Noname manuscript No.
(will be inserted by the editor)

Human Emotion Modeling (HEM): An Interface for IoT Systems

Received: date / Accepted: date

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 care-givers 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.

Address(es) of author(s) should be given

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; Dey, 2001). 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; Chatzigiannakis et al., 2012). 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 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 (see Fig. 1). 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 (***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). Furthermore, the proposed approach evaluated using several datasets which include multiple IoT emotion data sources, for instance, biometric data (*blood pressure, perspiration, breathing rate, etc.*).

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 metamodel, 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, Section 8 and 9 present the discussion, conclusions, and future work. Furthermor, a list of appendices is presented at the end of this paper.

[1] https://azure.microsoft.com/en-us/services/cognitive-services
[2] https://cloud.google.com/vision
[3] https://github.com/auduno/clmtrackr

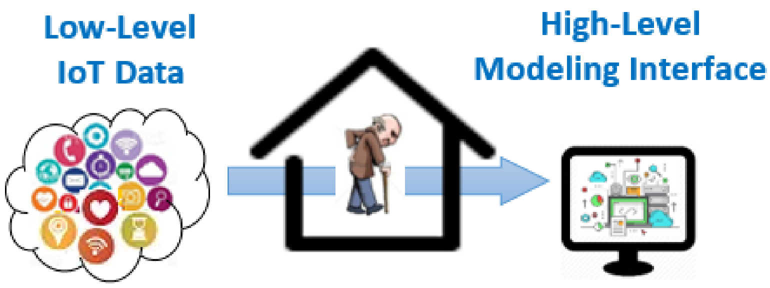

Fig. 1 Core concept of the modeling approach

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 representation languages (Schröder et al., 2011; Schröder, 2006; Prendinger and Ishizuka, 2004; Froumentin, 2004; Marriott and Stallo, 2002; P. Baggia, 2010) 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, (see Section 2), 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) (OMG®, 2021) and a systematic procedure like presented in (Michael and Mayr, 2015) to identify the structure and appearance 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 (see Section 4.3).

Domain-specific modeling languages (DSMLs) *"are specialized languages for a particular application area, which use the concepts and notations established in the field. They allow domain experts, who are usually non-programmers, to directly employ their domain knowledge about what a system under development should do"* (Zarrin and Baumeister, 2018). Therefore, the use of DSMLs has gained increasing popularity (Mayr et al., 2016) as they contribute supporting the productivity of modeling, help to increase model quality and comprehensibility, and come with graphical notations that are easy to understand by the users in a certain domain.

However, this work has been carried out within the framework of the ongoing 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 system behavior generation and reasoning. EmotionML can be described in terms of Ekman's basic emotion 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 oredr to represent emotion in technological contexts. EARL represents emotions as basic, dimensional, or sets of appraisal scale. It used also to generate emotional agent such as embodied conversational agents (ECAs) (Prendinger and Ishizuka, 2004). **(EMMA)** *Extensible MultiModal Annotation* (Froumentin, 2004) is a markup language used to represent multi-modal user inputs (e.g., speech, pen, keystroke, or gesture) in a standardised way for further processing. **(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* (P. Baggia, 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; Blackburn and Denno, 2015). Studies (Walter et al., 2012; Liao et al., 2015) explained how to integrate domain-specific languages with ontology languages and automated reasoning services at the meta-model level. The use of the formal semantics of the *Web Ontology Language (OWL)* [4] 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; García-Rojas et al., 2006; C. Khoonnaret, 2017; Francisco et al., 2007) 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 development of our modeling language and method is driven by *Meta Object Facility OMG*[5](OMG®, 2021). 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-L on three levels of language definition hierarchy introduced (Fig. 4):

– The meta-model for HEM-L is defined on M2 level by means of the ADOxx® metamodeling framework (ADOxx®, 2021b), and is supported in the HEM-L Modeling Tool generated by ADOxx® based on this meta-model.
– The modelers are able to create the HEM models by means of this tool using HEM-L 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-I *Instance* definition language optimized for executive data exchange.

[4] https://www.w3.org/TR/owl-features
[5] https://www.omg.org/mof

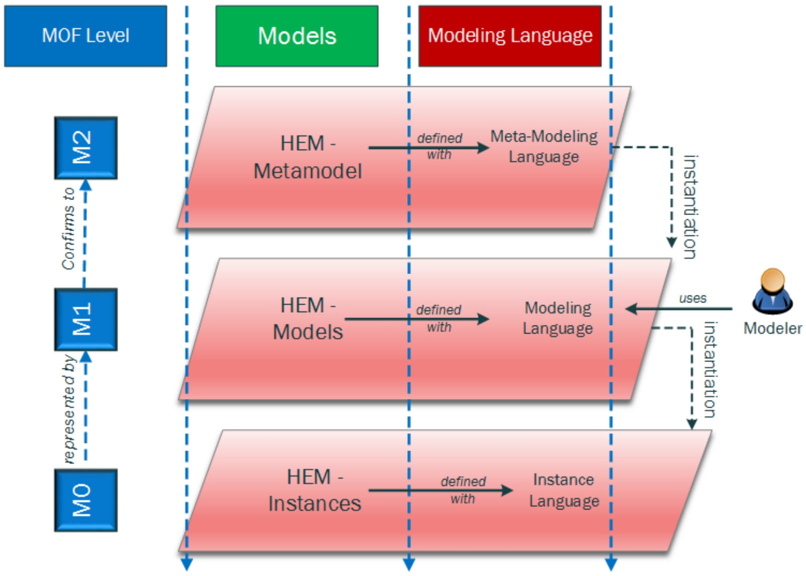

Fig. 2 Human Emotion Modeling (HEM) hierarchy (M2: Meta-model level, M1: Model level, and M0: Instance level)

To design HEM-L, there is a demand to support the design process with a suitable Domain-Specific Modeling Method (DSMM) as it uses domain concepts directly in a higher level of abstraction. To issue the characteristics of our language, we embedded it with DSMM (Frank, 2011; Michael and Mayr, 2015). Such modeling method supports five main phases: preparation, modeling language, modeling process, modeling tool, and evaluation. Each phase consists of several steps, for mit uses domain concepts directly in a higher level of abstraction. To issue the characteristics of our language, we embedded it with DSMM (Frank, 2011; Michael and Mayr, 2015). Such modeling method supports five main phases: preparation, modeling language, modeling process, modeling tool, and evaluation. Each phase consists of several further steps.

Generally, the design of Domain Specific Modeling Language (DSML) includes at least three main aspects (Kleppe, 2008; Cho et al., 2012): 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

Conceptualizing a meta-model facilitates analyzing the complexity of the real world; it is used as bases for defining our modeling system. In our case, a

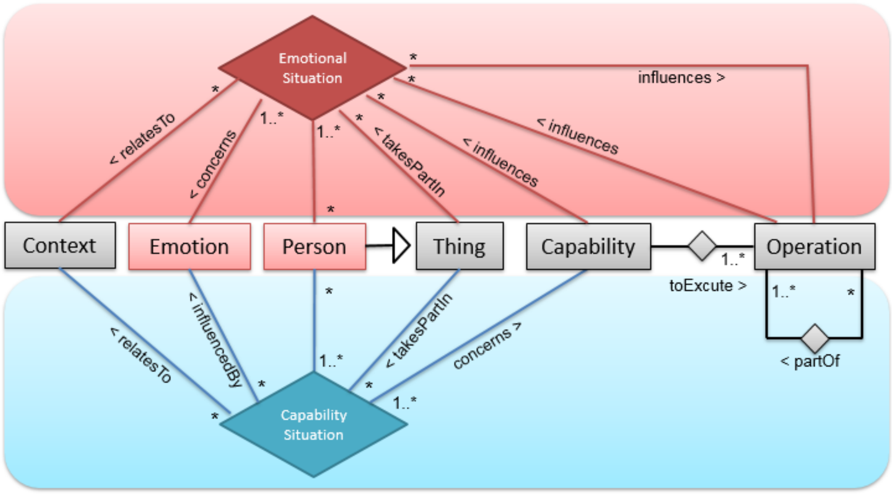

Fig. 3 A meta-model of proposed modeling system (HEM)

meta-model is divided into two symmetric parts (see Fig. 4.1). The upper part shows Emotional-Situation Relationship (dyed with red color) and the lower part depicts Capability-Situation Relationship (dyed with blue color). The core concepts/elements are shared between these two parts. We involved Capability-Situation Relationship in our meta-model for two reasons: a) our target is AAL, in this domain the elderly/handicapped are sometimes incapable to perform some operations/activities. b) a capability is more related to the emotion e.g., when the elderly are incapable to do an operation this may influence the emotional state. In the meta-model, a context represents variables that are related to the current emotional situation and may change when the same operation is performed repeatedly (*e.g., time, weather, location, companion, occasions, etc.*). In our meta-model when an operation changes, the emotional situation may also change and vice versa (e.g., a person has a negative emotion; the system starts slow music (*i.e. operation*) which may influence current emotional state).

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.

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

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.*

From the above example, we observe that the situation may be different from one context to another (in the "Watching movie" scenario e.g., a companion may be $\longrightarrow$ with partner, kids, or alone). Similarly, the situation is changed when a person is watching a horror movie at home and the online streaming gets interrupted, as a consequence, the person will feel angry.

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

$$ESR = < CE_1 \ \wedge \ CE_2 \ \wedge \ \ldots \wedge \ CE_n > \tag{2}$$

Then, ESR = < "Context"= (time: weekend, location: cinema, companion: girlfriend) ∧ "Emotion"=(anger:0, fear:0, happy:1, sad:0, surprise:0) >

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

$$CSR = < CE_1 \ \wedge \ CE_2 \ \wedge \ \ldots \wedge \ CE_n > \tag{3}$$

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

$$R_c = < \ ESR \ \ \wedge \ CSR > \tag{4}$$

As a result, the generated language (*i.e in M0 level*) is a lightweight language written with minimalist syntax and features (see Fig 5b). 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. In other words, environment-state and emotion are the preconditions, as well as the post-condition. In the previous example, the constraint (*i.e. preconditions*) of the "Watch" operation is "*a TV is ON*". However, a person is capable to execute operations op_1 and op_2 as a sequential or parallel that can be represented as

$$(\forall op_1 \ , op_2) \, sequential \, (op_1 \ , op_2) = (\forall \ t_1, \ t_2) << atTime \ (op_1, \ t_1) \ \wedge \\ atTime \ (op_2, \ t_2) > \ before \ (t_1 \ , \ t_2) >$$

or

$$(\forall op_1 \ , op_2) \ \ parallel \ \ (op_1 \ , op_2) = (\forall \ t_1, \ t_2) \ \ << atTime \ (op_1, \ t_1) \ \wedge \\ atTime \ (op_2, \ t_2) > \ \ (t_1 \ = \ t_2) >$$

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

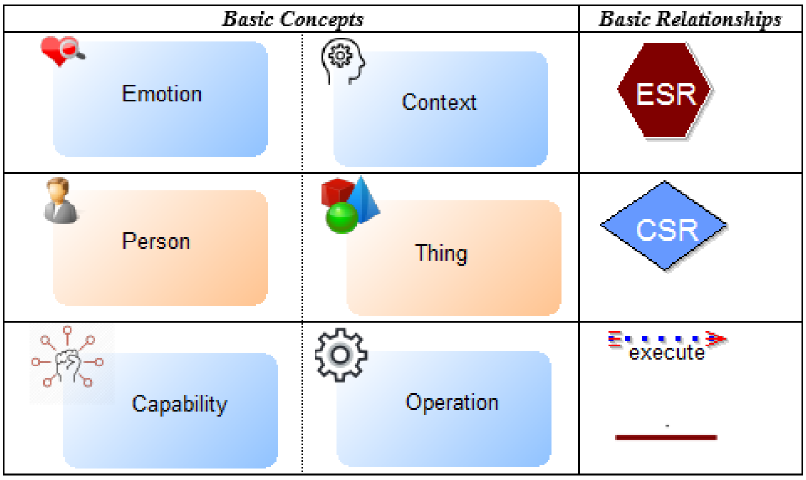

Fig. 4 Concepts of proposed meta-model *(ESR: Emotional Situation Relationship and CSR: Capability Situation Relationship)*

4.2 Modeling Elements

Based on the abstracted syntax and meta-model, we briefly discuss the concrete artifacts of our modeling language (see Fig 4.2). We implemented our language using ADOxx® (ADOxx®, 2021b), 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 (ADOxx®, 2021a) format. The resulted language 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. The visual notations of basic concepts and the relationships used in our meta-model are depicted in Fig. 4.2.

4.3 Modeling Consistency

Our system constraints modeling according to the syntax defined by the modeling language (*e.g., meta-model cardinality*). It 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

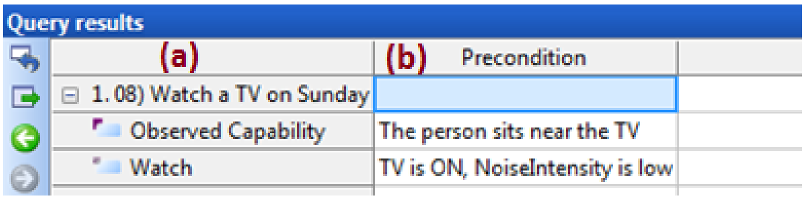

Fig. 5 Infer per-conditions of "Watch a TV" ((a) related observed capability and "watch" operation, (b) inferred pre-conditions)

be fulfilled before the execution of the operation.

$$(<\text{"}Operation\text{"}>[?\text{"}Name\text{"}=\text{"}Watch\text{"}]$$
$$[?\text{"}Precondition\text{"}=\text{"}TV\ is\ ON, NoiseIntensity\ is\ low\text{"}])$$

Rules are formulated using the SQL-like language AQL[6] 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. 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. 4.3), we need to combine the two expressions using AND operator.

$$(<\text{"}Operation\text{"}>[?\text{"}Name\text{"}=\text{"}Watch\text{"}][?\text{"}Precondition\text{"}=\text{"}TV\ is\ ON,$$
$$NoiseIntensityislow\text{"}])AND(<\text{"}Capability\text{"}>[?\text{"}Name\text{"}=$$
$$\text{"}ObservedCapability\text{"}][?\text{"}Precondition\text{"}=\text{"}The\,person\,sits\,neat\,the\,TV\text{"}])$$

As a type of constraint, the system recommends a person to regulate his/her emotion based on the current emotional state. In this case, the system may constraint the ambient according to the person's preferences. For example, when a person has a negative emotion, the system may suggest some operations through the settings of music, color, and light/or ask the person to do physical operations such as: walk, talk with Alexa[7], watch a movie, or drink fresh juice. The emotion regulation mechanisms are implemented to provide the best-suited conditions to attain the desired emotion in order to enhance the quality of life to the person (Domaradzka and Fajkowska, 2018; Gross, 2002).

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,

[6] https://www.adoxx.org/live/adoxx-query-language-aql
[7] https://alexa.amazon.com

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. 5 depicts the entities 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 5a); The representation language-based a particular situation (Fig 5b) is automatically produced in a way that can be easily understandable; The model is generated visually (Fig 5c) 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* $\longrightarrow$ *time, location, companion*). As we explained before, the categories of emotion can be 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 with related values. The observed person and the object (*i.e. TV*) are described as "Thing". The observed capability of a person as well as the operation includes optional attributes: pre- and post-condition that represent any condition must be fulfilled before the execution of the operation. Some per- and post-conditions belong to the capability and others belong to the operation. In our example, the pre-condition is related to the capability of the elderly "*The person sits near the TV*" and pre-condition is related to the operation "*The television is ON and NoiseIntensity is low*". "Watch" is represented as an operation that includes further attribute: start-time, end-time and whether if the operation is executed or not.

The resulted intuitive language (see Fig 5b) is created in such a way to eliminate extra syntax and the complexity. This representation seems readable and understandable. The goal of the system is not only to read out the abstract representation by non-technical persons, but also to reduce the misunderstanding between domain experts and everyone else. Intuitive language enables non-technical people to perform tasks using familiar words and phrases, with minimal training.

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 by encapsulating complex calculations and algorithms. This

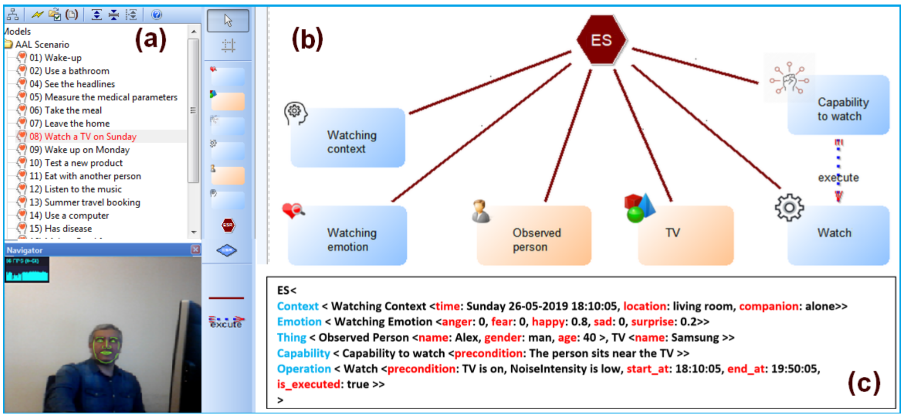

Fig. 6 HEM-L components (a) AAL scenarios selected by modeler, (b) generated HEM model by means of the ADOxx®, (c) generated Syntax by means of text-based HEM-I Instance definition language

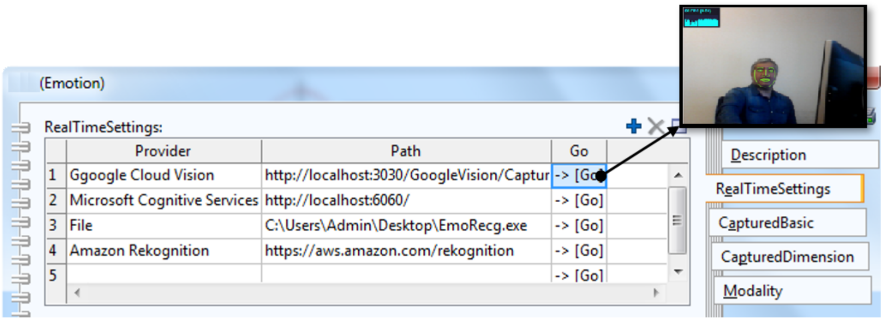

Fig. 7 Interface of Real-Time-Settings to integrate IoT applications

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 (Ekman, 1992) as well as complex dimensional emotion (see Fig. 5). Additional important part of emotion representation is how to fetch the results when 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 (see 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

System modelers have different conceptual perspectives than users because the developers are concerned with a problem for a long period, this makes them routine-blinded. Therefore, it is necessary to frequently examine a system with "*real users*". In this section, we present an experiment to evaluate the usability and learnability of the modeling approach. Furthermore, we measure execution time regarding the automatic generation of modeling artifacts.

Usability Evaluation. The evaluation of the proposed system is based on: a) the observation of the success rate of the participants w.r.t. different tasks, (*see Appendix B*) and b) the analysis of the obtained results using the System Usability Scale (SUS) (Brooke, 1996), (*see Appendix A*).

We evaluated whether the modeling artifacts were easy to use and easy to learn. We examined the usability by allowing the participants, independent of their ability or programming knowledge to model an emotional situation. Furthermore, we investigated how the participants were able to understand all the information and interact with the system sufficiently.

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 (Tullis and Albert, 2013; Sauro and Lewis, 2012) due to its reliability and validity (Bangor et al., 2008; Kirakowski, 1994), shortness (consists of 10 questions), comprehensiveness and robust even with a small sample size (10∼ 12 users) (Tullis and Stetson, 2004). The standard version of SUS consists of 10-items that can be rated on a five-point (Likert-type) scale ranging from 'strongly disagree' to 'strongly agree' (see Appendix A).
- **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) 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.

All participants have passed the tasks except one participant did not complete Task4 correctly and another participant failed to completely perform Task3 and Task4. This result was expected due to Task4 is slightly more complex than other tasks as explained in the feedback of the task description list (Appendix B), therefore, a smaller success rate was more likely. The modeling elements of Task4 are depicted visually in Fig.5c. 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 equations 1, 2 and 3).

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

$$X_2 = 25 - \sum_{n=1}^{N} (EvenNumberedQuestions) \tag{2}$$

$$SUS_{Score} = 2.5 * \sum (X_1 + X_2) \tag{3}$$

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 which is shown across the bottom (see Table 2 in Appendix C). The participants performed all tasks during different time intervals.

Graphically, Fig. 8 shows the mean SUS scores with 95% Confidence Intervals (CI) for each task. The Confidence Intervals (CI) is calculated in terms of equation 4,

$$CI = \bar{X} \pm Z_{\alpha/2} \times \frac{\sigma}{\sqrt{(n)}} \tag{4}$$

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.

As observed in Fig. 8, the meantime increased gradually for T1, T2, T3, and T4 respectively. This result is consistent with the feedback given to the participants in task description list (see Appendix B).

The two Items (#4 and #10 in Appendix A) provide the learnability dimension and the other eight items provide the usability dimension.

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. Therefore, the observation and SUS results show evidence of excellent usability. Without much training, participants could easily learn how to model, intuitively, and without help. The participants felt that the modeling system performed well, and the overall SUS score was 90.3, an "Excellent" result based

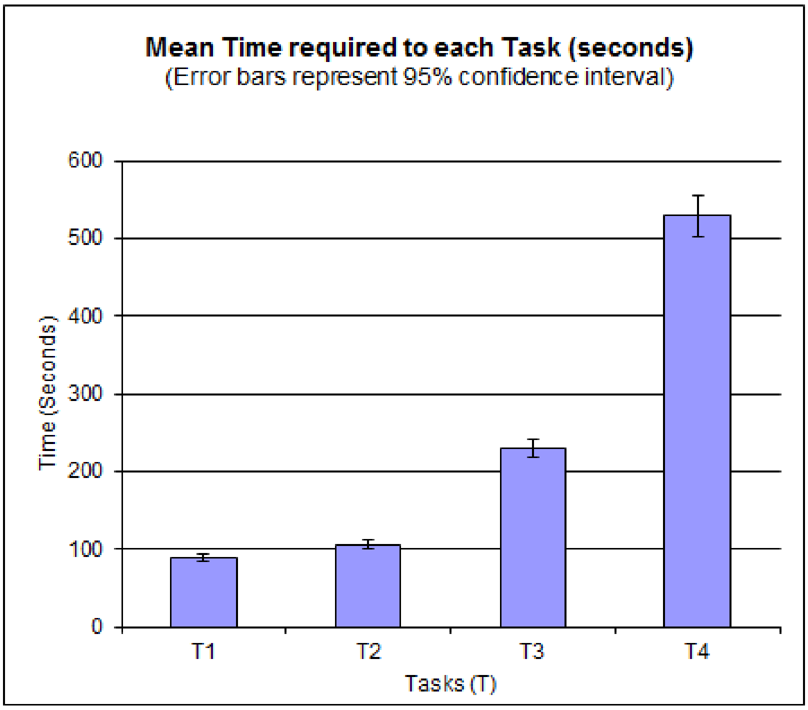

Fig. 8 Mean time and confidence interval per task with Error bars represent the 95% CI

on standard SUS (McLellan et al., 2012). The mean score for the "Usability" sub-scale was 90.6 and the mean score for the "Learnability" sub-scale was 88.8 (see Fig. 11 in Appendix C).

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 is shown in Fig. 9. The selected scenarios ideally comprised of different concepts with respect to the design complexity and modeling elements (see Table 1). However, the number of modeling elements (*i.e. concepts and relationships between concepts*) depends on the situation, clearly, the same situation may include different modeling elements depending on the context. 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 Google Cloud Vision (**GCV**) has lower execution time than other systems regarding automatically emotion

Table 1 Generated modeling artifacts based scenario

Scenario #	**Generated modeling artifacts**
S1: Making a Breakfast	16
S2: Exercising	17
S3: Leaving the home	18
S4: Sleeping	19
S5: Working on computer	21

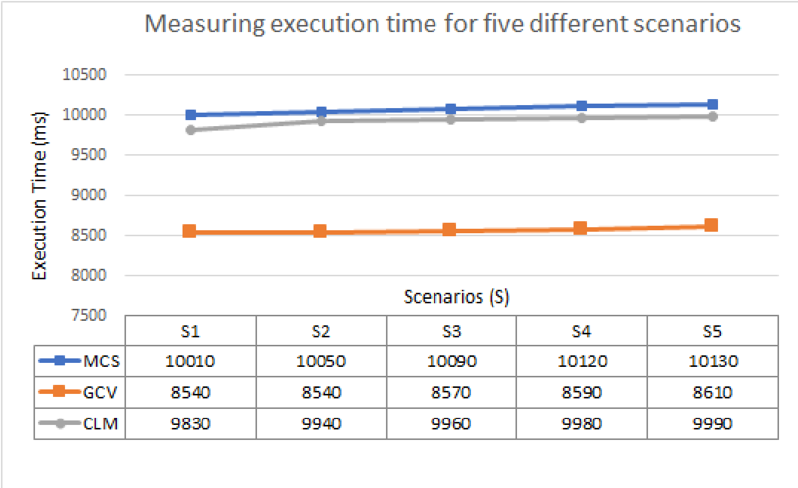

Fig. 9 Performance measures: Generating modeling artefact using IoT facial recognition systems ((MCS): Microsoft Cognitive Services, (GCV): Google Cloud Vision , and (CLM): Clmtrackr)

recognition and then the generation of domain-specific modeling artifacts. This result is also compatible with study (Filestack, 2019) that comparing features of popular recognition APIs. 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. Fig. 9 shows the trend of increasing execution time with respect to the growing modeling elements. However, productivity is also improved when comparing the difference between manual and automatic modeling time. Even for the largest execution time, productivity is increased significantly.

8 Discussion

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

However, 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. Additionally, the automatic generation of the formal textual representation reduces manual coding that enhances modeling efficiency and reduces the number of errors. 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. Finally, based on a person's emotion, the system offers the best-tailored operation to regulate negative emotions towards a positive when possible. In order to provide the best-suited conditions produced in the form of music, color, lighting, or physical activities to attain a target regular emotion.

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 et al., 2020).

Furthermore, to the best of our knowledge, there is a lack in annotation tools that combine physiological data with contextual information for AAL environments. Therefore, one of the possible use-cases that the proposed approach can be used for annotating training data to train machine learning models. As a result, machine learning models might be trained with respect to contextual data. Consequently, they can learn to predict emotion w.r.t. contextual information. This offers great potential to improve the overall emotion recognition performance.

Moreover, the usability test has been used to prove that the modeling language is suited for technical/non-technical users in terms of simplicity and real-life Smart Home IoT environments. The experiment result was very promising and received a high usability score. Furthermore, to generate visual modeling elements, we integrated IoT recognition systems with five different scenarios. Based on these scenarios as well as the performance of the emotion recognition system, the corresponding modeling artifacts are generated. In addition, the influence on productivity was also examined by measuring the time of manual and automatic modeling using our approach. Automatic modeling implementation would take less execution time than the manual method.

Ontologies provide high reasoning capabilities and knowledge inference. We therefore plan to provide, a comprehensive hybrid system, where the *conceptual data models* and *ontology components* are cooperated. In this way, the ontology description is relied on conceptual data models, while ontological consistency and incomplete information processing are carried out using logic reasoners. However, implementing such step will be a target for future research.

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. Furthermore, we plan to integrate the suggested system as a lightweight, connectable framework with other available platforms (like HCM-L (Mayr et al., 2016)) to share its implementation functionality. This may improve the recall, precision of both systems' algorithms by providing real-time emotion notifications.

Funding

This research received no external funding

Acknowledgements

We are very grateful to professor Heinrich C. Mayr for his support in creating the underlined meta-model. We also thank the participants who volunteered to take part in the experiment, and thereby helped make this study possible.

Conflict of interest

The authors declare no conflicts of interest. Authors ensure that there are no personal circumstances, interests, or sponsors that may be perceived as inappropriately influencing the representation or interpretation of reported research results.

A "Appendix A"

The standard version of System Usability Scale (SUS) used in the study (see Fig. 10)

	The System Usability Scale Standard Version		**Strongly Disagree** 1	2	3	4	**Strongly Agree** 5
1	I think that I would like to use this system frequently.		O	O	O	O	O
2	I found the system unnecessarily complex.		O	O	O	O	O
3	I thought the system was easy to use.		O	O	O	O	O
4	I think that I would need the support of a technical person to be able to use this system.		O	O	O	O	O
5	I found the various functions in this system were well integrated.		O	O	O	O	O
6	I thought there was too much inconsistency in this system.		O	O	O	O	O
7	I would imagine that most people would learn to use this system very quickly.		O	O	O	O	O
8	I found the system very awkward to use.		O	O	O	O	O
9	I felt very confident using the system.		O	O	O	O	O
10	I needed to learn a lot of things before I could get going with this system.		O	O	O	O	O

Fig. 10 The standard version of System Usability Scale (SUS) used in the study

B "Appendix B"

TASK LIST

Name: recognize emotion
Description: starting from emotion element, the user has ability to capture basic /dimensional emotions.
He/she is capable to define new emotion recognition provider in real-time settings.
Objective: *capture "basic emotion"*
Feedback: 4 steps – easy; Max time interval: 120 (second)

Name: create a simple relationship
Description: starting from element according to the task's requirements, the user has to know how to create a relationship between two or more elements correctly.
Objective: *create a relationship between emotion and person*
Feedback: 5 steps – easy; Max time interval: 180 (second)

Name: add context element
Description: starting from context element, user has to understand what the meaning of context. The context represents variables that may change when the same operation executed repeatedly (e.g., Watch a TV ⟶time, location, companion). He/she has to add contextual attributes.
Objective: *add context element, then add its attributes (Time, Location, Companion) with related values (Weekend, Living room, Partner) respectively.*
Feedback: 8 steps – medium; Max time interval: 300 (second)

Name: create a complex relationship with specific context.
Description: starting from context element, user has to understand how to create other elements depending on the suggested scenario. Other
elements such as "Thing" (e.g., TV), "Capability" of a person that includes optional attributes: pre- and post-condition which represent any condition must be fulfilled before the "Operation" being executed (e.g., sunny weather).
Objective: *Add new modelling elements according to the context "Watch a TV on Sunday at living room", and then connect each element with a suitable connector.*
In this task,

- The precondition of "Capability" is "the television must be on".
- "Watch" is represented as an operation that includes attribute: star-time, end-time and whether if the "Operation" is executed or not.

Feedback: 17 steps – hard; Max time interval: 600 (second)

For further feedback, comments and suggestions, please write below:
....
....
....

C "Appendix C"

Table 2 Times (in seconds) for 10 participants and 4 tasks

	T1	**T2**	**T3**	**T4**
P1	71	97	248	463
P2	106	88	211	576
P3	81	115	197	492
P4	84	124	217	489
P5	98	92	287	519
P6	91	116	200	528
P7	84	89	214	496
P8	75	74	213	525
P9	98	126	217	600
P10	101	140	300	600
Mean:	88.9	106.1	230.4	528.8
Std Dev:	11.7	21.0	36.1	48.1
N:	10	10	10	10

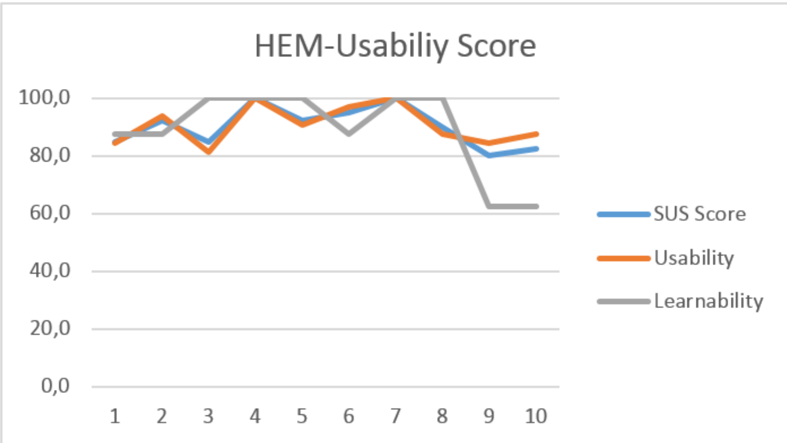

Fig. 11 SUS score with relevant Subscales

References

ADOxx® (2021a) Adoxx definition language (adl). URL `https://www.adoxx.org/live/adoxx-development-languages-syntax-support`, last accessed April 2021

ADOxx® (2021b) Adoxx metamodelling platform. URL `https://www.adoxx.org/live/home`, last accessed July 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 Informatics Model Q 1:1–23

Bangor A, Kortum PT, Miller JT (2008) An empirical evaluation of the system usability scale. International Journal of Human–Computer Interaction 24:574 – 594

Blackburn M, Denno PO (2015) Using semantic web technologies for integrating domain specific modeling and analytical tools. In: Complex Adaptive Systems

Brooke J (1996) Sus: A quick and dirty usability scale, in usability evaluation in industry. p 189–194

Brooke J (2013) Sus: a retrospective. Journal of Usability Studies archive 8:29–40

C Khoonnaret SAN (2017) Face characteristic detection system using ontology and supervised learning. International Journal of the Computer, the Internet and Management 25:62–69

Chatzigiannakis I, Hasemann H, Karnstedt M, Kleine O, Kröller A, Leggieri M, Pfisterer D, Römer K, Truong C (2012) True self-configuration for the iot. 2012 3rd IEEE International Conference on the Internet of Things pp 9–15

Cho H, Gray J, Syriani E (2012) Creating visual domain-specific modeling languages from end-user demonstration. 2012 4th International Workshop on Modeling in Software Engineering (MISE) pp 22–28

Dey A (2001) Understanding and using context. Personal and Ubiquitous Computing 5:4–7

Domaradzka E, Fajkowska M (2018) Cognitive emotion regulation strategies in anxiety and depression understood as types of personality. Frontiers in Psychology 9

Ekman P (1992) An argument for basic emotions. Cognition & Emotion 6:169–200

Elkobaisi MR, Mayr H, Shekhovtsov V (2020) Conceptual human emotion modeling (hem). In: ER

Filestack (2019) Comparing image tagging services: Google vision, microsoft cognitive services, amazon rekognition and clarifai. URL https://blog.filestack.com/thoughts-and-knowledge/comparing-google-vision-microsoft-cognitive-amazon-rekognition-clarifai, last accessed: April 25 (2019)

Francisco V, Gervás P, Peinado F (2007) Ontological reasoning to configure emotional voice synthesis. In: RR

Frank U (2011) Some guidelines for the conception of domain-specific modelling languages. In: EMISA

Froumentin M (2004) Extensible multimodal annotation markup language (emma): invited talk

García-Rojas A, Vexo F, Thalmann D, Raouzaiou A, Karpouzis K, Kollias S, Moccozet L, Magnenat-Thalmann N (2006) Emotional face expression profiles supported by virtual human ontology. Computer Animation and Virtual Worlds 17

Gross J (2002) Emotion regulation: affective, cognitive, and social consequences. Psychophysiology 39 3:281–91

Kim J, André E (2008) Emotion recognition based on physiological changes in music listening. IEEE Transactions on Pattern Analysis and Machine Intelligence 30:2067–2083

Kirakowski J (1994) The use of questionnaire methods for usability assessment

Kleppe A (2008) Software language engineering: Creating domain-specific languages using metamodels

Liao C, Lin PH, Quinlan D, Zhao Y, Shen X (2015) Enhancing domain specific language implementations through ontology. Proceedings of the 5th International Workshop on Domain-Specific Languages and High-Level Frameworks for High Performance Computing

Machot FA, Elkobaisi MR, Kyamakya K (2020) Zero-shot human activity recognition using non-visual sensors. Sensors (Basel, Switzerland) 20

Marriott A, Stallo J (2002) Vhml – uncertainties and problems . a discussion

Mayr H, Machot FA, Michael J, Morak G, Ranasinghe S, Shekhovtsov V, Steinberger C (2016) Hcm-l: Domain-specific modeling for active and assisted living. In: Domain-Specific Conceptual Modeling

McLellan S, Muddimer A, Peres S (2012) The effect of experience on system usability scale ratings. Journal of Usability Studies archive 7:56–67

Mehrabian A, Russell J (1974) An approach to environmental psychology
Michael J, Mayr H (2015) Creating a domain specific modelling method for ambient assistance. 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
OMG® (2021) Object management group: Meta object facility (MOF)™specification. URL https://www.omg.org/mof, last accessed July 2021
P Baggia MB P Bagshaw (2010) Speech synthesis markup language (ssml) version 1.1, world wide web consortium, recommendation rec-speechsynthesis11-20100907
Perera C, Zaslavsky A, Christen P, Georgakopoulos D (2014) Context aware computing for the internet of things: A survey. IEEE Communications Surveys & Tutorials 16:414–454
Prendinger H, Ishizuka M (2004) Life-like characters - tools, affective functions, and applications. In: Life-like characters
Sam K, Chatwin C (2012) Ontology-based text-mining model for social network analysis. 2012 IEEE International Conference on Management of Innovation & Technology (ICMIT) pp 226–231
Sauro J, Lewis JR (2012) Quantifying the user experience: Practical statistics for user research
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. 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
Tullis T, Albert W (2013) Measuring the user experience, second edition: Collecting, analyzing, and presenting usability metrics
Tullis T, Stetson JN (2004) A comparison of questionnaires for assessing website usability
Walter T, Parreiras FS, Staab S (2012) An ontology-based framework for domain-specific modeling. Software & Systems Modeling 13:83–108
Zarrin B, Baumeister H (2018) An integrated framework to specify domain-specific modeling languages. In: MODELSWARD