



Human emotion modeling (HEM): an interface for IoT systems

Mohammed R. Elkobaisi¹ · Fadi Al Machot²

Received: 15 May 2020 / Accepted: 5 August 2021 / Published online: 22 August 2021
© The Author(s) 2021

Abstract

The use of IoT-based Emotion Recognition (ER) systems is in increasing demand in many domains such as active and assisted living (AAL), health care and industry. Combining the emotion and the context in a unified system could enhance the human support scope, but it is currently a challenging task due to the lack of a common interface that is capable to provide such a combination. In this sense, we aim at providing a novel approach based on a modeling language that can be used even by caregivers or non-experts to model human emotion w.r.t. context for human support services. The proposed modeling approach is based on Domain-Specific Modeling Language (DSML) which helps to integrate different IoT data sources in AAL environment. Consequently, it provides a conceptual support level related to the current emotional states of the observed subject. For the evaluation, we show the evaluation of the well-validated System Usability Score (SUS) to prove that the proposed modeling language achieves high performance in terms of usability and learn-ability metrics. Furthermore, we evaluate the performance at runtime of the model instantiation by measuring the execution time using well-known IoT services.

Keywords Internet of things (IoT) · Domain-Specific Modeling Language (DSML) · Human emotion (HE) · Active and assisted living (AAL)

1 Introduction

Currently, most *Emotion Recognition (ER)* systems focus on identifying a small and specific set of emotional states. Considering them in the current form does not provide enough information for deriving appropriate and comprehensive human support in a given environment.

Despite the advances in modern emotion recognition technologies, representing the relationship between emotional response and the context was not deeply investigated. However, understanding the context leads to better human support performance. Context understanding is related to context-aware applications that allow sensing the context information associated with sensor data and acting upon it, such that the interpretation of knowledge-based situation

becomes more meaningful (Perera et al. 2014). Consequently, the support system must adapt to the dynamic environment that is more context- and situation- dependent. In other words, AAL-domain users require artifacts to represent the changes in an emotional state for the subject that immersed in various situations in real-life settings. Considering the variation in the situation is essential to establish accurate knowledge about a person's emotion over time.

Besides, one of the fundamental challenges for the IoT is how the complex system is simplified for end users so that they can configure and use it easily (Alberti 2013). End-user can probably be a professional or a common person without a modeling experience background. Inspired by this idea, a domain-specific modeling language DSML could be a powerful tool for the non-technical or non-expert users by hiding the implementation complexity.

This paper addresses this problem by introducing a model-based ER interface based on conceptual foundations that can independently connect existing ER systems with the context. To specify such foundations, a metamodel-based language is introduced to conceptualize the context.

For evaluation purposes, we conducted real-world IoT applications, e.g., facial emotions in real-life settings using

✉ Mohammed R. Elkobaisi
M3mohammed@edu.aau.at
Fadi Al Machot
fadialmachot@gmail.com

¹ Institute for Applied Informatics, Application Engineering, Alpen-Adria University, 9020 Klagenfurt, Austria

² Faculty of Science and Technology, Norwegian University for Life Science (NMBU), 1430 Ås, Norway

(*Microsoft Cognitive Services-API*¹ is used to recognize emotion without machine-learning expertise, *Google Cloud Vision-API*² provides a service that allows developers to detect emotion as well as signs, landmarks, objects, text within a single image, and *CLMtrackr*³ allows systems to read facial expressions in videos or images).

The rest of this paper is organized as follows. Section 2 presents the problem and the contribution of the current work. Section 3 discusses the related work and its limitations. Section 4 illustrates the methodology including meta-model, modeling elements, and modeling consistency. Section 5 shows a running example based on the AAL scenario case. Section 6 explains how to integrate IoT Applications at runtime. Section 7 discusses the evaluation method. Finally, Sections 8 and 9 present the discussion, conclusions, and future work.

2 Problem definition and contribution

Existing emotion representation methods have not reached their full potential, and need to be extended or changed to make them usable in practice across a broad range of domain users. The existing emotion languages (see Sect. 3) are pure textual without featuring higher capabilities for knowledge representation or automated reasoning. None of these languages satisfy the representation of emotions in dynamic situations. Additionally, representation languages should consider stakeholder diversity regarding the knowledge, and experience, thereby making it easier for non-programmers and domain experts to implement modeling tasks. Consequently, we propose (a) a modeling tool that overcomes the limitation of existing emotion representation languages w.r.t. semantic coverage and universality, and (b) the proposed tool is an interface that is user-friendly and offers a comprehensive meta-model and simple modeling elements that can be used by non-experts.

We follow a Meta-Object Facility (MOF)⁴ and systematic procedure like presented in Frank (2011), Michael and Mayr (2015) to identify the structure of the modeling method to be designed. Furthermore, we implemented the proposed system as a set of concepts, rules, and constraints to provide an IoT system interface. The model can be generated in a machine-readable format which is a formal description that can be used for further reasoning tasks. Domain-specific modeling languages (DSMLs) “*are specialized languages for a particular application area, which use the*

concepts and notations established in the field” (Zarrin and Baumeister 2018).

However, this work has been carried out within the framework of the on-going HBMS project (Michael et al. 2018). The HBMS aims at assisting people with cognitive impairments to live independently at home. The current work should enable the HBMS to deal with emotional aspects.

3 Related work

This section describes popular emotion representation languages and the purpose behind their development.

(EmotionML) *Emotion Markup Language* (Schröder et al. 2011) was developed to express emotions in three main ways: manual annotation for emotion data such as (images, videos, or speech), automatic emotion-based state recognition, and emotion-related behavior generation and reasoning. EmotionML describes emotion in terms of Ekman’s theory (Ekman 1992), dimensional theory (Mehrabian and Russell 1974), appraisals and/or action tendencies. (EARL) *Emotion Annotation and Representation Language* (Schröder 2006) was created in order to represent emotion in technological contexts. EARL represents emotions as basic, dimensional, or sets of appraisal scale. (VHML) *Virtual Human Markup Language* was designed to adapt different aspects of Human-Computer Interaction (HCI) with regards to Facial Animation, Body Animation, Dialogue Manager interaction, Text to Speech production, Emotional Representation plus Hyper and Multimedia information. (SSML) *Speech Synthesis Markup Language* (Baggia and Bagshaw 2010) It is an XML-based markup language for supporting the creation of synthetic speech in Web and other applications. The essential role of this language is to provide authors of synthesizable content a standard way to control aspects of speech such as pronunciation, volume, pitch, rate, etc. across different synthesis-capable platforms.

However, the above mentioned languages are generic purpose languages. They convert emotion data to XML format, without offering higher-level semantic concepts or a graphical representation. Besides, they lack the notions of context or capability of the person, which limits their applicability for describing context- or capability-related situations. In addition, different domain-specific concepts come with various rules, constraints, and semantics. These have to be properly described to provide benefits like semantic consistency across different representations, guidance, and error avoidance. Moreover, the ontologies may play an important role here as they support the definition of constraints, rules, and semantics using logic-based concepts (Terkaj et al. 2012). Studies (Walter et al. 2012; Liao et al. 2015) explained how to integrate domain-specific languages with ontology languages and automated

¹ <https://azure.microsoft.com/en-us/services/cognitive-services>.

² <https://cloud.google.com/vision>.

³ <https://github.com/auduno/clmtrackr>.

⁴ <https://www.omg.org/mof>.

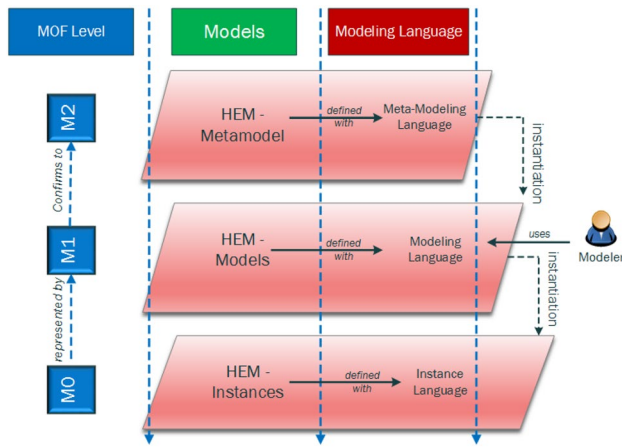


Fig. 1 Human Emotion Modeling (HEM) hierarchy (M2, meta-model level; M1, model level, and M0, instance level)

reasoning services at the meta-model level. The use of the formal semantics of the *Web Ontology Language (OWL)* together with reasoning services for addressing constraint definition, suggestions, and debugging is discussed in Antunes et al. (2014). As a summary, we identify the following challenges of the current emotion ontologies: (1) several emotion ontologies, e.g., (Sam and Chatwin 2012; Sykora et al. 2013; Khoonnaret SAN 2017) introduce similar aspects, such as similarity in the classes and the emotion type. For instance, the Ekman basic emotion (Ekman 1992) is adapted in most ontologies. (2) The available emotion ontologies are not general enough to cover all emotion properties. (3) Ontological representation is often difficult to understand by stakeholders in the AAL domain (e.g., doctor, nurse, caregiver) compared with the conceptual representation. Generally, conceptual models support direct modeling, leading to representations that are close to how humans perceive things in the real world. Conceptual models provide better understandability. Instead, most ontological representations that rely on formal semantics structures. This could lead to large numbers of concepts that are needed, for instance, a one-page conceptual representation is likely to require several pages of ontological axioms to describe the same situation.

4 Methodology

The representation language is defined on both conceptual and implementation level to capture emotional response in a way which is compatible with possible relevant contextual concepts. We define HEM (Fig. 1) on three levels of language definition hierarchy introduced:

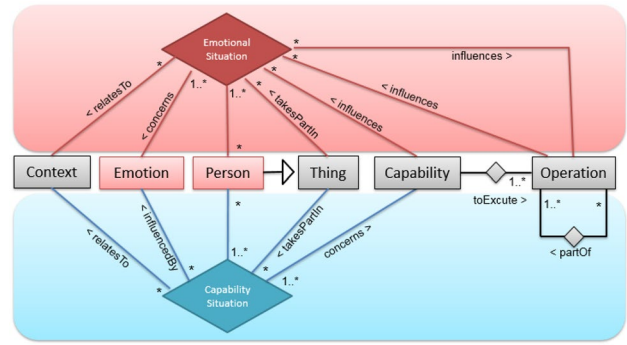


Fig. 2 A meta-model of proposed modeling system (HEM)

- The meta-model for HEM is defined on M2 level by means of the ADOxx[®] metamodeling framework (ADOxx 2021).
- The modelers are able to create the HEM models by means of this tool using HEM graphical notation (*instantiating the meta-model on M1 level*).
- To encode the data coming from the external emotion sources, HEM-models are *instantiated on M0 level* by means of text-based HEM-Instance definition language optimized for executive data exchange.

Generally, designing HEM as a Domain Specific Modeling Language (DSML) comprises at least three main aspects (Cho et al. 2012; Kleppe 2008): (a) abstract syntax that describes the concepts of the domain and relationships between concepts that is usually identified by a meta-model, (b) concrete syntax based on abstract syntax that introduces textual or graphical notations to the modeler, and (c) semantic that usually involves a formal analysis over the models and translation between the language itself and another language (*such as XML or Java*).

4.1 Meta-model

A meta-model facilitates analyzing the complexity of the real world; it is used as bases for defining our modeling system. Figure 2 shows the core elements of the metamodel without going into detail, as there is not enough space available for this.

According to the meta-model, we have two main relationships and shared core elements. Each Core Element (CE) and its related value is obtained as a set of attributes.

$$CE = \{ a_1, a_2, \dots, a_n \} \tag{1}$$

Where ‘‘CE’’ is a core element and ‘‘a’’ its attribute (for instance, in the scenario of ‘‘Watch a Movie’’ a core element ‘‘Context’’ = *time*: weekend, *location* : cinema, *companion*:

girlfriend, and a core element “**Emotion**” = *anger:0, fear:0, happy:1, sad:0, surprise:0*.

To represent an Emotional Situation Relationship (ESR) as a cumulative relationship, we conjunct, each core element that participates in the ESR.

$$ESR = \langle CE_1 \wedge CE_2 \wedge \dots \wedge CE_n \rangle \tag{2}$$

Then, $ESR = \langle \text{“Context”} = (\text{time: weekend, location: cinema, companion: girlfriend}) \wedge \text{“Emotion”} = (\text{anger:0, fear:0, happy:1, sad:0, surprise:0}) \rangle$

In the same way, we can declare a Capability Situation Relationship (CSR) as,

$$CSR = \langle CE_1 \wedge CE_2 \wedge \dots \wedge CE_n \rangle \tag{3}$$

From above, we can define a complete Relationship (R_c) as a conjunction between Emotional Situation Relationship (ESR) and Capability Situation Relationship (CSR),

$$R_c = \langle ESR \wedge CSR \rangle \tag{4}$$

As a result, the generated language (*i.e in MO level*) is a lightweight language written with minimalist syntax and features (see Fig. 4b). Moreover, our system provides a perspective on representing the interaction between a “person” and “operation” in the AAL environment. As a response to an emotional state, the environment (*a person is a part of that environment*) can lead to change person’s emotion by recommending him/her or initiating additional operation. For example, when a meal is cold, person’s emotion may be changed to anger, in such case a person may re-heat that meal in the microwave, which, in turn, influence the environment configuration by initiating a new operation. However, a person is capable to execute operations op_1 and op_2 as a sequential or parallel that can be represented as

$$(\forall op_1, op_2) \text{sequential}(op_1, op_2) = (\forall t_1, t_2) \ll \text{atTime}(op_1, t_1) \wedge \text{atTime}(op_2, t_2) > \text{before}(t_1, t_2) >$$

or

$$(\forall op_1, op_2) \text{parallel}(op_1, op_2) = (\forall t_1, t_2) \ll \text{atTime}(op_1, t_1) \wedge \text{atTime}(op_2, t_2) > (t_1 = t_2) >$$

Where t_1 and t_2 are the beginning of the operations op_1 and op_2 respectively.

4.2 Modeling elements

Based on the abstracted syntax and meta-model, we briefly discuss the concrete artifacts of our modeling language. We implemented our language using ADOxx® (ADOxx 2021), a widely used metamodeling platform for developing of DSMLs. ADOxx® is very flexible to generate language in several forms like XML, RTF, HTML or ADL format. The resulted language

Query results		
(a)	(b)	Precondition
1. 08) Watch a TV on Sunday	Observed Capability	The person sits near the TV
	Watch	TV is ON, NoiseIntensity is low

Fig. 3 Infer per-conditions of “Watch a TV” (a, related observed capability and “watch” operation; b, inferred pre-conditions)

can be imported and re-used. We used also user-defined queries in ADOxx® to codify the dynamic inference rules and check syntax of mathematical computations. A sample of the proposed visual notations and the relationships used in our meta-model are depicted in Fig. 4.

4.3 Modeling consistency

HEM forces the modeler to build the right syntax of logical operators, insert consistent attributes’ values, perform a comprehensive syntax check during modeling, and allow the right connection between two or more elements. For example, when a modeler creates mistakenly a wrong relationship, then the system will throw an error message. However, the modeling system allows the modeler to build static and dynamic reasoning rules within a knowledge base for different analysis purposes. For instance, a rule for inferring the pre-condition of “watch” operation that must be fulfilled before the execution of the operation (see Fig. 3).

$$\langle \rangle \text{Operation''} > [?] \text{Name''} = \rangle \rangle \text{Watch''} [?] \text{Precondition''} = \rangle \rangle \text{TV is ON, NoiseIntensity is low''} \rangle$$

Rules are formulated using the SQL-like language AQL that may concern checking the values of attributes, the coherence of axioms, the compliance with defined rules and constraints. To construct more complex rules, the expression can be extended or combined using logical operators such as AND, OR, DIFF.

$$\begin{aligned} &\langle \rangle \text{Operation''} > [?] \text{Name''} \\ &= \rangle \rangle \text{Watch''} [?] \text{Precondition''} \\ &= \rangle \rangle \text{TV is ON, NoiseIntensity is low''} \rangle \\ &\text{AND}(\langle \rangle \text{Capability''} > [?] \text{Name''} \\ &= \rangle \rangle \text{ObservedCapability''} [?] \text{Precondition''} \\ &= \rangle \rangle \text{The person sits near the TV''} \rangle \end{aligned}$$

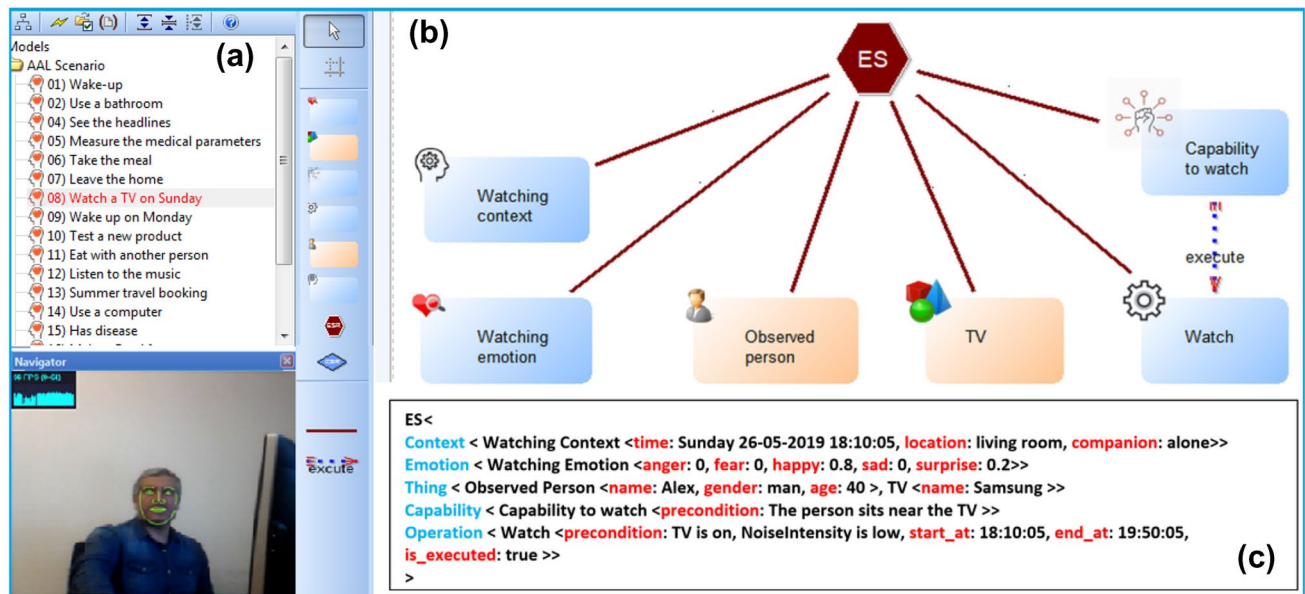


Fig. 4 HEM-L components **a** AAL scenarios selected by modeler, **b** generated HEM model by means of the ADOxx®, **c** generated Syntax by means of text-based HEM-Instance definition language

The concept of logical operators is used to combine multiple conditions. To display the pre-condition of “Watch” that is related to both capability and operation (see Fig. 3), we need to combine the two expressions using AND operator.

5 Use-case scenario implementation (running example)

To show the efficiency of our approach, in what follows, we discuss how to create the visual notations and the textual representation. For this purpose, we need to consider some scenario cases especially in the context of AAL. Each scenario includes more relevant context elements, depending on the requirements of the use case. For example, Fig. 4 depicts the elements to describe a real-world scenario “Watch a TV on Sunday”.

Moreover, system’s modeler can define different scenarios in everyday life as shown in (Fig. 4a); the representation language-based a particular situation (Fig. 4b) is automatically produced in a way that can be easily understandable; The model is generated visually (Fig. 4c) based on the scenario case and the constraints of the meta-model. Consequently, the modeling tool helps to model the human emotions and the context in a high abstraction level. In addition, the model can be exported in different machine readable formats, e.g., XML.

To generate the formal representation of the model, the modeling system parses every element in the scenario model to the corresponding description. The generated

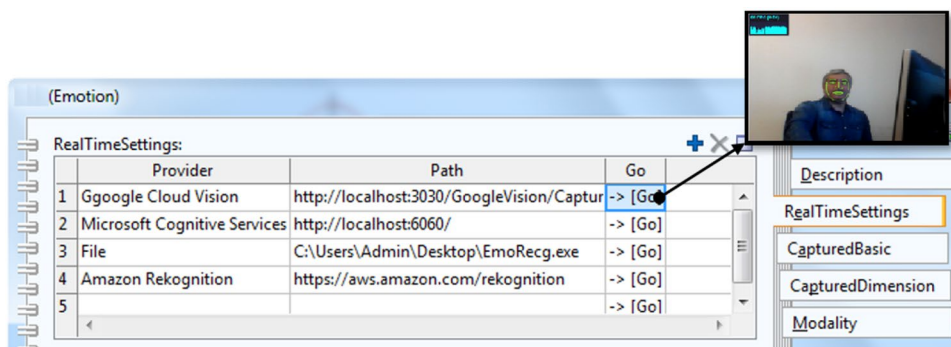
textual representation can be converted to the original modeling elements. In this model, the observed context represents variables that may be changed when the same operation is executed again and again (e.g., Watch a TV \rightarrow time, location, companion). The emotion categories can be represented in HEM as basic (Ekman 1992), dimensional (Mehrabian and Russell 1974), or a set of user-defined emotions.

In the current example, the emotion is recognized as a basic category. The observed person and the object (i.e. TV) are described as “Thing”. The capability of a person as well as the operation includes attributes: pre- and post-condition which represent any condition must be fulfilled before the execution of the operation. “Watch” is represented as an operation that includes further attribute: start-time, end-time and whether if the operation is executed or not.

6 Integrating IoT applications at runtime

To simplify the usability of IoT applications, the proposed system hides low-level implementation complexities. It enables emotion recognition systems to be integrated easily by encapsulating complex calculations and algorithms. This may help the users to focus on a domain problem without worrying about the implementation details. Besides, the system provides necessary flexibility to represent optionally emotion models such as Ekman’s basic emotion as well as complex dimensional emotion. Additional important part of emotion representation is how to fetch the results when

Fig. 5 Interface of Real-Time-Settings to integrate IoT applications



emotion is recognized. In some cases we need only to recognize facial expressions, whereas others need to combine more than one modality to analyze the emotion. In other words, the “*RealTimeSettings*” is a user-defined interface that allows domain user to insert, edit and delete ER applications (Fig. 5). The interface allows integrating different IoT input data such as video cameras, microphones, body gestures, physiological signals, etc.

However, one issue in this stage that must be addressed, is the heterogeneity of IoT data which gathered from multiple IoT sensors that return data in their own specific format. This format should be converted to a common format in order to use it inside the interface. For instance, in our experiment, we have accessed the features of emotion application interfaces APIs by processing them to offer low-level emotion categories. The available methods for getting emotion results presented in the form of JavaScript Object Notation (JSON) as an input. The interface is capable to accept captured emotions that are represented by different emotion models, for example: Basic model (Ekman 1992), 2-Dimensional (i.e. Arousal and Valence) (Kim and André 2008), 3-Dimensional (i.e. Arousal, Valence, and Dominance)(Mehrabian and Russell 1974), or a custom set of user-defined emotions.

7 Evaluation

In this section, we present an experiment to evaluate the usability and learnability of the modeling approach. Furthermore, we measure execution time regarding the generation of modeling artifacts.

Usability Evaluation The evaluation process is based on: (a) the observation of the success rate of the participants w.r.t. different tasks, and (b) the analysis of the obtained results using the System Usability Scale (SUS) (Brooke 1996), (see Appendix A in (Elkobaisi 2021)). We have omitted some proofs and supplementary results from this paper due to space limitations; these can be found as a reference in (Elkobaisi 2021). The evaluation method is organized as follows:

- *Materials* There are several available instruments to assess the usability of software systems. System Usability Scale (SUS) is one of the most popular adopted methods due to its reliability and validity (see Appendix A in (Elkobaisi 2021)).
- *Experimental participants* The sample size consisted of ten students from the University of Klagenfurt. The participants were selected independent of their programming knowledge, or experience with IoT or DSML.
- *Task description* Four tasks (see Appendix B in (Elkobaisi 2021)) were described as a natural language in the context of AAL. Participants should read and understand the tasks and visually modeling them using the modeling system.
- *Evaluation procedure* We introduced 30-minutes training, involving basic content and usage of our framework. Following the four tasks, to be graphically modeled. The participants have to execute each task and answer SUS test after completing all tasks.

Concerning the SUS calculation, we transformed the raw individual values across multiple participants into SUS score based on Brooke’s standard scoring method (Brooke 1996). However, interpreting SUS scoring can be complex. The participant score for each question is converted to new number (Brooke 2013), added together and then multiplied by 2.5 to convert the original scores of 0–40 to 0–100 (see below Eqs. 5–7).

$$X_1 = \sum_{n=1}^N (OddNumberedQuestions - 5) \quad (5)$$

$$X_2 = 25 - \sum_{n=1}^N (EvenNumberedQuestions) \quad (6)$$

$$SUS_{Score} = 2.5 * \sum (X_1 + X_2) \quad (7)$$

Table 1 Generated modeling elements based scenario

Scenario #	Expected modeling elements
S1: Making a breakfast	16
S2: Exercising	17
S3: Leaving the home	18
S4: Sleeping	19
S5: Working on computer	21

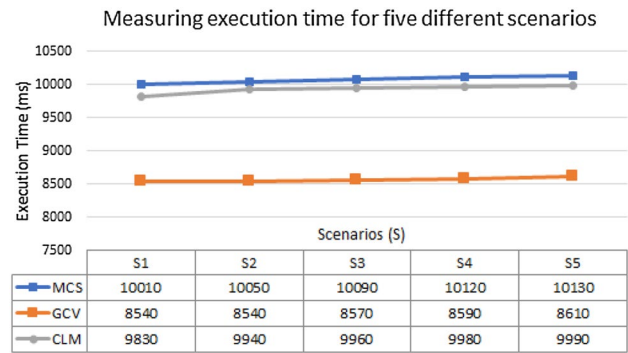
In our experiment, the total average of final scores was (90.3) which is above the average of usability expectation (68%). For reporting results, we calculated “average task times” per task for each participant, standard deviation, and confidence interval for each task. The participants performed all tasks during different time intervals. The mean SUS scores is also computed with 95% Confidence Intervals (CI) for each task. The Confidence Intervals (CI) is calculated in terms of Eq. (8),

$$CI = \bar{X} \pm Z_{\alpha/2} \times \frac{\sigma}{\sqrt{n}} \quad (8)$$

where \bar{X} is the mean, α is normally 0.05 for a 95% confidence interval, σ is the standard deviation, and n is a size of the sample. We analyzed the responses as values and applied descriptive statistics on it. We noticed a central tendency toward a positive perception of our framework. About 90% of the participants believed that the usability of the system was positive. The mean score for the “Usability” sub-scale was 90.6 and the mean score for the “Learnability” sub-scale was 88.8 (see Appendix C in (Elkobaisi 2021)).

Performance Evaluation at Runtime The previous evaluation relies on modeling different tasks that the participants should perform manually. In this section, we measure the performance of generating modeling elements (instantiation) automatically with respect to the effectiveness of IoT systems and model transformation. To achieve that, five further scenarios are presented in order to calculate execution time using various emotion recognition systems. The tested recognition systems in the evaluation were Microsoft Cognitive Services (MCS), Google Cloud Vision (GCV), and Clmtrackr (CLM). The measured runtime for instantiating the models of five different scenarios within the domain-specific tool. The selected scenarios ideally comprised of different concepts with respect to the design complexity and modeling elements (see Table 1).

The performance study was conducted in C# on hardware: Intel Core i5-2520M CPU, 2.50 GHz, 4.00 GB RAM, Windows 7 Professional 64 Bit. Obviously, the result demonstrated that the Google Cloud Vision (GCV) has lower

**Fig. 6** Performance measures: Generating modeling artefact using IoT facial recognition systems (MCS, Microsoft Cognitive Services; GCV, Google Cloud Vision and CLM, Clmtrackr)

execution time than other systems regarding automatically emotion recognition and then the generation of domain-specific modeling artifacts. This result is also compatible with study (Filestack, 2019). Moreover, the number of modeling elements has a strong impact on execution time as they required more transformation time when the number of defined modeling elements is higher. Figure 6 shows the trend of increasing execution time with respect to the growing modeling elements.

8 Discussion

In previous sections, we introduced novel artifacts to model human emotion.

The main contribution of this paper is to have a comprehensive human emotion description by combining existing IoT based recognition systems. The paper proposes tackling this problem with a meta-model and a Domain-Specific Modeling Language (DSML). Such model-based ER interface can: (1) enrich recognition with more features than the underlying standalone system, (2) help in building the components which increase self-adaptability of the users by making them able to connect to the necessary ER system without human intervention by learning their capabilities, (3) provide enhanced support for reasoning using the collected recognition data by allowing such reasoning over recognition structures and not over raw sensor data, and (4) offer an intuitive modeling tool to the relevant AAL stakeholders, e.g., people in general and their relatives, caregivers or doctors.

The system has been evaluated with ten voluntary participants, each participant has been performed four tasks. The results of the evaluation demonstrated that the approach provides the user with a suitable and practical tool for describing human emotion. Although there are

some *generic* modeling tools; these tools are not designed specifically to represent human emotion. The current approach provides more benefits such as specialized syntax and error checking in the modeling environment. The functionality of our approach can be reused as a plug-in (Elkobaisi et al. 2020) among multiple applications that leads to increase the productivity.

It is a complex process to ensure high-quality information when supporting the human based emotions during short periods of time. Therefore, the system should perform better when human is monitored over a long-period of time combined with expert knowledge. This will improve the quality of the support by allowing the practitioners to evaluate persons' emotions over a long period of time. However, emotional reaction is differing from one person to another, therefore, the system must be adjusted according to the person's preferences. This requires many details about individual profiles with personal and private data that must be achieved according to the legal and ethical requirements. In this regard, we can rely on non-visual sensors by integrating semantic similarity of word vectors with existing human activities (Machot 2020).

9 Conclusions and future work

Traditional techniques to identify emotions have focused on pure emotion analysis. Meanwhile, the situation around emotion plays a different role in representing the emotion, depending on relevance situational aspects. Recently, there is a lack of modeling methods supporting easy construction and conceptualization of human emotion related to different situations. In this paper, we proposed a DSML for modeling the emotion by analyzing the concept of domain comprising both abstract and concrete syntax. Based on meta-model constraints, we implemented a novel approach that provides practical artifacts for representing the emotion in a dynamic situation. The evaluation and validation performed using the SUS test. The result showed evidence of high adoption, usability, and learnability scores. In our experiment, the total average of SUS final scores was 90.3%. With little training, the users could easily learn how to model human emotion, intuitively, and without help. The outcome of this study demonstrates the modeling approach as an important tool that can be utilized, and it has been put into practice. As future work, we plan to do an evaluation with more complex tasks and extend our framework with a set of system-defined rules to infer complex queries in the knowledge base.

Supplementary Information The online version supplementary material available at <https://doi.org/10.1007/s12652-021-03437-w>.

Funding Open access funding provided by University of Klagenfurt. This research received no external funding.

Declarations

Conflicts of interest The authors declare no conflicts of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

- ADOxx[®] (2021) Adoxx metamodeling platform. <https://www.adoxx.org/live/home>. Accessed Jul 2021
- Alberti A (2013) Internet of things: perspectives, challenges and opportunities
- Antunes G, Bakhshandeh M, Mayer R, Borbinha J, Caetano A (2014) Using ontologies for enterprise architecture integration and analysis. *Complex Syst Inf Model* 1:1–23
- Brooke J (1996) Sus: a quick and dirty usability scale, in usability evaluation in industry, p 189–194
- Brooke J (2013) Sus: a retrospective. *J Usability Stud Arch* 8:29–40
- Khoonnaret C, Nitsuwat SA (2017) Face characteristic detection system using ontology and supervised learning. *Int J Comput Internet Manag* 25:62–69
- Cho H, Gray J, Syriani E (2012) Creating visual domain-specific modeling languages from end-user demonstration, pp 22–28
- Ekman P (1992) An argument for basic emotions. *Cogn Emotion* 6:169–200
- Elkobaisi (2021) Appendices and supplementary material of hem. <https://github.com/ELkobaisi/HEM>. Accessed Jul 2021
- Elkobaisi MR, Mayr HC, Shekhovtsov VA (2020) Conceptual human emotion modeling (HEM). In: Grossmann G, Ram S (eds) *Advances in conceptual modeling*. ER 2020. Lecture notes in computer science, vol 12584. Springer, Cham. https://doi.org/10.1007/978-3-030-65847-2_7
- Filestack (2019) Comparing image tagging services: Google vision, microsoft cognitive services, amazon rekognition and clarifai. Accessed 25 Apr 2019
- Frank U (2011) Some guidelines for the conception of domain-specific modelling languages. In: EMISA
- Kim J, André E (2008) Emotion recognition based on physiological changes in music listening. *IEEE Trans Pattern Anal Mach Intell* 30:2067–2083
- Kleppe A (2008) *Software language engineering: Creating domain-specific languages using metamodels*. Pearson Education, London
- Liao C, Lin PH, Quinlan D, Zhao Y, Shen X (2015) Enhancing domain specific language implementations through ontology
- Machot FA, Elkobaisi MR, Kyamakya K (2020) Zero-shot human activity recognition using non-visual sensors. *Sensors (Basel, Switzerland)* 2020:20

- Mehrabian A, Russell J (1974) An approach to environmental psychology. MIT Press, Cambridge
- Michael J, Mayr H (2015) Creating a domain specific modelling method for ambient assistance. In: 2015 fifteenth international conference on advances in ICT for emerging regions (ICTer), pp 119–124
- Michael J, Steinberger C, Shekhovtsov V, Machot FA, Ranasinghe S, Morak G (2018) The hbms story—past and future of an active assistance approach. *Enterp Model Inf Syst Archit Int J Concept Model* 13:345–370
- Baggia P, Bagshaw MBP (2010) Speech synthesis markup language (ssml) version 1.1, world wide web consortium
- Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Context aware computing for the internet of things: a survey. *IEEE Commun Surv Tutor* 16:414–454
- Sam K, Chatwin C (2012) Ontology-based text-mining model for social network analysis, pp 226–231
- Schröder M (2006) First suggestions for an emotion annotation and representation language
- Schröder M, Baggia P, Burkhardt F, Pelachaud C, Peter C, Zovato E (2011) Emotion markup language (emotionml) 1.0. In: w3c last call working draft
- Sykora M, Jackson T, O'Brien A, Elayan S (2013) Emotive ontology: extracting fine-grained emotions from terse, informal messages
- Terkaj W, Pedrielli G, Sacco M (2012) Virtual factory data model
- Walter T, Parreiras FS, Staab S (2012) An ontology-based framework for domain-specific modeling. *Softw Syst Model* 13:83–108
- Zarrin B, Baumeister H (2018) An integrated framework to specify domain-specific modeling languages. In: *MODELSWARD*

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.