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

Masood Khan ¹ and Jordan Vice ²

¹Curtin University
²Affiliation not available

November 1, 2023

Abstract

This paper builds upon the theoretical foundations of the Accountable explainable Artificial Intelligence (AXAI) capability framework presented in part one of this paper. This part demonstrates the incorporation of the AXAI capability in the real time Affective State Assessment Module (ASAM) of a robotic system. The paper argues that adhering to the extreme Programming (XP) practices would help in understanding user behavior while systematically incorporating the AXAI capability in AI systems. Issues pertaining to identification of ethical, technical, functional, and domain-specific system requirements were resolved using an appropriate software design process, thus establishing confidence in the system. The presented ASAM synthesizes discrete and continuous models of affective state expressions and classifies them in real-time. Input, processed and output data are continuously shared with users via a graphical user interface (GUI) for providing reasons behind decisions. The GUI also disseminates information about local reasoning, data handling and decision-making. We hope this work will initiate further investigations on selection of suitable software design practices for incorporating the AXAI capability in AI systems, enhancing AI system acceptability and utility and, establishing a chain of responsibility.
Toward Accountable and Explainable Artificial Intelligence Part Two: The Framework Implementation

JORDAN VICE1 AND MASOOD M. KHAN1 (Member, IEEE)
1Faculty of Science and Engineering, Curtin University, Perth, Australia
Corresponding author: Masood M. Khan (e-mail: 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. This part demonstrates the incorporation of the AXAI capability in the real time Affective State Assessment Module (ASAM) of a robotic system. The paper argues that adhering to the eXtreme Programming (XP) practices would help in understanding user behavior while systematically incorporating the AXAI capability in AI systems. Issues pertaining to identification of ethical, technical, functional, and domain-specific system requirements were resolved using an appropriate software design process, thus establishing confidence in the system. The presented ASAM synthesizes discrete and continuous models of affective state expressions and classifies them in real-time. Input, processed and output data are continuously shared with users via a graphical user interface (GUI) for providing reasons behind decisions. The GUI also disseminates information about local reasoning, data handling and decision-making. We hope this work will initiate further investigations on selection of suitable software design practices for incorporating the AXAI capability in AI systems, enhancing AI system acceptability and utility and, establishing a chain of responsibility.

INDEX TERMS Artificial Intelligence, Explainable Artificial Intelligence, Affective Computing, System Design, Classifier Design, Interactive Graphical User Interface, Human-Computer Interface

I. INTRODUCTION

Experts from the domains of logic programming, automated reasoning and software engineering lead the artificial intelligence (AI) system design process [1-3]. Consequently, practitioners, not 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 offerings and practitioners’ needs. Such gaps need to be filled for embedding explainability and transparency in AI systems. Presenting statistical and probabilistic data alone does not help practitioners in understanding how domain-specific requirements were met [4, 5]. Particularly, the limited amount of explanations offered to users does not 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 systems decisions would suffice incorporating the XAI in AI systems [11]. However, real-life use of these features for incorporating XAI in AI systems is not common yet. 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 de-
sign 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]. This work makes four major contributions:

1) It demonstrates how to add the AXAI capability into AI systems using an appropriate software design and development process (SDDP).
2) It shows how XP practices naturally align with the AI SDDP and help in embedding AXAI capabilities into a system.
3) It introduces the architectural and functional details of a portable, real-time ASAM that can be attached to any robotic system equipped with the necessary peripheral devices (e.g., camera and microphone).
4) It demonstrates using visual and vocal cues in the GUI of an AI system to disseminate information pertaining to system’s predictive accuracy, transparency, and accountability.

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 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.

The AXAI capability framework proposed in part one of this paper was designed to overcome these limitations. The following sections describe the process of incorporating AXAI capabilities and shows how quantifiable measures of explainability, accuracy, and accountability are added to ASA systems. The process is explained by providing AXAI-relevant information and explaining local ASA reasoning at each stage. This process would help system designers and users in reducing AI system algorithmic biases. A comparison of the conventional ASA system with an AXAI-capable system in Fig. 2 shows how the two systems differ. 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. If designers fail to notice the problem during the design stages, users would discover them while inspecting the quantifiable measures provided by adding the AXAI capability in 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 [30]. 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 [30]:

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 formal structures that ignores 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.

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 pro-
cesses 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 [32]. 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, [33] examines user requirements in the context of data warehouse design, [34] attempts to determine requirements for interface design of a mobile location-based fair guide and [35] 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 [36] 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 [37]. We carried out the dynamic requirement analy-
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 [33]. 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.

ses 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 [38]. The modular structure of the ASAM, functional descriptions of various modules, and elements of the AXAI capability framework are visualised in Fig. 4.

We initially developed a high-level functional description of the ASAM that helped in performing user requirement analyses and developing specifications for each submodule of the ASAM. A “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 [38]. 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.

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

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

The use case described in Figure 7 illustrates how the GUI display use case is formalized. The use case is structured to provide a clear understanding of the interactions between the system and the user. The use case is divided into different scenarios, each describing a specific aspect of the interaction. The use case is based on a formal description that includes the following elements:

1. **Actor**: The ASAM user
2. **Goal**: To provide the user with comprehensive information about the system's functionality
3. **Scope**: GUI display use case
4. **Level**: Low-level
5. **Success Scenarios**:
   - The user is able to interact with the GUI and receive comprehensive information
   - The user is able to navigate through the system and access relevant data

The use case is formalized using a combination of concepts borrowed from the fields of artificial intelligence and game theory. The formal description is structured using a set of rules and notations that are used to define the interactions and the system's behavior. The use case is designed to be scalable and extendable, allowing for the integration of new features and functionalities.

Figure 8. Formal description of the partial information use case.

The use case presented in Figure 8 illustrates how the partial information use case is formalized. The use case is structured to provide a clear understanding of the interactions between the system and the user. The use case is divided into different scenarios, each describing a specific aspect of the interaction. The use case is based on a formal description that includes the following elements:

1. **Actor**: The ASAM user
2. **Goal**: To provide the user with partial information about the system's functionality
3. **Scope**: Partial information use case
4. **Level**: Low-level
5. **Success Scenarios**:
   - The user is able to interact with the GUI and receive partial information
   - The user is able to navigate through the system and access relevant data

The use case is formalized using a combination of concepts borrowed from the fields of artificial intelligence and game theory. The formal description is structured using a set of rules and notations that are used to define the interactions and the system's behavior. The use case is designed to be scalable and extendable, allowing for the integration of new features and functionalities.

Figure 9. Formal description of the comprehensibility use case.

The use case presented in Figure 9 illustrates how the comprehensibility use case is formalized. The use case is structured to provide a clear understanding of the interactions between the system and the user. The use case is divided into different scenarios, each describing a specific aspect of the interaction. The use case is based on a formal description that includes the following elements:

1. **Actor**: The ASAM user
2. **Goal**: To provide the user with comprehensible information about the system's functionality
3. **Scope**: Comprehensibility use case
4. **Level**: High-level
5. **Success Scenarios**:
   - The user is able to interact with the GUI and receive comprehensible information
   - The user is able to navigate through the system and access relevant data

The use case is formalized using a combination of concepts borrowed from the fields of artificial intelligence and game theory. The formal description is structured using a set of rules and notations that are used to define the interactions and the system's behavior. The use case is designed to be scalable and extendable, allowing for the integration of new features and functionalities.

In order to iteratively and collaboratively assess system performance at each stage, we used a table for recording user motivation, user behavior, and user perceived value of the ASAM features. The table was based on the Gallois-Lattices structure shown in Figure 12. In our scheme, user motivation comprised of three elements: need, desire, and ability. Need refers to the necessity of the feature, desire refers to the user's willingness to use the feature, and ability refers to the user's capability to use the feature.

The use case presented in Figure 9 illustrates how comprehensibility was incorporated in the ASAM’s AXAI capabilities. This use case 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 Figure 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 Figure 11 illustrates how the predictive accuracy of the ASAM would be embedded for assessing ASAM’s AXAI capability.
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 behavior 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 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.

As shown previously in Fig. 3 two collaborative iterations of user requirement analysis and assessment were sufficient 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.
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 in acoustic features [45, 46]. Linguistics in the context of the ASAM describe the content of “what was said,” representing 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 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 in acoustic features [45, 46].

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 ASAM. 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 [49] 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 [50]. The construction of the ASAM’s categorization and classification models was performed using variations of the InceptionV3 convolutional neural network (CNN) architecture described by [50]. The construction of the ASAM’s categorization and classification models was performed using variations of the InceptionV3 convolutional neural network (CNN) architecture described by [50].

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

| Category | Model | Valence | Parameters | Epochs | Time to Train | Output Classes | Cost-function Parameters | Validation Accuracy | Validation Loss |
|----------|-------|---------|------------|--------|---------------|------------------|------------------------|--------------------|-----------------|
| Categorization Model 1 | Classifier Model 1 | Positive-High Valence | (24, 1) | 2 | 100 | Sigmoid | 0.0708 | 0.4887 | 98.95% |
| Categorization Model 2 | Classifier Model 2 | Low Valence | (24, 1) | 100 | 0.01 | Linear | 87.10% | 0.6959 | 98.57% |
| Categorization Model 3 | Classifier Model 3 | Negative-High Valence | (24, 1) | 1 | 0.01 | Mixed7 | 80.00% | 0.1471 | 95.24% |

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

| Category | Model | Valence | Parameters | Epochs | Time to Train | Output Classes | Cost-function Parameters | Validation Accuracy | Validation Loss |
|----------|-------|---------|------------|--------|---------------|------------------|------------------------|--------------------|-----------------|
| Categorization Model 1 | Classifier Model 1 | Positive-High Valence | (24, 1) | 2 | 100 | Sigmoid | 0.0708 | 0.4887 | 98.95% |
| Categorization Model 2 | Classifier Model 2 | Low Valence | (24, 1) | 100 | 0.01 | Linear | 87.10% | 0.6959 | 98.57% |
| Categorization Model 3 | Classifier Model 3 | Negative-High Valence | (24, 1) | 1 | 0.01 | Mixed7 | 80.00% | 0.1471 | 95.24% |

The textual data was 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 in acoustic features [45, 46]. Linguistics in the context of the ASAM describe the content of “what was said,” representing 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 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 in acoustic features [45, 46].

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 [53] model, which represents states in a 3-dimensional pleasure/valence, arousal, and dominance.
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 |

(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 [54] 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:

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., $V_{\text{ave}}, A_{\text{ave}}, D_{\text{ave}}$:
   
   $$X_{\text{ave}} = \frac{\sum_{n=1}^{N} X_n}{N}.$$  \hspace{1cm} (2)

4) For each affective state class ‘c’, determine the 3D Euclidean distance $\Delta AB_c$ between the input utterance and $c$, modelled by:
   
   $$\Delta AB_c = \sqrt{(V_c - V_{\text{ave}})^2 + (A_c - A_{\text{ave}})^2 + (D_c - D_{\text{ave}})^2}.$$  \hspace{1cm} (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’:
   
   $$DBS_c = \frac{1}{1 + \Delta AB_c}.$$  \hspace{1cm} (4)

The ASAM was further validated on the Ryerson Audio-visual Database of Emotional Speech and Song (RAVDESS) [55], 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: $r_{\text{tst-trn}} = 1.0$, $d_{\text{tst}} = 0.811$ and $O_{\text{trn}} = 0.8559$, 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 [58]. Each sub-classifier output (facial expression, paralinguistic, linguistic) combines as having weights as such (for each discrete state):

$$P_{\text{Final}} = W_1P_{\text{FACE}} + W_2P_{\text{PARA}} + W_3DBS$$  \hspace{1cm} (5)

where ‘$P_{\text{Final}}$’ is the final prediction accuracy, ‘$P_{\text{FACE}}$’ is the facial expression prediction accuracy, ‘$P_{\text{PARA}}$’ 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 [59] 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:

$$P_{\text{Final}} = 0.55P_{\text{FACE}} + 0.38P_{\text{PARA}} + 0.07DBS$$  \hspace{1cm} (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
Vice and Khan: Toward AXAI Part two: The Framework Implementation

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 [55]. The Transcription box contains the string “this is a test purely for the purposes of paper” with the highlighted words being {test, purely, paper}.

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 = \( \frac{n_P}{N_T} \times 100\% \), where \( n_P \) 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 overabundance of information would not benefit the AI system design in becoming more transparent and explainable [60]. 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 machine learning (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]. The AXAI capability framework proposed in part one of this paper and its implementation reported in the preceding sections of this part, exploited and relied on these four features. As discussed prior, 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 and 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 SSDP 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 AI systems having accountability, comprehensibility, and accuracy built into them.

VII. CONCLUSION
The need for incorporating explanations in AI and ML systems was realized in the 1970’s. A good number of recently published research papers in medical diagnosis, cognition, psychiatric and psychological sciences, law and criminal investigations, governance, image understanding, and some other domains highlight the significance of incorporating AXAI capabilities in relevant AI and ML systems.

The first part of this paper introduced the AXAI capability framework. This part of the paper details the incorporation of AXAI capabilities in an AI system. Hence, this paper serves as a tutorial, providing specific information on various functional modules, link structures to connect users with use cases and, examples of use cases for incorporating AXAI capabilities in AI systems. Though the domain of our applica-
tion was affective computing, this paper introduces a domainagnostic and portable methodology for incorporating AXAI capability in AI systems.

Little work has been carried out and reported on the use of software design practices in building AI systems. Current literature offers very little on the application of agile software design practices in AI system design. Consequently, ad hoc practices are used for designing and implementing AI and ML systems. This work provides insight into the suitability of agile software methods for AI system design. Of various agile methodologies, extreme programming (XP) practices were found to be useful for incorporating AXAI capability. The systematic, localized and iterative development of an AI system was helpful in collaboratively discovering user requirements and understanding user behavior. Tools such as conceptual models and use cases augmented the progressive enhancement of the system and allowed identifying any shortcomings in functions and explanations of the system.

During the ASAM design, user motivation (as need, desire and expectations), user behavior (as either use, revise or reject a function) and user-perceived value of features (seen as either good, complicated or revisable) helped in determining system requirements. Discussions on user motivation, user behavior and user perceived value of an AI system in the context of affective state classifier design are not common in affective computing literature. Hopefully, these discussions will promote the idea of engaging AI system developers and users during the system analysis and design process. Through the user-developer engagements, the design and implementation of the ASAM was sensitive to the ethical, functional, technical and domain-specific implications of a contemporary AI system. The AXAI capability incorporation was reliant on the quality of inputs, level and nature of interactions between input cues, robust mechanisms of combining the input and processed data, and quantity and quality of the training data. Building the AXAI capability of a system using these features would help in building trust in system decisions, thus, helping users learn from inferences.

The proposed framework and its sample implementation in the ASAM is anticipated to initiate investigations on various aspects of building, assessing and incorporating AXAI capabilities in affective state assessment systems. This work is also expected 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, 2021.
[2] D. Michie, “Machine Learning in the next Five Years,” in Proc. 3rd Eur. Conf. on Eur. Working Session on Learn., Glasgow, UK, 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. on Log. Program., London, UK, 2017, pp. 52-67.
[4] T. Miller, “Explanation in artificial intelligence: Insights from the social sciences,” Artif. Intell., vol. 267, pp. 1-38, 2019.
[5] M. - A. Vu et al., “A Shared Vision for Machine Learning in Neuroscience,” J. of Neuroscience, vol. 38, no. 7, pp. 1601-1607, 2018.
[6] U. Pawar, D. O’Shea, S. Rea, and R. O’Reilly, “Explainable AI in healthcare,” presented at the 2020 Int. Conf. on Cyber Situational Awareness, Data Analytics and Assessment (CyberSA), Dublin, Ireland, Jun. 15-19, 2020.
[7] Q. V. Liao, D. Gruen, and S. Miller, “Questioning the AI: Informing design practices for explainable AI user experiences,” in Proc. 2020 CHI Conf. on Human Factors in Comput. Syst., Honolulu, HI, USA, 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, 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, 2021.
[11] B. J. Murray et al., “Explainable AI for the Choquet Integral,” IEEE Trans. on Emerging Topics in Comput. Intell., vol. 5, no. 4, pp. 520-529, 2020.
[12] K. Beck, “Embracing change with extreme programming,” Computer, vol. 32, no. 10, pp. 70-77, 1999.
[13] S. Nayak, V. Sharma, K. Panda, and S. Uuttarkabat, “Affective State Analysis Through Visual and Thermal Image Sequences. 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. 2016 Int. Conf. on Adv. Robot. and Mechatronics (ICARM), Macau, China, 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. 2016, IEEE Int. Conf. on Adv. Intell. Mechatronics (AIM), Banff, Canada, 2016, pp. 463-467.
[16] D. Cernea, and A. Kerren, “A survey of technologies on the rise for emotion-enhanced interaction,” J. of Vis. Lang. & Comput., vol. 31, pp. 70-86, 2015.
[17] M. M. Khan, “Cluster Analytic Detection of Disgust-Arousal,” Proc. 9th Int. Conf. of Intell. Syst. Des. and 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. on Affective Comput., vol. 8, no. 3, pp. 412-425, 2017.
[19] S. K. D’Mello, and J. Kory, “A review and meta-analysis of multimodal affect detection systems,” ACM Comput. Surv. (CSUR), vol. 47, no. 3, pp. 1-36, 2015.
[20] C. E. Izard, The psychology of emotions. New York, NY, USA: Springer Science & Business Media.
[21] K. Lochner, “Affect Mood and Emotions,” in Successful emotions, Wiesbaden, Germany: Springer, 2016, pp. 43-69.
[22] D. Watson, L. A. Clark, A. Tellegen, “Development and validation of brief measures of positive and negative affect: the PANAS scales,” J. of Personality and 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. of Behav. Therapy and Exp. Psychiatry, vol. 25, no. 1, pp. 49-59, 1994.
[24] R. Steyer, P. Schwenkmezger, P. Noltz, and M. Eid, “Testtheoretische Analysen des Mehrdimensionalen Befindlichkeitsfragebogen (MDBF) [Theoretische Analyse des Befindlichkeitstests der Mehrdimensionalen Befindlichkeitsfragebogen (MDBF)],” Diagnostica, vol. 40, no. 4, pp. 320-328, 1997.
[25] A. M. Rief, and R. W. Levenson, “Continuous measurement of emotion,” in Handbook of emotion elicitation and assessment, J. A. Coan, and J. J. Allen, Eds. Oxford, UK: Oxford University Press, 2007, pp. 286-297.
[26] J. Bose, M. M. Khan, and S. Yanushkevich, “Multimodal Models for Contextual Affect Assessment in Real-Time,” in Proc. 2019 IEEE 1st Int. Conf. on Cogn. Mach. Intell. (CogMI), Los Angeles, CA, USA, 2019, pp. 1-14.
[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. on Affective Comput., pp. 1-14, 2021.
[28] K. Mangaroska, K. Sharma, D. Galievich, and M. Giannakos, “Exploring students’ cognitive and affective states during problem solving through multimodal data: Lessons learned from a programming activity,” J. of Comput. Assisted Learn., pp. 1-20, 2021.
[29] Y. Wang, “On the cognitive processes of human perception with emotions, motivations, and attitudes,” Int. J. of Cogn. Inform. and Natural Intell. (IJICIN), vol. 1, no. 4, pp. 1-13, 2007.
[30] L. Williams, “Agile software development methodologies and practices,” Advances in Comput., vol. 80, pp. 1-44, 2010.

[31] Manifesto for agile software development. K. Schwaber, and M. Beedle, 2nd ed., New York, NY, USA: Prentice Hall, 2002.

[32] N. H. Z. Abai, J. H. Yahaya, and A. Deraman, “User requirement analysis in data warehouse design: a review,” Procedia Technol., vol. 11, pp. 801-806, 2013.

[33] K. Dupuis, and M. K. Pichora-Fuller, “Toronto Emotional Contextual Recognition for Cognitive Assistance Services,” IEEE Trans. on Cybern., pp. 1-10, 2020.

[34] A. Mehrabian, “Communication without words,” in Communication Theory, C. D. Mortensen, Ed. New York, NY, USA: Routledge, 2018, pp. 193-200.

[35] D. Wang, Q. Yang, A. Abdul, and B. Y. Lim, “Designing theory-driven user-centric explainable AI,” in Proc. 2019 CHI Conf. on Human Factors in Comput. Syst., Glasgow, Scotland, 2019, pp. 1-15.