would require them. It properly conveys decisions and inferences for a user to analyze and comprehend the displayed information. The user is able to confidently and continuously analyze or freeze the presented information. Similar data would consistently produce similar results.
**Actor:** The ASAM
**Scope:** Enable presentation, comprehension and future use of data/information, decisions and inferences in classifying and labeling the input data as expression of affective states.
**Level:** Internal functionality
**Precondition:** The GUI is able to share all speech and visual data/information fed to the ASAM vis-a-vis presenting ASAM's analytics, decisions and inferences.
**Trigger:** The user inspects the ASAM's capabilities. The user is able carefully examine the input speech and facial expression data/information, processed data, decisions and inferences.
**Precaution:** The user needs to be familiar with the domain and should possess a general understanding of AI systems.
**Success:** The actor is able to relate or differentiate between any set of received speech and facial input data and a plausible affective state expression in real-time.

**Success Scenarios:**
1. The user is able to freeze or unfreeze the GUI for a desired amount of time.
2. The actor is able to show and continuously update the input image sequence.
3. The actor is able to plot, show and continuously update the paralinguistic (acoustic) feed as a waveform.
4. The actor is able to plot, show and continuously update the linguistic feed and the transcribed text in a legible form.
5. The actor is able to show and continuously update the facial expression data, analyses and relevant ASAM output.
6. The actor is able to show, plot and continuously update the probabilities of the incoming facial expression information's belonging to each of the seven facial expressions.
7. The actor is able to show, plot and continuously update the probabilities of the incoming paralinguistic information's belonging to each of the seven expressions of affective states.
8. The actor is able to show, plot and continuously update the probabilities of the incoming linguistic feed's belonging to each of the seven expressions of affective states.
9. The user is able to freeze and unfreeze the GUI and all displayed fields at each level of the GUI.
10. The user is able to determine inconsistencies in the data within the displayed fields at each level of the GUI.

**Extensions:**
1a. The actor is able to navigate through the levels of the GUI whether the GUI is frozen or unfrozen.
2a. The user is able to freeze and unfreeze the dominance factor data using the incoming facial expression information.
3a. The actor is able to freeze and unfreeze the dominance factor data using the incoming paralinguistic information.
4a. The actor is able to freeze and unfreeze the dominance factor data using the incoming linguistic information.

**FIGURE 10.** Formal Description of the accountability use case.

**Use case:** Predictive Accuracy
**Goal:** This use case ensures the level of accuracy of the ASAM is assessable. The actor is able to display the elements of ASAM's accuracy.
**Actor:** The ASAM
**Scope:** Display ratio of tested samples and the samples used for training the ASAM. Display the number of instances of false-positive results. Display the absolute size of the samples used for training the ASAM.
**Level:** Internal functionality
**Precondition:** The actor has been commissioned and is in an active state.
**Trigger:** The actor is engaged with a user.
**Precaution:** The system is able to display this set of information in both frozen and unfrozen states via the GUI of the ASAM.
**Success:** The actor is able to display the stored information.

**Success Scenarios:**
1. The actor is able to display the stored information in real-time.
2. The actor is able to display the stored information while frozen.
3. The user is able to initiate information update session.
4. The user is able to terminate an information update session.

**Extensions:**
4a. The actor is able to store and display all relevant information after an update session is over.

**FIGURE 11.** Formal Description of the predictive accuracy use case.

the collaborative SDDP. A similar assessment was performed using speech data.

The use case presented in Fig. 9 illustrates how comprehensibility was incorporated in the ASAM's AXAI capabilities. This use case ensures that important aspects of comprehensibility are included in the system. In order to achieve the desired AXAI capabilities, comprehensibility was iteratively assessed and improved. Similar use cases were employed for incorporating accountability and accuracy in the ASAM. The use case presented in Fig. 10 illustrates how the elements of accountability would be incorporated in the ASAM while developing the system and assessing its AXAI capability. The use case presented in Fig. 11 illustrates how the predictive accuracy of the ASAM would be embedded for assessing ASAM's AXAI capability.

In order to iteratively and collaboratively assess system performance at each stage, we used a table for recording user motivation, user behaviour and user perceived value of the ASAM features. The table was based on the Gallois-Lattices structure [42] shown in Fig. 12. In our scheme, user motivation comprised of three elements: need, desire, and expectations. User perceived value also comprised of three elements: good, complicated and revise. Finally, user behavior had a set of three elements: use, revise and reject. All elements are measurable on a scale of $0 \rightarrow 1$. Each of the three main features (user motivation, user behaviour and user perceived value of the ASAM features) would help in determining and understanding user requirements. The main benefit of this scheme was that it provided a qualitatively separable, globally applicable, and easily quantifiable method of user requirement assessment. Furthermore, such an approach allows for faster, iterative and progressive refinement of each sub-system and its features.

As shown previously in Fig. 3 two collaborative iterations of user requirement analysis and assessment were sufficient

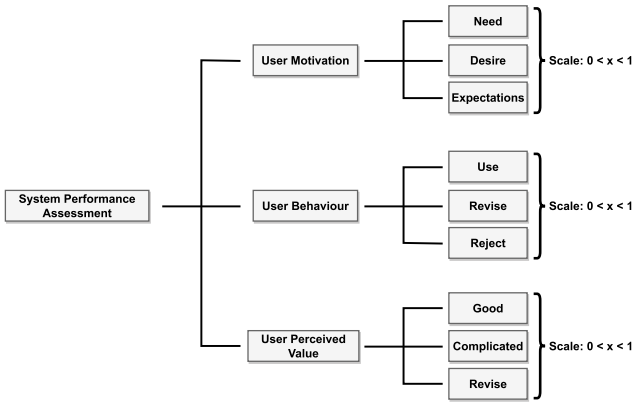


**FIGURE 12.** Gallois-Lattices structure modelling how system performances are assessed on the basis of user motivation, behaviour, and perceived value. These elements are monitored throughout the SDDP of an AI system including the ASAM discussed in this work.

for providing the needed insight into the desired features of the ASAM. It should be noted that users belonging to this assessment scheme need to be fully aware of the requirements and cognizant of the domain-specific details and system application scenarios.

## V. AN OVERVIEW OF THE ASAM

The ASAM treats affective states as dynamic representations of signals and uses an evolutionary approach for characterizing affective states in terms of human response tendencies. The ASAM deploys a novel mapping functionality that exploits continuous and discrete affective state expression models. Specifically, The ASAM exploits the continuous nature of the Plutchik spectrum [43] and the discrete nature of Ekman's discrete state theory [44]. Through the mapping functionality, the ASAM provides higher-level categorization followed by lower-level, discrete classification systems. As evident in Fig. 13, "Categorization" and "Classification" sub-modules facilitate this mapping functionality. The categorization sub-module draws from the position of discrete states within the continuous Plutchik spectrum [43]. Categorization in this case is based on a ternary classification scheme, which determines the high-level state that subsequently leads to the appropriate discrete state classifier i.e., the classification sub-module. These discrete affective state models draw on Ekman's discrete emotion theory arguing six basic emotions: happiness, sadness, anger, fear, disgust, and surprise [44]. The neutral state is often added in ASA systems to provide a benchmark for monitoring fluctuations from the norm. The Plutchik-Ekman mapping architecture visualized in Fig. 13 employs a two-tier Bayesian Classifier ensemble and is applied for both facial expression and paralinguistic speech signals. It can be modelled using the Bayesian formalism as:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)} \tag{1}$$

where $P(A)$ and $P(B)$ describe the prior and posterior probabilities for the categorization and classification sub-modules.

The classifier ensemble transforms a septenary classification schema into an ensemble of ternary and binary classifiers, making the system more explainable vis-à-vis reducing the ASAM's complexity and improving its predictive accuracy. The decision to transform the system was made through latter stages of the 'architectural design' stage of the SDDP, leading into the "AXAI framework incorporation" stage. Furthermore, as shown in Fig. 13, the same functionality was not applied to linguistic signals. This was decided after realizing how linguistic signals could be classified using a similarity approach instead of a dedicated classification machine.

Being a traditional embedded system, the ASAM was divided at the highest level into hardware and software components as visualized in Fig. 13. The hardware could be divided into monitoring devices, operational devices and sensory devices, with the latter usually contained in existing robotic systems. Likewise, monitoring devices such as displays/keyboards are ever-present across all industries in the modern age. Therefore, at its core, the ASAM would easily work with existing infrastructures, having a development board (with an on-board operating system), WIFI/Ethernet communication capabilities and a large-storage micro-SD card. The simplicity of the design allows for a cost-effective solution that would turn an existing robotic system into an AXAI-capable ASA system. In order to enable deployment of the ASAM into existing robotic systems, two development boards: 4G/64GB LattePanda and the NVIDIA Jetson Nano [43], [44] were tested during the deployment and maintenance stage of the SDDP, as shown earlier in Fig. 13. This decision was based on the idea that the ASAM should not be limited to support only personal computers. Rather, easily porting the ASAM to a modular device was required under the assumption that portable devices would have equal (or even greater) processing power than a personal computer. Throughout prototyping, a USB-powered web-camera was used to model the sensory device integration. An HDMI-connected external monitor and USB-connected mouse and keyboard were attached for monitoring capabilities and effective operation of the GUI. Wireless intra- and inter-machine communication allowed for detection of smart devices on the same network vis-à-vis allowing for cloud storage of observed data.

The sensory devices were carefully considered for achieving the desired overall functionality of the system. Once the ASAM is equipped with the required peripheral devices and these devices are detected, the ASAM would extract the necessary features from the input cues. Specifically, the sensory devices would detect facial expressions and speech inputs, the latter would be quickly divided into paralinguistic and linguistic cues. Classification of affective state expressions using all input cues would involve pre-processing and feature extraction. Regarding the split of speech, paralinguistics would describe "how" speech sounds based on fluctuations

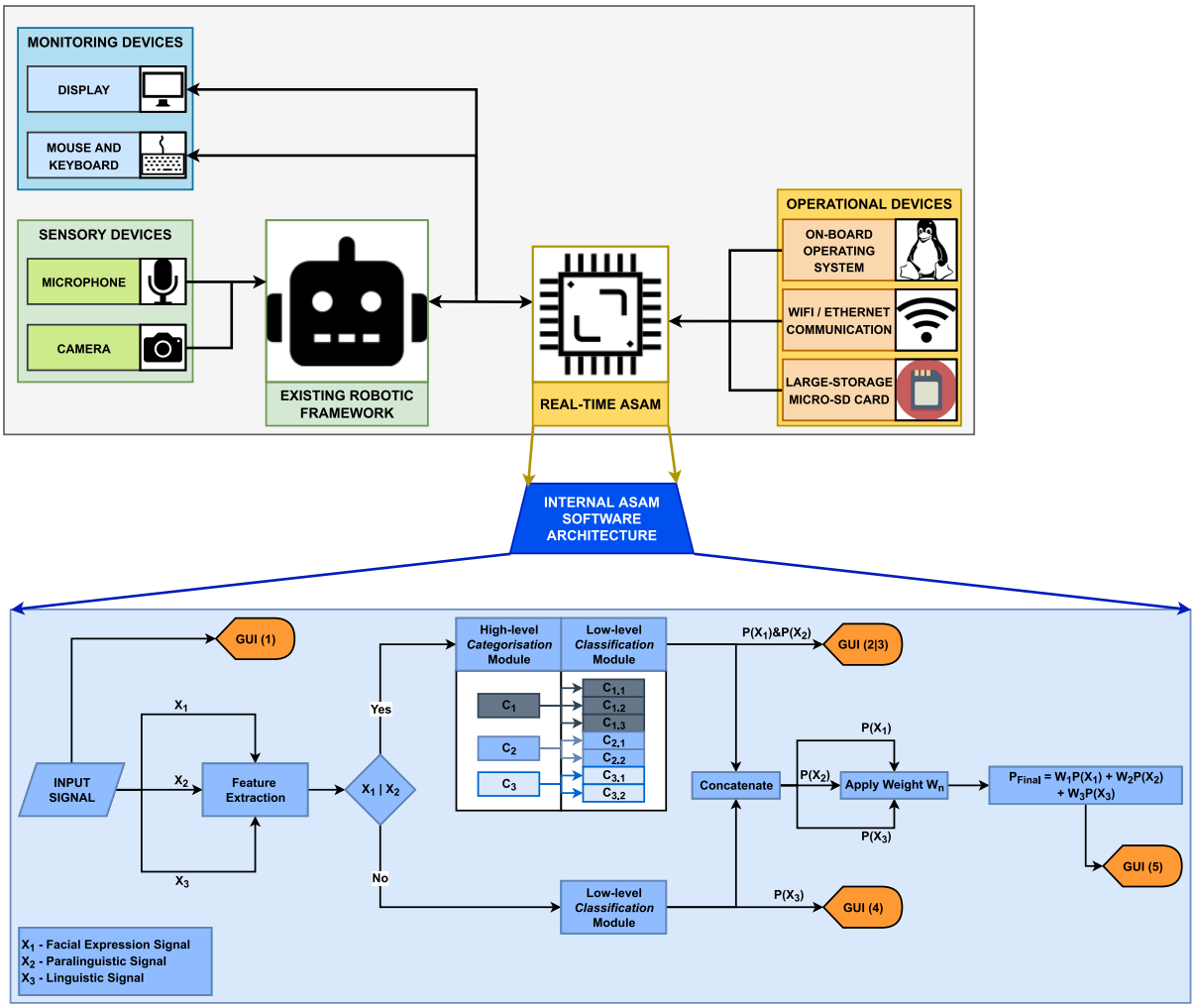


**FIGURE 13.** ASAM's high-level embedded hardware and functional layout showing its intuitive integration into existing robotic frameworks. Operational, sensory, and monitoring devices are integrated to ensure ASAM's full functionality. (BOTTOM) Internal software architecture of the ASAM showing how input and output signals are manipulated through the system and relayed to the user via the Graphical User Interface. The Bayesian classifier ensemble is also shown through "Categorization" and "Classification" sub-modules. Outputs to the GUI windows (1-5) exhibit the AXAI capabilities of the system highlighting how information is disseminated to the users at various stages of the ASA process.

in acoustic features [45], [46]. The use of linguistics in the ASAM focuses on describing the content of ''what was said,'' and represents the structure of words in an utterance. The conscious and subconscious manipulation of linguistic and paralinguistic features augment each other, such that humans can manipulate the way we express ourselves through speech. The ASAM deploys a speech-to-text function to separate the linguistic features from the raw speech input used for the paralinguistic assessment of affective states.

The Facial Action Coding System (FACS) and the Emotional Facial Action Coding System (EM-FACS) [47], [48] are widely used in facial expression recognition and ASA. They allow determining what facial muscles are activated using ''Action Units'' which are combined with discrete expressions of affective states. This approach has been used for developing several facial affective state expression datasets such as the Extended Cohn-Kanade (CK+) dataset [51] used in this work for training and initial validation of the facial expression classification models. Feature extraction was performed using variations of the InceptionV3 Convolutional Neural Network (CNN) architecture described by [52]. The construction of the ASAM's categorization and classification models w.r.t facial expression assessment are outlined in Table 1. Paralinguistic expressions of affective states within the ASAM are fluctuations in acoustic features. Modelling these feature fluctuations was necessary for developing suitable supervised models of affective expressions. Outlining these features was meant to assists users in understanding how the ASAM would detect an affective state in speech by realizing ''how'' something was said. There are many combinations of features and classifiers used for detecting emotion in speech as discussed by [47] and [53]. After collaborative and progressive review during the SDDP and through iterations in the ''AI system

**TABLE 1. Facial expression Bayesian classifier ensemble parameters used for training the ASAM with the CK+ dataset (using Python and Keras/TensorFlow packages).**

| | Categorization Model (Valence) | Classifier Model 1 (Positive-High Valence) | Classifier Model 2 (Low Valence) | Classifier Model 3 (Negative-High Valence) |
|---|---|---|---|---|
| Input Shape | (100, 100, 1) | (150, 150, 1) | (150, 150, 1) | (100, 100, 1) |
| Parameters | 3,482,787 | 9,238,719 | 9,238,711 | 10,677,410 |
| Epochs | 50 | 100 | 200 | 100 |
| InceptionV3 Output Layer | Mixed4 | Mixed8 | Mixed8 | Mixed7 |
| Output Classes | Positive-High, Low, Negative-High | happy, fear, surprise | neutral, sad | anger, disgust |
| Validation Loss (CK+) | 0.0708 | 0.4887 | 0.6959 | 0.1471 |
| Validation Accuracy (CK+) | 98.57% | 87.10% | 80.00% | 95.24% |

**TABLE 2. Paralinguistic Bayesian classifier ensemble parameters used for training the ASAM with the TESS dataset (executed through Python and SKLearn packages).**

| | Categorization Model (Valence) | Classifier Model 1 (Positive-High Valence) | Classifier Model 2 (Low Valence) | Classifier Model 3 (Negative-High Valence) |
|---|---|---|---|---|
| Input Shape | (24, 1) | (24, 1) | (24, 1) | (24, 1) |
| Gamma ($\gamma$) | $3.5e^{-6}$ | 2 | 100 | 100 |
| Cost-function Parameter (C) | 1 | 1 | 0.01 | 0.01 |
| Kernel Type | Sigmoid | Linear | Linear | Sigmoid |
| Output Classes | Positive-High, Low, Negative-High | happy, fear, surprise | neutral, sad | anger, disgust |
| Validation Accuracy (TESS) | 87.71% | 98.83% | 100% | 100% |

**TABLE 3. ASAM statistical performance metrics when validated on the RAVDESS dataset for facial expression and paralinguistic classifiers (using model parameters discussed prior).**

| | Facial Classifiers | | Paralinguistic Classifiers | |
|---|---|---|---|---|
| | Categorization Module | Classification Module | Categorization Module | Classification Module |
| Observed Accuracy | 0.948 | 0.773 | 0.985 | 0.811 |
| Average Precision | 0.941 | 0.773 | 0.980 | 0.791 |
| Average Recall | 0.954 | 0.784 | 0.987 | 0.838 |
| Cohen's Kappa | 0.920 | 0.760 | 0.975 | 0.802 |
| $F_1$ Score | 0.946 | 0.768 | 0.983 | 0.797 |

Implementation'' stage, we decided on the following paralinguistic features for extraction: (i) Acoustic Power, (ii) Root Mean Square Energy, (iii) Vocal stress – represented through Teager-Energy Operator (TEO) coefficients, (iv) Mean Spectral Centroid Frequency and (v) Mel-Frequency Cepstral Coefficients (MFCC). These features have been used in several speech recognition systems, though not as optimal sets of voice parameters for discriminating between emotive states. However, our use of spectral factors was based on the fact that valuable paralinguistic information could be derived from prodosic and spectral features [47]. The aforementioned paralinguistic features resulted in a 24-dimensional feature array, with categorization and classification sub-modules using Support-Vector Machine (SVM) models. The Toronto Emotional Speech Set (TESS) [54] was used for training and initial validation of the paralinguistic speech models. The specifications of the classifier ensemble are discussed in Table 2, showing Gamma and Cost-function parameters assisting in construction of the SVM hyperplane boundaries.

Linguistic expressions of affective states were determined using the structure of ''what was spoken'' in an utterance. Therefore, the text string provided by the speech-to-text transcription process laid the foundations for feature extraction and classification in this subsystem. The two affective state expression and classification theories that were applicable within the context of the ASAM (Ekman's discrete state theory and Plutchik's continuous spectrum-based state theory) were exploited in this work [41], [42]. Another example of a continuous model is Russell and Mehrabian's Three-Factor Theory of Emotions [55] model, which represents states in a 3-dimensional pleasure/valence, arousal, and dominance (VAD) space. Researchers have developed ''Emotional Lexicons'' which represent words across various languages including English through normalized VAD values, which allows for the modelling and classification of linguistic speech features as evident in the ASAM.

The ASAM employs the Canadian National Research Council (NRC) Lexicon [56] and a similarity approach to determine speakers' affective states in real-time using the VAD parameters of the input transcription and mapping the parameter values to the VAD values for each discrete affective state as per the NRC Lexicon. Mapping the 3D Euclidean distance (obtained from the input transcription) to each affective state allowed determining which affective state is close to the input linguistic signal for classification. The algorithmic implementation of the linguistic classifier is summarized hereunder:

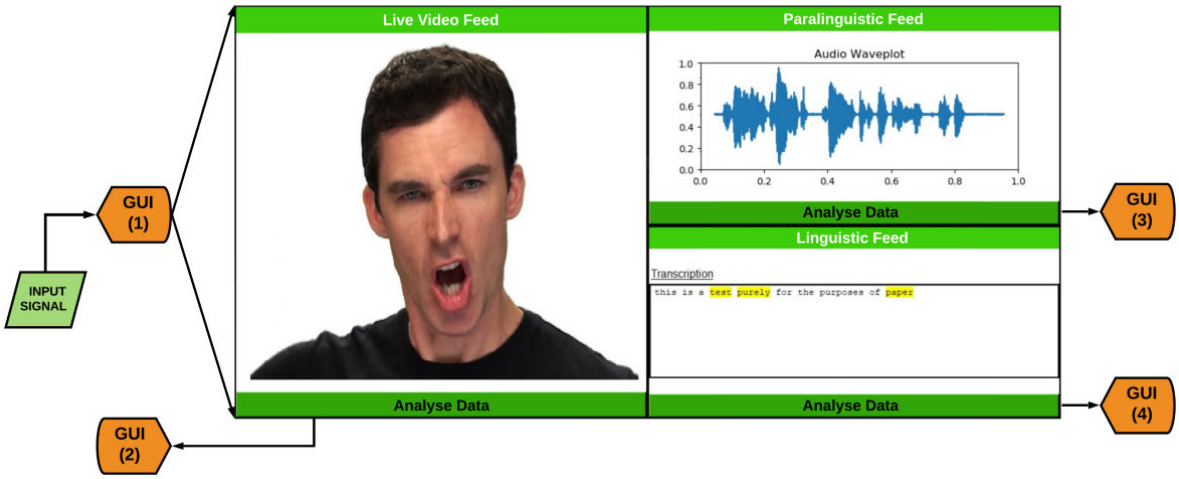


**FIGURE 14.** Graphical User Interface window – GUI(1), which is shown to users upon execution of the ASAM software. The placeholder image used for the live video feed is extracted from the RAVDESS dataset [57]. The Transcription box contains the string "this is a test purely for the purposes of paper" with the highlighted words being {test, purely, paper}.

1) Transcribe the input audio signal in real-time.
2) Scan the transcription for matched words as per the NRC Lexicon and store the VAD values for each matched word.
3) For '$N$' matched words in a transcription, calculate the mean VAD values, i.e., $V_{ave}, A_{ave}, D_{ave}$:

$$X_{ave} = \frac{\sum_{n=1}^{N} X_n}{N}, \tag{2}$$

4) For each affective state class '$c$', determine the 3D Euclidean distance $\Delta AB_c$ between the input utterance and $c$, modelled by:

$$\Delta AB_c = \sqrt{(V_c - V_{ave})^2 + (A_c - A_{ave})^2 + (D_c - D_{ave})^2} \tag{3}$$

5) Determine the closest affective state based on the relative position of the transcribed input utterance within the VAD space, i.e.: the distance-based similarity '$DBS$' for each state '$c$':

$$DBS_c = \frac{1}{1 + \Delta AB_c} \tag{4}$$

The ASAM was further validated on the Ryerson AudioVisual Database of Emotional Speech and Song (RAVDESS) [57], a multimodal dataset that was used for validating the system with foreign, 'wild' data unknown at the time of training. The facial expression classifiers, described above, were validated using twenty thousand (20000) active frames of data corresponding to approximately thirteen (13) minutes of speech data. Due to the construction of the linguistic classifier and the nature of the RAVDESS dataset (all utterances were not emotionally charged), a similar validation could not be performed for this work.

Table 3 summarizes each classifier's performance when validated on the RAVDESS dataset. Using this information and understanding the ASAM, we could determine the predictive accuracy $P_A$ of the ASAM for estimating the AXAI capability of the ASAM as discussed in part one of this paper. To summarize, we concluded that the normalized scores for the $P_A$ components were as follows: $r_{tst-trn} = 1.0$, $d_{trn} = 0.811$ and $O_{fp} = 0.8559$, resulting in a predictive accuracy score of: $P_A = 1.546$ (see more details in Section 5 of Part one of this paper). Details pertaining to predictive accuracy within the context of the ASAM were given in use case 6 (Fig. 11).

Assessing affective states through the three independent channels - individually, discounts the multimodal nature of human affective state expression patterns and fails to provide a holistic representation of human intent and affective experience. Recent works [56], [57] also suggest so in the context of ASA systems, reckoning benefits of multimodal models over their unimodal counterparts. The ASAM is a multimodal assessment system, as shown in Fig. 13. The input signals are combined in a final classifier ensemble – a weighted-sum rule-fusion system, which is derived after [60]. Each sub-classifier output (facial expression, paralinguistic, linguistic) combines as having weights as such (for each discrete state):

$$P_{Final} = W_1 P_{FACE} + W_2 P_{PARA} + W_3 DBS \tag{5}$$

where '$P_{Final}$' is the final prediction accuracy, '$P_{FACE}$' is the facial expression prediction accuracy, '$P_{PARA}$' is the paralinguistic prediction accuracy and '$DBS$' is the linguistic distance-based similarity output. $W_n$ defines the hard-coded weight applied to the $n^{th}$ signal. In the ASAM, these weights are based on theoretical foundations proposed by [61] stating that through conversation, facial expressions and body language constitute for 55% ($W_1 = 0.55$) of the total expression, paralinguistics constitute 38% ($W_2 = 0.38$) and linguistics constitute the final 7% ($W_3 = 0.07$), thus resulting in the

system being modelled as:

$$P_{Final} = 0.55P_{FACE} + 0.38P_{PARA} + 0.07DBS \quad (6)$$

The transparency of the classifier weights is output to the user to ensure that accountability capabilities '$S_A$' adhere to the AXAI capability framework guidelines, one of the three components that are used to determine the explainability of an AI system (discussed in Part one of this paper). Transparency is also achieved through the display of classifier 'confidence' and 'dominance' values, with the former describing how ''confident'' the classifier is about a particular state being expressed and the latter describing how long a particular affective experience is expressed within a particular period of time. Hence, Dominance $= \frac{n_{P_i}}{N_T} \times 100\%$, where '$n_{P_i}$' refers to the number of predictions made for the $i^{th}$ state and $N_T$ is the total number of predictions made during a period of time '$T$'.

The ASAM's integrated GUI enables simultaneous monitoring of all input data feeds and the fifty-six (56) unique and corresponding outputs. The ASAM becomes AXAI-capable by transparently sharing the classification and decision-making processes (thus improving accountability) and optimized display of the GUI elements for incorporating system comprehensibility. These elements were designed, tested, improved and implemented through various iterations during the SDDP for ensuring the AXAI capabilities in the ASAM. The iterative and collaborative SDDP also helped in exploiting the best of the SDDP team abilities. The ASAM GUI consists of five unique windows each contributing towards different areas of system accountability and comprehensibility. The orange output blocks in Fig. 13 define the flow of information from the back-end to the front-end. Figures 14 and 15 highlight these GUI windows in greater details. Figure 14 (GUI window (1)) is important when considering the inspectability of input signals '$I_{in}$' given that all inputs are visible to the user upon execution of the program. As expressed previously, the transparency of classifier weights is also important for system accountability specifically in regard to the inspectability of the processed data '$I_{pro}$'. Furthermore, given the ASAM is a multimodal system, GUI windows (2-4) assist this aspect giving the processed data accounts for the final weighted-sum rule-fusion output. However, as expressed in Part one of this paper, users suggested that this could be improved by introducing more information about how each individual signal was processed – specifically when considering that sometimes a user may only be concerned with one signal and not the multimodal output. Finally, the inspectability of output cues '$I_{out}$' is evident across all output GUI windows (2-5). This was achieved through the graphical and tabular representations of data and enabling users to focus on any affective state that would be necessary for a particular application. This was a design decision that was made in the 'AXAI capability assessment' stage of the SDDP. It was observed that an over-abundance of information would not benefit the AI system design in becoming more transparent and explainable [62]. More information pertaining to the accountability within the context of the ASAM was given in use case 5, visualized in Fig. 10.

Regarding comprehensibility components, the predicate naming time '$T_{pn}$' is improved by displaying predicted class labels as affective state names rather than outputting back-end class indices {0, 1, 2, 3, 4, 5, 6}. Showing a gradient from white-to-green in the tabular output highlights how confident the system is of a particular state being expressed. This feature attracted user attention to the predicted output as affective states are everchanging, effecting the predicate recognition time '$T_{pr}$'. The predicate inspection time '$T_i$' was enhanced by separating classifiers into four different windows, allowing users to isolate and inspect the information that was of concern to them without being burdened by information which they may have deemed ''not useful.'' More information pertaining to comprehensibility within the context of the ASAM was also introduced in use-case 4, visualized in Fig. 9.

Overall, this section has highlighted the AXAI capabilities of the ASAM and how these elements were added to the system in an iterative manner during the SDDP of the ASAM. Furthermore, through elements of the GUI windows and the system architecture we have highlighted how the ASAM would adhere to XP principles. Overall, we have demonstrated the iterative SDDP through several cases and application of the AXAI theory.

## VI. DISCUSSION

Most AI and ML systems behave like black boxes as they fail to explain their decisions [11]. Incorporation of XAI in AI and ML systems requires attention to four system features: (i) the quality of inputs and interactions between them, (ii) method of combining the input information, (iii) the quality of the training data and, (iv) levels of trust users put in system decisions [11]. Incorporation of the proposed AXAI capability framework also exploited and relied on these four features. As discussed earlier, the ASAM's GUI continually presents the aforementioned features to users and lets them develop confidence in the system's AXAI capabilities.

The GUI provides information on the quality of input data in real-time and continuously updates data fields vis-à-vis displaying the currently dominant affective state. If multiple affective states score high percentages (shown to users through the dominance plots), users would be automatically alarmed about inconsistencies in the input information. The GUI is also able to show interactions between various inputs through the real-time data display and regular updates of visual, paralinguistic and linguistic data. Furthermore, displays of explicit and implicit elements of accountability, comprehensibility, and predictive accuracy help users in establishing the corresponding level of trust in the ASAM. This would also help in establishing a chain of responsibility.

An appropriate level of AXAI incorporation in the ASAM suggests that XP practices support an iterative and collaborative AI software design & development process (SSDP). This work shows that an iterative and collaborative AI SDDP

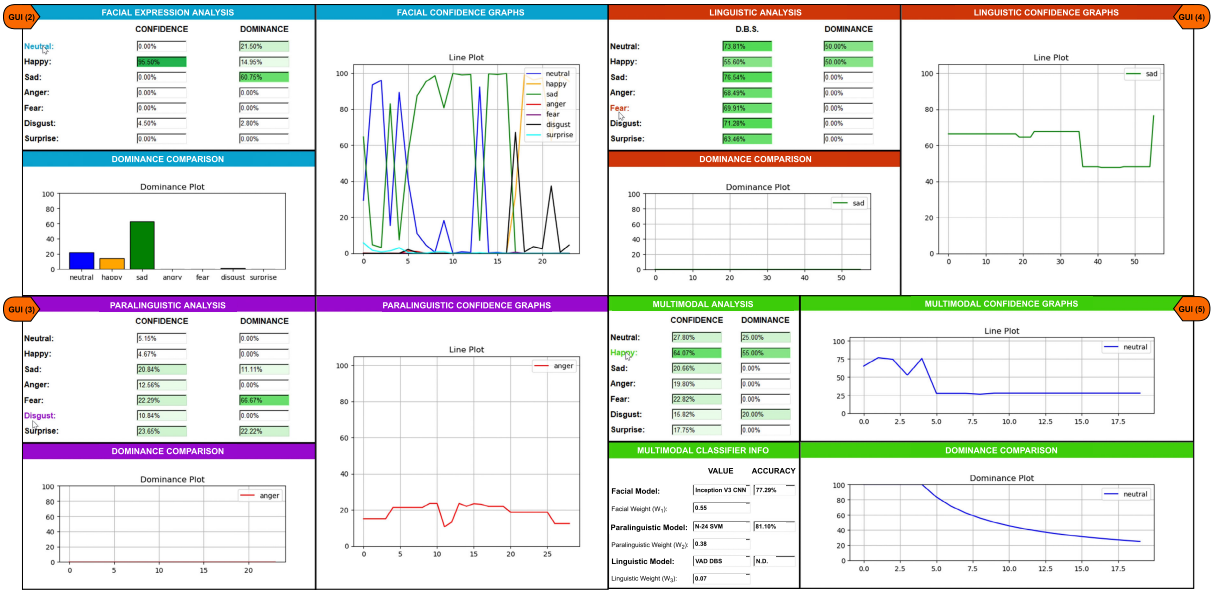


**FIGURE 15. Overview of the multimodal ASA monitoring windows embedded in the ASAM. These windows are accessible through their corresponding "Analyse Data" buttons as shown in Fig. 14. The Multimodal Analysis window – GUI (5) is opened in parallel to the GUI (1) window shown in Fig. 14. Tetradic colour theory was chosen for GUI window design where: CYAN = Facial Expression Analysis, ORANGE = Linguistic Analysis, PURPLE = Paralinguistic Analysis and LIME.**

would allow developers to analyze and understand user behavior when building a domain-specific AI application. However, the proposed AXAI capability framework remains largely domain-independent. The demonstrated AXAI implementation would hopefully initiate further investigations on developing methods and norms for domain-agnostic and application-independent AXAI capabilities.

As the proposed AXAI capability framework was built upon previous works, this work should be considered as a step forward in the direction of developing better ML and AI systems having accountability, comprehensibility, and accuracy built into them.

## VII. CONCLUSION

Researchers were able to realize the need for incorporating explanations in AI and ML systems in the 1970's. Nonetheless, a good number of recently published papers in domains like medical diagnosis, cognition, psychiatric and psychological sciences, law and criminal investigations and image understanding highlight that systems still lack explanations. The literature also mentions significance of incorporating AXAI capabilities in AI systems.

The part one of this paper introduced the AXAI capability framework. This paper detailed incorporation of the AXAI capabilities in an affective state assessment system. Hence, this work serves as a tutorial and provides specific information on various functional modules and link structures required to connect users with use cases. Examples of use cases that facilitated incorporating AXAI capabilities in the system are also given. Though the domain of our application was affective computing, this paper introduced a domain-agnostic and portable methodology for incorporating the AXAI capability in ML and AI systems.

So far, little work has been done on selecting an appropriate SDDP for building AI systems. Consequently, current literature offers very little information on application of agile software design practices in AI system design. Usually, *ad hoc* practices are used for designing and implementing AI and ML systems. This work provided insight into suitability of agile software methods for AI system design and showed how eXtreme Programming (XP) practices would help in incorporating the AXAI capability. We demonstrated systematic, localized and iterative development of an AI system. We showed that an appropriate SDDP would help in collaboratively discovering user requirements and understanding user behaviour. We also exhibited that use of conceptual models and use cases would augment progressive system enhancement and allow identifying any shortcomings in functional capabilities of an AI system.

During the ASAM design, user motivation was observed in terms of need, desire and expectations. The user behaviour was determined through either use, revision or rejection of a function. The user-perceived value of features was established as either good, complicated or revisable. These parameters of user motivation ware helpful in determining the system requirements. As discussions on user motivation, user behaviour and user perceived value of an AI system are not common in the affective computing literature, it would be safe to assume that we contributed to the prevailing knowledge on affective system design. This work also demonstrated that the user-developer engagement would make the SDDP sensitive to ethical, functional, technical and domain-specific

implications of a contemporary AI system. This demonstration is anticipated to promote the idea of engaging system developers and users during the AI system design process. However, the success of our proposed AXAI capability framework and its implementation would depend on the quality of input data, level and nature of interactions between input cues, mechanisms of combining the input and processed data, and the magnitude and quality of the training data. Without paying attention to these factors, building trust in an AI system's decisions would not be possible. In order for users to understand the system inferences, design of a suitable GUI and dynamic display of all pertinent data would be necessary as well.

The proposed AXAI framework and its incorporation in the ASAM are anticipated to initiate investigations on various aspects of building, assessing and incorporating AXAI capabilities in affective computing systems. We also expect to motivate further investigations in social, ethical, legal and cognitive implications of designing, assessing and incorporating AXAI capabilities.

## REFERENCES

[1] B. Kim and F. Doshi-Velez, "Machine learning techniques for accountability," *AI Mag.*, vol. 42, no. 1, pp. 47–52, Apr. 2021.
[2] D. Michie, "Machine learning in the next five years," in *Proc. 3rd Eur. Conf. Eur. Work. Session Learn.*, Glasgow, U.K., 1988, pp. 107–122.
[3] U. Schmid, C. Zeller, T. Besold, A. Tamaddoni-Nezhad, and S. Muggleton, "How does predicate invention affect human comprehensibility?" in *Proc. 26th Int. Conf. Inductive Logic Program.*, London, U.K., 2017, pp. 52–67.
[4] T. Miller, "Explanation in artificial intelligence: Insights from the social sciences," *Artif. Intell.*, vol. 267, pp. 1–38, Feb. 2019.
[5] M.-A.-T. Vu, T. Adali, D. Ba, G. Buzsáki, D. Carlson, K. Heller, C. Liston, C. Rudin, V. S. Sohal, A. S. Widge, H. S. Mayberg, G. Sapiro, and K. Dzirasa, "A shared vision for machine learning in neuroscience," *J. Neurosci.*, vol. 38, no. 7, pp. 1601–1607, Feb. 2018.
[6] U. Pawar, D. O'Shea, S. Rea, and R. O'Reilly, "Explainable AI in healthcare," in *Proc. Int. Conf. Cyber Situational Awareness, Data Anal. Assessment (CyberSA)*, Dublin, Ireland, Jun. 2020.
[7] Q. V. Liao, D. Gruen, and S. Miller, "Questioning the AI: Informing design practices for explainable AI user experiences," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Honolulu, HI, USA, Apr. 2020, pp. 1–15.
[8] D. Gunning and D. Aha, "DARPA's explainable artificial intelligence (XAI) program," *AI Mag.*, vol. 40, no. 2, pp. 44–58, Jun. 2019.
[9] L. Longo, R. Goebel, F. Lecue, P. Kieseberg, and A. Holzinger, "Explainable artificial intelligence: Concepts, applications, research challenges and visions," in *Machine Learning and Knowledge Extraction*, A. Holzinger, P. Kieseberg, T. A. Min and E. Weippl, Eds. Cham, Switzerland: Springer, 2020, pp. 1–16.
[10] L. Ai, S. H. Muggleton, C. Hocquette, M. Gromowski, and U. Schmid, "Beneficial and harmful explanatory machine learning," *Mach. Learn.*, vol. 110, no. 4, pp. 695–721, Mar. 2021.
[11] B. J. Murray, M. A. Islam, A. J. Pinar, D. T. Anderson, G. J. Scott, T. C. Havens, and J. M. Keller, "Explainable AI for the choquet integral," *IEEE Trans. Emerg. Topics Comput. Intell.*, vol. 5, no. 4, pp. 520–529, Aug. 2021.
[12] K. Beck, "Embracing change with extreme programming," *IEEE Comput.*, vol. 32, no. 10, pp. 70–77, Oct. 1999.
[13] S. Nayak, V. Sharma, K. Panda, and S. Uttarkabat, "Affective state analysis through visual and thermal image sequences," in *Emerging Technologies in Data Mining and Information Security*. Singapore: Springer, 2021, pp. 65–73.
[14] D. Benson, M. M. Khan, T. Tan, and T. Hargreaves, "Modeling and verification of facial expression display mechanism for developing a sociable robot face," in *Proc. Int. Conf. Adv. Robot. Mechatronics (ICARM)*, Macau, China, Aug. 2016, pp. 76–81.
[15] T. Hargreaves, M. M. Khan, D. Benson, and T. Tan, "Closed-loop Petri net model for implementing an affective-state expressive robotic face," in *Proc. IEEE Int. Conf. Adv. Intell. Mechatronics (AIM)*, Banff, AB, Canada, Jul. 2016, pp. 463–467.
[16] D. Cernea and A. Kerren, "A survey of technologies on the rise for emotion-enhanced interaction," *J. Vis. Lang. Comput.*, vol. 31, pp. 70–86, Dec. 2015.
[17] M. M. Khan, "Cluster analytic detection of disgust-arousal," in *Proc. 9th Int. Conf. Intell. Syst. Design Appl.*, Pisa, Italy, 2009, pp. 641–647.
[18] M. M. Khan, R. D. Ward, and M. Ingleby, "Toward use of facial thermal features in dynamic assessment of affect and arousal level," *IEEE Trans. Affect. Comput.*, vol. 8, no. 3, pp. 412–425, Jul. 2017.
[19] S. K. D'mello and J. Kory, "A review and meta-analysis of multimodal affect detection systems," *ACM Comput. Surv.*, vol. 47, no. 3, pp. 1–36, Apr. 2015.
[20] C. E. Izard, *The Psychology of Emotions*. New York, NY, USA: Springer, 1991.
[21] K. Lochner, "Affect mood and emotions," in *Successful Emotions*. Wiesbaden, Germany: Springer, 2016, pp. 43–69.
[22] D. Watson, L. A. Clark, and A. Tellegen, "Development and validation of brief measures of positive and negative affect: The PANAS scales," *J. Personality Social Psychol.*, vol. 54, no. 6, pp. 1063–1070, 1988.
[23] M. M. Bradley and P. J. Lang, "Measuring emotion: The self-assessment manikin and the semantic differential," *J. Behav. Therapy Exp. Psychiatry*, vol. 25, no. 1, pp. 49–59, 1994.
[24] R. Steyer, P. Schwenkmezger, P. Notz, and M. Eid, "Testtheoretische analysen des mehrdimensionalen befindlichkeitsfragebogen (MDBF)," *Diagnostica*, vol. 40, no. 4, pp. 320–328, 1997.
[25] A. M. Ruef and R. W. Levenson, "Continuous measurement of emotion," in *Handbook of Emotion Elicitation and Assessment*, J. A. Coan and J. J. Allen, Eds. Oxford, U.K.: Oxford Univ. Press, 2007, pp. 286–297.
[26] J. Vice, M. M. Khan, and S. Yanushkevich, "Multimodal models for contextual affect assessment in real-time," in *Proc. IEEE 1st Int. Conf. Cognit. Mach. Intell. (CogMI)*, Los Angeles, CA, USA, Dec. 2019, pp. 87–92.
[27] Y. Zhao, X. Cao, J. Lin, D. Yu, and X. Cao, "Multimodal affective states recognition based on multiscale cnns and biologically inspired decision fusion model," *IEEE Trans. Affect. Comput.*, early access, Jul. 1, 2021, doi: 10.1109/TAFFC.2021.3093923.
[28] K. Mangaroska, K. Sharma, D. Gasevic, and M. Giannakos, "Exploring students' cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity," *J. Comput. Assist. Learn.*, vol. 38, no. 1, pp. 40–59, Feb. 2022.
[29] Y. Wang, "On the cognitive processes of human perception with emotions, motivations, and attitudes," *Int. J. Cognit. Informat. Natural Intell.*, vol. 1, no. 4, pp. 1–13, Oct. 2007.
[30] A. Holzinger, M. Errath, G. Searle, B. Thurnher, and W. Slany, "From extreme programming and usability engineering to extreme usability in software engineering education (XP+UE⟶XU)," in *Proc. 29th Annu. Int. Comput. Softw. Appl. Conf. (COMPSAC)*, Jul. 2005, pp. 169–172.
[31] L. Williams, "Agile software development methodologies and practices," *Adv. Comput.*, vol. 80, pp. 1–44, Jan. 2010.
[32] Manifesto for agile software development. K. Beck. (2001). *Manifesto for Agile Software Development*. Accessed: Dec. 22, 2021. [Online]. Available: https://agilemanifesto.org/
[33] K. Schwaber and M. Beedle, *Agile Software Development With Scrum*. Upper Saddle River, NJ, USA: Prentice-Hall, 2002.
[34] N. H. Z. Abai, J. H. Yahaya, and A. Deraman, "User requirement analysis in data warehouse design: A review," *Proc. Technol.*, vol. 11, pp. 801–806, Jan. 2013.
[35] F. Herman and F. Heidmann, "User requirement analysis and interface conception for a mobile, location-based fair guide," in *Proc. Int. Conf. Mobile Human-Comput. Interact.*, Pisa, Italy, 2002, pp. 388–392.
[36] A. Sediyono and A. Ariwibowo, "Software requirement specification of intelligent system for monitoring and preventing smartphone addiction," in *Proc. Int. Conf. Smart Cities, Autom. Intell. Comput. Syst. (ICONSONICS)*, Yogyakarta, Indonesia, Nov. 2017, pp. 54–58.
[37] X. Li, Z. Liu, and J. He, "Formal and use-case driven requirement analysis in UML," in *Proc. 25th Annu. Int. Comput. Softw. Appl. Conf. (COMPSAC)*, Chicago, IL, USA, 2001, pp. 215–224.
[38] Y. Wang, S. Yu, and T. Xu, "A user requirement driven framework for collaborative design knowledge management," *Adv. Eng. Inform.*, vol. 33, pp. 16–28, Aug. 2017.
[39] A. Cockburn, *Writing Effective Use Cases*. Upper Saddle River, NJ, USA: Addison-Wesley, 2001.

[40] F. Calefato and F. Lanubile, "A Hub-and-Spoke model for tool integration in distributed development," in *Proc. IEEE 11th Int. Conf. Global Softw. Eng. (ICGSE)*, Aug. 2016, pp. 129–133.
[41] C. Burr, N. Cristianini, and J. Ladyman, "An analysis of the interaction between intelligent software agents and human users," *Minds Mach.*, vol. 28, no. 4, pp. 735–774, 2018.
[42] C. Andor, A. Joó, and L. Mérö, "Galois-lattices: A possible representation of knowledge structures," *Eval. Educ.*, vol. 9, no. 2, pp. 207–215, Jan. 1985.
[43] R. Plutchik, "A general psychoevolutionary theory of emotion," in *Theories of Emotion*. Cambridge, MA, USA: Academic Press, 1980, pp. 3–33.
[44] P. Ekman, "An argument for basic emotions," *Cognit. Emotion*, vol. 6, nos. 3–4, pp. 169–200, 1992.
[45] (2019). *LattePanda V1.0 4G/64G Specifications, LattePanda*. Accessed: Dec. 22, 2021. [Online]. Available: http://docs.lattepanda.com/
[46] (2019). *NVIDIA Jetson Nano: Technical Specifications, NVIDIA*. Accessed: Dec. 22, 2021. [Online]. Available: https://developer.nvidia.com/embedded/jetson-nano-developer-kit
[47] C.-N. Anagnostopoulos, T. Iliou, and I. Giannoukos, "Features and classifiers for emotion recognition from speech: A survey from 2000 to 2011," *Artif. Intell. Rev.*, vol. 43, no. 2, pp. 155–177, Feb. 2015.
[48] A. M. Badshah, J. Ahmad, N. Rahim, and S. W. Baik, "Speech emotion recognition from spectrograms with deep convolutional neural network," in *Proc. Int. Conf. Platform Technol. Service (PlatCon)*, Busan, South Korea, Feb. 2017, pp. 1–5.
[49] P. Ekman and W. V. Friesen, "Facial action coding system," in *Environmental Psychology & Nonverbal Behavior*, 1978.
[50] 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 ed. New York, NY, USA: Oxford Univ. Press, 2020.
[51] P. Lucey, J. F. Cohn, T. Kanade, J. Saragih, Z. Ambadar, and I. Matthews, "The extended Cohn-Kanade dataset (CK+): A complete dataset for action unit and emotion-specified expression," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (Workshops)*, San Francisco, CA, USA, Jun. 2010, pp. 94–101.
[52] C. Szegedy, V. Vanhoucke, S. Ioffe, J. Shlens, and Z. Wojna, "Rethinking the inception architecture for computer vision," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA, Jun. 2016, pp. 2818–2826.
[53] C. H. Wu, J. C. Lin, and W. L. Wei, "Survey on audiovisual emotion recognition: Databases, features, and data fusion strategies," *APSIPA Trans. Signal Inf. Process.*, vol. 3, no. 12, pp. 1–18, Nov. 2014.
[54] K. Dupuis and M. K. Pichora-Fuller. *Toronto Emotional Speech Set*. Distributed by University of Toronto. Accessed: Jun. 21, 2010. [Online]. Available: https://tspace.library.utoronto.ca/handle/1807/24487
[55] J. A. Russell and A. Mehrabian, "Evidence for a three-factor theory of emotions," *J. Res. Pers.*, vol. 11, no. 3, pp. 273–294, 1977.
[56] S. Mohammad, "Obtaining reliable human ratings of valence, arousal, and dominance for 20,000 English words," in *Proc. 56th Annu. Meeting Assoc. Comput. Linguistics*, Melbourne, VIC, Australia, vol. 1, 2018, pp. 174–184.
[57] S. R. Livingstone and F. A. Russo, "The Ryerson audio-visual database of emotional speech and song (RAVDESS): A dynamic, multimodal set of facial and vocal expressions in North American English," *PLoS ONE*, vol. 13, no. 5, May 2018, Art. no. e0196391.
[58] N. Ayari, H. Abdelkawy, A. Chibani, and Y. Amirat, "Hybrid model-based emotion contextual recognition for cognitive assistance services," *IEEE Trans. Cybern.*, early access, Sep. 10, 2020, doi: 10.1109/TCYB.2020.3013112.
[59] A. Hong, N. Lunscher, T. Hu, and Y. Tsuboi, "A multimodal emotional human–robot interaction architecture for social robots engaged in bidirectional communication," *IEEE Trans. Cybern.*, vol. 51, no. 12, pp. 5954–5968, Dec. 2021.
[60] A. K. Jain and A. Ross, "Multibiometric systems," *Commun. ACM*, vol. 47, no. 1, pp. 34–40, Jan. 2004.
[61] A. Mehrabian, "Communication without words," in *Communication Theory*, C. D. Mortenson, Ed. New York, NY, USA: Routledge, 2017, pp. 193–200.
[62] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, "Designing theory-driven user-centric explainable AI," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, Glasgow, U.K., May 2019, pp. 1–15.

**JORDAN VICE** received the B.Eng. degree (Hons.) in mechatronic engineering from Curtin University, where he is currently pursuing the Ph.D. degree in mechatronic engineering. His research interests include artificial intelligence, explainable artificial intelligence, machine learning, real-time assessment of affective states, and multimodal affective computing. He received the 2019 Proxima Consulting Prize for Most Outstanding Final Year Project in mechatronic engineering.

**MASOOD MEHMOOD KHAN** (Member, IEEE) received the B.E. (mechanical), M.S. (mechanical), and Ph.D. (computational engineering) degrees. He had taught at the National University of Computer and Emerging Sciences, Jefri Bolkiah College of Engineering, and the American University of Sharjah, before joining the Faculty of Science and Engineering, Curtin University, WA. He has published more than 50 peer-reviewed articles in top quality journals and conference proceedings. His research interests include machine learning, affective computing, machine vision, human–computer interaction, and robotics. He is a fellow of the Higher Education Academy.