TASC4RE: a data-driven context model to support the elicitation of context-aware functionalities

Rodrigo Falcão[1], Maria Clara Pestana[2] and Vaninha Vieira[3]

[1]*Fraunhofer IESE, Fraunhofer-Platz 1, Kaiserslautern, 67663, Germany*
[2]*Computer Science Graduate Program, UFBA, Av. Milton Santos s/n, Salvador, 40170-110, Brazil*
[3]*Institute of Computing, UFBA, Av. Milton Santos s/n, Salvador, 40170-110, Brazil*

Abstract

In requirements elicitation for context-aware systems, context modeling is an essential early step. However, it has been overlooked by practitioners due to its perceived high complexity, in particular when it comes to the analysis of how different contexts may influence a user task of interest. To change this situation, data-driven context modeling approaches appear to be promising: Based on contextual data, algorithms can identify how contextual elements may, in combination with each other, influence user tasks. In this paper, we contribute the Task-specific Context Model for Requirements Elicitation (TASC4RE). TASC4RE is a context model representation aimed at supporting data-driven context modeling approaches. It combines the expressiveness of natural language, which is appreciated by human readers, with formality. TASC4RE was preliminarily evaluated in a controlled experiment with professionals in the context of a software development project. The results showed a positive trend towards the acceptance of the model.

Keywords

Context modeling, context awareness, data-driven, requirements engineering

1. Introduction

Almost three decades have passed since Schilit et al. coined the term "context-aware computing" [1], and much research has been conducted in the field of context modeling since then (as summarized in several secondary studies, e.g., [2] [3] [4] [5]). Just as there are several types of context models, their applicability is also diverse.

In requirements elicitation for context-aware systems, context modeling is an early step that requires the identification of contextual elements for the application domain [6] [7] [8], technical understanding regarding which of them are accessible [9] [10], evaluation of which contextual elements are relevant for a given user task [6] [9] [11] [12], and analysis of meaningful combinations of contextual elements [5] [9] [10] [13]. As a result, the context modeling process is expected to produce a context model that expresses how the context influences user tasks, so that requirements engineers can benefit from it to identify and describe novel context-aware functionalities.

ER'2022 Forum and PhD Symposium.
rodrigo.falcao@iese.fraunhofer.de (R. Falcão)
0000-0003-1222-0046 (R. Falcão); 0000-0001-8382-7844 (M. C. Pestana)

In practice, however, context models have not been of much help for requirements engineers. A recent survey [14] showed that when it comes to support for requirements elicitation, context models have been rarely used, and when they are, the benefit provided by them is generally low. One reason presented in the survey is that context modeling activities are perceived by practitioners as very complex, time-consuming, and non-intuitive. In a scenario with dozens of contextual elements, the number of potential combinations of contextual elements is too high to be analyzed manually. Without the analysis of relevance and combinations, context models are limited to declaring *what the context is*; the next and fundamental step, understanding *how the context influences user tasks*, has been left to the experience of professionals.

In order to overcome the complexity of context modeling for requirements elicitation, data-driven context modeling approaches have been considered adequate [15] [16]. Based on contextual data, such mechanisms aim to identify meaningful combinations of contextual elements that may influence a user task of interest. In our research, we implemented a data-driven context modeling process, which leads to a structured list of contexts found to influence a targeted user task. Although data-driven context modeling processes can be implemented using different techniques, these techniques have in common the analysis of large amounts of contextual data in order to discover relevant contexts that might influence user tasks of interest. The question is: Independent of the technique used – how can the outcome of the data analysis be represented? It must be translated into a context model from which requirements engineers can benefit to elicit context-aware functionalities. In other words, the outcome of data-driven context modeling techniques must be converted into a format that properly supports the comprehension of task-relevant contexts, so that humans can get insights from it in order to creatively describe context-aware functionalities.

In this paper, we contribute the *Task-specific Context Model for Requirements Elicitation* (TASC4RE). TASC4RE is a context model representation designed to be used in data-driven context modeling approaches that search for meaningful combinations of contextual elements. It combines the formality required for integration into automated context modeling approaches and the expressiveness of natural language appreciated by human readers. We used TASC4RE to support our data-driven context modeling process and evaluated it preliminarily in a software development project at Fraunhofer IESE. The results showed a positive trend towards acceptance of the model [17].

The remainder of this paper is organized as follows: Section 2 provides the research background; Section 3 presents related work; Section 4 introduces the syntax and semantics of TASC4RE; Section 5 reproduces its evaluation; Section 6 discusses the initial findings; and Section 7 concludes the paper.

2. Background

2.1. Definitions

Since there exist a rather large number of definitions of context and its related concepts such as entities, contextual elements, situations, and relationships, we consider it important to make it clear what we mean when we refer to the terms *contextual element*, *context*, and *combination.*

A widely accepted definition of context was proposed by Dey [18], which describes context and entity as follows: *"Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves"*.

Vieira [6] makes a distinction between context and contextual elements (CE): *"A contextual element (CE) is any piece of data or information that enables to characterize an entity in a domain."*, whereas *"The context of an interaction between an agent and an application, in order to execute some task, is the set of instantiated contextual elements that are necessary to support the task at hand."*.

We slightly adapt Vieira's definition to state that *context is a set of instantiated contextual elements that influences the task at hand*, because from our point of view, context can be used not only to enable user tasks (necessary), but also to improve them (influence). The set of instantiated contextual elements can have one or more contextual elements. In the latter case, we say that the context is composed of a *combination* of contextual elements.

2.2. Data-driven context modeling

In our research, we implemented a data-driven context modeling process, aiming at creating context models to support the elicitation of context-aware functionalities. The process (whose preliminary version was described in [15]) is semi-automated and has three sub-processes: task decomposition, data collection, and data processing.

Task decomposition is a manual sub-process responsible for the identification of which step in the use case that describes the user task will be used to drive the data collection. The *data collection* sub-process is responsible for gathering the contextual data from the context sources and organizing them in a dataset. This sub-process may or may not be automated. Finally, the *data processing* sub-process is responsible for analyzing the dataset and creating the context model based on the results. *Data processing* is an automated sub-process implemented with two software components that cooperate to create context models. At the core, there is a contextual data processor, which takes the dataset (and a metadata file with an additional description of the dataset) as input and produces a list of contexts found to influence the targeted user task. This result is used by the other component, a context model generator, which in turn creates the context model that will be used by requirements engineers.

The context model generator requires a language to represent the model. However, it is important to understand the purpose of the context model we are looking for because context models can support RE activities in different phases.

2.3. Positioning the desired context model

Many context models aim to support the implementation of context-aware systems or context-aware middleware artifacts. As a rule, they are more technical in nature and target software architects and developers who design and implement software components. In contrast, context models[1] intend to help requirements engineers characterize the context on a conceptual level.

[1]Sometimes the terms context model and context meta-model are used interchangeably in the literature to refer to the same level of abstraction. In this paper, we use the term "context model", which is related to level M1 in the meta-model hierarchy defined by the OMG [19].

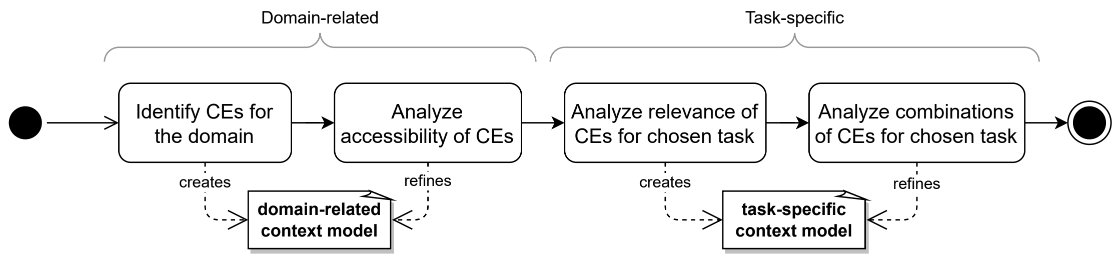

Domain-related
Task-specific
Identify CEs for the domain
Analyze accessibility of CEs
Analyze relevance of CEs for chosen task
Analyze combinations of CEs for chosen task
creates
refines
domain-related context model
creates
refines
task-specific context model

Figure 1: Domain-related and task-specific phases of context modeling in RE.

The activities performed by the involved stakeholders can be split into two phases, as depicted in Figure 1.

The first phase is domain-related. It is more generic and requirements engineers identify the contextual elements that are related to the application domain. As a result, a context model is elaborated. Context models that are elaborated during the domain-related phase are then refined, taking into consideration whether, or to which extent, it is technically feasible to access the identified contextual elements. The second phase is task-specific. It narrows down the focus on the identification of the relevant contextual elements and their combinations for selected user tasks. This is essential for deriving the adaptation logic of the system. In summary, the first phase focuses on answering *what* the context for the application domain is, and the second phase answers *how* the context influences the user tasks. Since automated context modeling is all about analyzing relevance and combinations of contextual elements, it asks for a context model that fits the task-specific phase.

3. Related Work

Some context model representations that relate to the requirements engineering (RE) phase are based on the Tropos goal model [20]. Ali et al. [21] extended it to model and analyze location-based software, but since this model was location-specific and did not relate to context in general, its applicability is limited. Later, the authors introduced the *contextual goal models* (CGM) [22]. CGMs are designed to support RE. They are task-related, expressive in terms of combinations of CEs, domain-independent due to their high level of abstraction, and described using a formal language. However, this model expresses activation contexts and required contexts for tasks and goals, and not which contexts influence tasks. Contexts that influence a user task are not necessarily enablers of the task, as implied by the semantics of CGMs. Later, Ali et al. [23] extended the CGM concept and introduced the notion of *target context*, which relates to goals in the model. It could be adapted to express the idea of intrinsic CEs: From our example in Section 4.1, the goal could be “Prepare a cup of coffee with coffee size = LARGE”, and the context that leads to this goal could combine location and time, as we have in our example. Still, the semantics of the CGM would be broken, as we would need at least one task leading to the goal, which does not exist in the data-driven context modeling process.

Mostéfaoui and Brézillon [24] use contextual graphs, a model that is syntactically similar to TASC4RE, as it is also structured as a directed acyclic graph. This model is, however, semantically different from TASC4RE because it expresses contexts that are necessary to perform actions in a process, whereas our model expresses contexts that influence one specific user task of interest. From an RE point of view, their model is better suited for documenting already known context-aware functionalities; contextual graphs do not help too much in identifying new ones.

Jeong-Dong et al. [25] propose a model based on the 5W1H method, which has been frequently used in requirements engineering of context-aware systems [6]. The method includes answering six questions (What, Why, Who, Where, When, and How) to identify contextual elements. The model applies ontological concepts to support the development of context-aware services. This model is well suited to the domain-related scope of context models (see Section 2.3), but falls short in expressing task-specific contexts.

Bauer and Spiekermann [13] propose a framework for conceptualizing context during the requirements engineering phase of context-aware systems. At the end, a context model in the form of a taxonomy is created. However, their model is domain-related and not task-specific.

Saidani et al. [26] introduced a context modeling approach to support BPM. Their generic meta-model associates business process components (which can include what we consider user tasks) and contextual situations (which matches our definition of context). From a semantic point of view, however, their model does not fit our purposes because the association between business process contexts and contextual situations "expresses the fact that a business process component cannot be executed unless one or more contextual conditions are met".

TASC4RE contrasts existing work by providing a unique set of characteristics. It is task-specific and designed to tell requirement engineers not simply which contexts are necessary to perform user tasks, but how the context influences user tasks. Furthermore, TASC4RE has abstract constructs to express how contexts influence user tasks, and is therefore domain-independent. Its expressiveness is improved by its differentiation between intrinsic and extrinsic properties of contextual entities. Finally, TASC4RE is formally described to support automated context modeling approaches.

4. TASC4RE

We propose a context model representation to support data-driven context modeling, called *Task-specific Context Model for Requirements Elicitation* (TASC4RE). It represents *task-specific contexts* in a directed acyclic graph that resembles natural language sentences such as "When... $context_1$ then... *task* with... $context_2$". The idea is to express how the context typically looks like, according to the data analysis, when the user performs a task of interest. By expressing which contexts influence which tasks, TASC4RE can provide better support for the elicitation of concrete functionalities in comparison to domain-related models.

To get the greatest benefit from an automated approach, minimal effort should be shifted to the preparation steps, which would include the tailoring of a context meta-model for a specific application domain. For this reason, TASC4RE has a formal description and generic construct elements. Here we introduce its graphical domain-specific language (DSL), the notion of intrinsic contextual elements used by it, and formalize its language.

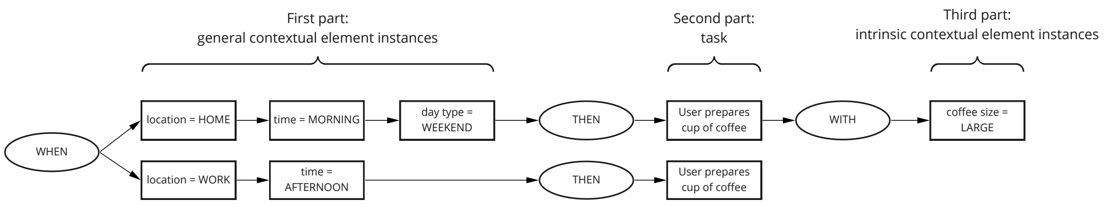


Figure 2: Example of a task-specific context model (adapted from [17]).

Table 1
Elements of the task-specific context model's graphical DSL.

Element	Type	Usage
WHEN	Structural	Starting point, preceding contextual element instances
THEN	Structural	Predecessor of task declaration
WITH	Structural	Predecessor of contextual element instances that are intrinsic to the object entity
→	Structural	Connector
▭	Content	Container for dynamic content. Can be filled with the task description or contextual element instances

4.1. Graphical DSL

The task-specific context model is expressed through a graphical DSL consisting of five graphical elements; when connected, they express task-specific contexts. Table 1 lists the elements of the language. The ellipses (WHEN, THEN, WITH) are structural elements that bring the expression of relevant contexts closer to natural language, so practitioners can read the model more easily. The rectangle (▭) is the content element, which is used as a container for the description of contextual element instances and user tasks. The arrow (→) is a structural element that acts as a connector between the other elements.

Fig. 2 shows an example. The diagram is a directed acyclic graph with one root node (WHEN). Each path from the root node towards a leaf is a *task-specific context* (*TSC*) and describes how a context influences a user task of interest. The task-specific context has three parts: the first with general contextual element instances; the second with the task description; and the third with contextual element instances that are intrinsic to the *object entity* (see Section 4.2).

Consider the user task "Prepare a cup of coffee" in Fig.2. Each of the two paths contains a set of contextual element instances that, together, were found to influence the task. For example, in "When location = WORK and time = AFTERNOON then user prepares coffee", the context "location = WORK and time = AFTERNOON" (combination of instances of two contextual elements: location and time) influences the user task "Prepare a cup of coffee", according to the model.

We formalized the language to describe the task-specific contexts using the Extended Backus-Naur Form (EBNF) [27] (see Listing 1).

Listing 1: Formal syntax for describing a task-specific context.

TSC = WHEN, $\left\{ \rightarrow, \boxed{\textit{General CE Instance}} \right\}$-, →, THEN, →, $\boxed{\textit{Task Description}}$, →, $\left\{ \text{WITH}, \left\{ \rightarrow, \boxed{\textit{Intrinsic CEInstance}} \right\}\text{-} \right\}$;

General CE Instance = ? Textual description of contextual element instance that is not intrinsic to the object entity ?;

Task Description = ? Textual description of the task (preferably in active voice) ?;

Intrinsic CE Instance = ? Textual description of contextual element instance that is intrinsic to the object entity ?;

4.2. The object entity and its intrinsic properties

The name of the user task usually names the corresponding use case, and use case names, as described in [28], "are short active verb phrases naming some behavior found in the vocabulary of the system you are modeling". The verb may have an object. For example, in the user task "Prepare a cup of coffee", the object is "a cup of coffee"; in the user task "Watch a movie", the object is "a movie". During the identification of the contextual elements that should be taken into consideration in a certain application, each identified contextual element characterizes a certain contextual entity. When the user task object is a contextual entity, we say that it is an *object entity*.

Some contextual elements characterize the object entity *intrinsically* and others *extrinsically*. An intrinsic property of a thing is entirely about the thing, whereas an extrinsic property is not [29][2]. The concept of "intrinsic/extrinsic context" was previously introduced by Costa et al. in [30]; here we refer to "intrinsic/extrinsic contextual elements". In the task-specific context model, it is important to identify the contextual elements that characterize the object entity intrinsically. This information is used by the automated modeling process to position the contextual element instance either in the first part of the task-specific context or in the third part (see Fig. 2).

For example, consider the user task "Prepare a cup of coffee". The object entity is "cup of coffee", and the list of contextual elements that characterizes this entity could include:

1. coffee size
2. coffee type
3. time of the day when the cup of coffee is prepared
4. location where the coffee is drunk
5. person who prepares the cup of coffee

Only the first two contextual elements (size and type) are intrinsic, as they relate only to the entity *cup of coffee*. The others relate at least in part to other entities – *environment* (elements 3 and 4) and *user* (element 5).

[2]There is a long discussion in philosophy about the definition of intrinsic and extrinsic properties. The definition presented by [29] provides a basis for further consideration. For our purposes, though, if suffices, for it provides an intuitive definition that can be used to help identify intrinsic contextual elements.

A task-specific context could be “WHEN *user location = home* THEN *user prepares coffee* WITH *coffee size = LARGE*”. Without the differentiation between intrinsic and extrinsic contextual elements, the task-specific context would be “WHEN *user location = home* and *coffee size = LARGE* THEN *user prepares coffee*”, which would sound strange because the size of the coffee does not properly characterize the external circumstances (“When...”) that drove the execution of the task at first place. The presence of the element WITH perfectly bridges the introduction of contextual element instances that are intrinsic to the object entity.

5. Evaluation

We evaluated TASC4RE in the context of a controlled experiment where we implemented a data-driven context modeling process and applied it to a live software project. As a result, a data-driven context model was created, which used TASC4RE as the language to represent it. The goals and content of the controlled experiment go beyond the scope of this paper; its complete description can be found in our previous publication [17]. All materials and raw results are available. In this section, we highlight the aspects of the controlled experiment that concern the usage of TASC4RE, and add qualitative results, which were not present in the previous paper.

5.1. The project setting

In the project Digital Villages [31], the needs of people living in villages were elicited and various software-based solutions were created to address problems regarding communications, public administration, local supply, and logistics. One of these solutions is DorfFunk[3], a communication app with the characteristics of a social network, which has approximately 25,000 active users. In DorfFunk, village residents can interact and get informed about their local community by checking the newsfeed, creating posts, making comments, giving likes, and using other functionalities. DorfFunk was developed and has been maintained by Fraunhofer IESE for more than five years.

5.2. The tool support

The tools xPACE and TASC Modeler realized the contextual data processor and the context model generator, respectively, which are the automated part of the data-driven context modeling process (see Section 2.2). The former applies statistical methods to analyze the contextual dataset and produces a machine-readable list of relevant contexts that were found to influence the user task of interest; the latter takes the relevant contexts identified by xPACE as input to create a human-friendly representation of a context model to support requirements engineers in the elicitation of context-aware functionalities. TASC Modeler uses TASC4RE as the language to express the context models. Further information about these tools, including a description of the software architecture, details about the data-processing algorithm, and screenshots of the TASC Modeler can be found in [32].

[3]https://www.digitale-doerfer.de/unsere-loesungen/dorffunk/

5.3. The experiment

We performed a controlled experiment using a one-factor two-treatments randomized design [33] with thirteen experienced professional software engineers, where we asked them to elicit context-aware functionalities to improve the system-supported user task "Create a comment" of the DorfFunk app. In this task, the user writes a comment to an existing post that they saw in their feed (similar to what a Facebook[4] user would do when they comment on a post in the social network). The control group received a list of contextual elements available to support them in elaborating context-aware functionalities, whereas the treatment group (five participants) received the data-driven context model generated with our tools and therefore expressed using TASC4RE. The number of participants in the groups were different due to a deviation that was observed during the execution of the experiment.

Our hypothesis was that *the data-driven context model is perceived by individuals as a useful instrument to support the elicitation of context-aware functionalities.* To test this hypothesis, we adapted the UTAUT (Unified Theory of Acceptance and Use of Technology [34]), tailoring the items of five aspects – Performance expectancy (PE), Effort expectancy (EE), Attitude toward using technology (AT), Self-efficacy (SE), and Anxiety (AX) – to our needs. For each aspect, the participants of the treatment group provided their level of agreement to a set of items using a 5-point Likert scale. We coded the answer options numerically (1-Strongly disagree, 2-Disagree, 3-Neutral, 4-Agree, 5-Strongly agree). We reversed the code in item AT.1 of aspect AT, and in all items of aspect AX, for they were stated in a negative way.

Furthermore, we asked participants to provide (optionally) further comments about the experiment through an open-end question, from where we could get additional insights that were not captured in the hypothesis testing.

5.3.1. Preparation

Before using the tools, we performed post-hoc data collection to create a contextual dataset containing information about fourteen contextual elements that were identified for the application and which where accessible in our project setting. The list of contextual elements was: user identity, user account type, and user role, concerning the contextual entity "User"; event start date/time, post author, post creation time, post day type, post date/time of last comment, part of the day of the post, and post type, concerning the contextual entity "Post"; current date/time, current day type, and current part of the day, concerning the contextual entity "Environment"; and content type, concerning the contextual entity "Comment". Once the contextual dataset had been created, we used the tools to create the data-driven context model.

5.3.2. Results

Hypothesis testing Table 2 shows the description of the items, the scores of each aspect, and their reliability. The item scores were calculated based on the mean value of the participants' ratings; the aspect score was calculated as the mean of the item scores composing each aspect. To verify the reliability of the scores, we calculated their Cronbach's alpha, which was higher

[4]https://facebook.com

Table 2
The five investigated aspects of usefulness of the data-driven context model (expressed using TASC4RE).

Aspect	Item	Description	Item score	Aspect score	Cronbach's α
PE - Performance Expectancy	PE.1	I would find the data-driven context model useful in the elicitation of context-aware functionalities.	4.8	4.6	0.778
	PE.2	Using the data-driven context model enables me to accomplish the elicitation of context-aware functionalities more quickly.	4.4		
	PE.3	Using the data-driven context model increases my productivity.	4.6		
EE - Effort Expectancy	EE.1	My usage of the data-driven context model would be clear and understandable.	3.8	4.05	0.468
	EE.2	It would be easy for me to become skillful at using the data-driven context model.	4.2		
	EE.3	I would find the data-driven context model easy to use.	3.8		
	EE.4	Learning to interpret the data-driven context model is easy for me.	4.4		
AT - Attitute towards using Technology	AT.1	Using the data-driven context model is a bad idea.	5	4.34	0.804
	AT.2	The data-driven context model makes elicitation of context-aware features more interesting.	4.2		
	AT.3	Working with the data-driven context model is fun.	3.8		
	AT.4	I like working with the data-driven context model.	4.4		
SE - Self-Efficacy	SE.1	I could elicit context-aware functionalities using the data-driven context model if there was no one around to tell me what to do as I go.	3.2	3.75	0.752
	SE.2	I could elicit context-aware functionalities using the data-driven context model if I could call someone for help if I got stuck.	4.4		
	SE.3	I could elicit context-aware functionalities using the data-driven context model if I had a lot of time to complete the job for which the data-driven context model was provided.	4.4		
	SE.4	I could elicit context-aware functionalities using the data-driven context model if I had just the data-driven context model itself for assistance.	3		
AX - Anxiety	AX.1	I feel apprehensive about using the data-driven context model.	3.8	4.2	0.903
	AX.2	I hesitate to use the data-driven context model for fear of making mistakes.	4.6		
	AX.3	The data-driven context model is somehow intimidating to me.	4.2		

than 0.7 for PE, AT, SE, and AX, hence acceptable [35], whereas for EE it was not acceptable. Apart from the aspect Self-efficacy (SE), all aspects had high scores (> 0.4); therefore, we can conclude that the participants evaluated the model positively.

Content analysis of open-end question Nine participants – five from the treatment group – answered to the open-end question and we organized their comments into five clusters:

Difficult eliciting context-sensitive functionalities: One participant of the treatment group mentioned that "[w]hile in theory the task [elicitation of context-aware functionalities] was clear, it was not easy in practice, but I really like the idea of this model as it provides many useful information that one would typically not really think of and which can tremendously improve the requirements and hence the solutions". Four participants of the control group (i.e., participants that did not have the data-driven context model at hand) commented on the difficult of eliciting context-aware functionalities (e.g., "It is really hard to come up with valuable use cases", "I found it difficult to create the requirements", "I struggled a bit with coming up with new requirements at a approx. 15 minutes", "I would suggest to make the task more concrete").

Usefulness of the model: Two participants from the treatment group explicitly praised the support provided by the data-driven context model to help them eliciting context-aware functionalities. One of them mentioned that the insights provided by the model were more evident in contexts that included the intrinsic part (which was not always the case). According to the participant, "[t]hrough the complementary part of the model (with...), it was easier to derive the goal of the user".

Usage of the model: Two participants from the treatment group mentioned that they felt pressured by time during the experiment. One of them wrote: "when provided with the model I was overwhelmed (...) it required cognitive efforts to interpret the data"; the other commented that in real life they "would be able to analyze the context model better and use it to ask meaning questions to [their] stakeholders".

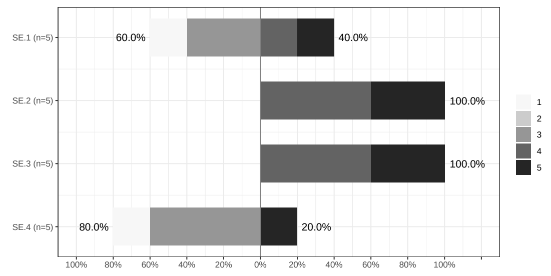
Post processing the model: Two participants from the treatment group referred to the existence of noise in the data-driven context model and suggested a human-based post-processing step to validate it, "to remove things that do not make sense at all", as pointed by one of them.

Experimental procedure: Two participants from the treatment group complimented the experimental procedure (e.g. "Very well explained", "I felt prepared and knew what to do", "good explanation").

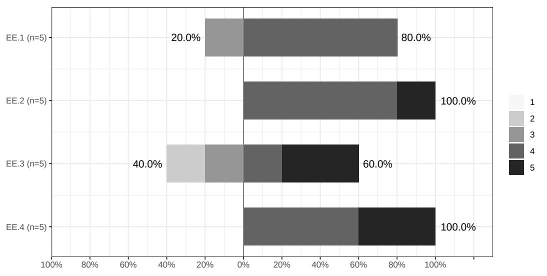
6. Discussion

The focus of this work is not on data-driven context modeling processes, but rather on context models that can be used to support such automated approaches. TASC4RE serves as a formal representation for task-specific context models designed to support data-driven context modeling approaches. It also expresses task-specific contexts in a human-friendly way through its graphical DSL, which structures the contexts influencing the user task of interest as in natural-language phrases. An important contribution of TASC4RE is the incorporation of the notion of extrinsic and intrinsic contextual elements into the context model representation to avoid compromising its expressiveness.

The intended simplicity of the TASC4RE notation aims at making it easy to understand, while giving it the flexibility to be used generally. This claim regarding its generalizability must be investigated, though. The question is whether all possible contexts that may influence user tasks of interest can be expressed in terms of "when *context* then *task* [with *intrinsic context*]". In our specific application (see Section 5), TASC4RE was successfully used to express complex contexts found to influence the user task with algorithms that sometimes combined five contextual



(a) Self-efficacy.



(b) Effort expectancy.

Figure 3: Part of participants' assessment with respect to the usefulness of the model.

elements (out of fourteen) in the same context.

The results of the experiment showed a positive trend towards acceptance of TASC4RE. All aspects but Self-efficacy (see Figure 3a) showed high scores. This is partially confirmed by the feedback received from some participants through the open-end question, and led us to reflect on how we could improve the DSL, or the overall user experience with the model, to make its usage even clearer. Concerning the reliability of the results, the scores of all aspects but Effort expectancy (see Figure 3b) were considered reliable. We found the cause in item EE.1, which referred to how clear and understandable the usage of the data-driven context model would be. If EE.1 was removed, the aspect score would be 4.13 and the reliability, acceptable (0.733).

It is worth noting that, in the experiment, participants of the treatment group worked individually, though. In practice, we envision the usage of the data-driven context model as part of a group activity to elicit context-aware functionalities. We expect that the synergy among participants will speed up the ability of getting more insights from the model and improving the chances of correctly discarding occasional noise in the model. Furthermore, the participants evaluated TASC4RE in the context of a concrete task-specific model that was generated using contextual data of a certain user task of DorfFunk, meaning that the positive evaluation of the model is bound not only to TASC4RE itself, but also to the content of the concrete model and to the data-driven process (in particular the algorithm) that analyzed the contextual data – i.e., confounding constructs are one of the threats to the validity of the results.

7. Conclusion

Context models have provided very limited support when it comes to requirements elicitation of context-aware functionalities. Context modeling for requirements elicitation is perceived as a very complex activity by practitioners, and has largely been overlooked. As data-driven context modeling seems to be promising to overcome the complexity of context modeling for requirements elicitation, suitable context models are required by such approaches.

In this paper, we presented TASC4RE, a context model designed to support data-driven context modeling approaches in representing how contexts influence user tasks of interest. It is a formally specified context model that combines the intuitiveness of a directed graph representation and the expressiveness of natural language to help requirements engineers come up with novel, unexpected context-aware functionalities based on data insights. The usage of TASC4RE was positively evaluated in a controlled experiment with professionals in the context of a software development project.

As future work, we plan to use TASC4RE in more projects to validate its expressiveness in multiple domains. It would also be beneficial to conduct a comparative study to get more empirical evidence about the expressiveness of TASC4RE in contrast to other RE-focused context models that aim at supporting the elicitation of context-aware functionalities.

Acknowledgments

This work was partially supported by CNPq, Brazil ; and by the German Federal Ministry of Education and Research (BMBF), in the project DynaSoS (grant no. 01|S21104).

References

[1] B. Schilit, N. Adams, R. Want, Context-aware computing applications, in: 1994 first workshop on mobile computing systems and applications, IEEE, 1994, pp. 85–90.

[2] C. Bettini, O. Brdiczka, K. Henricksen, J. Indulska, D. Nicklas, A. Ranganathan, D. Riboni, A survey of context modelling and reasoning techniques, Pervasive and Mobile Computing 6 (2010) 161–180.

[3] M. Baldauf, S. Dustdar, F. Rosenberg, A survey on context-aware systems, International Journal of Ad Hoc and Ubiquitous Computing 2 (2007) 263–277.

[4] Z. Aarab, R. Saidi, M. D. Rahmani, Context modeling and metamodeling: A state of the art, in: Proceedings of the Mediterranean Conference on Information & Communication Technologies 2015, Springer, 2016, pp. 287–295.

[5] P. Pradeep, S. Krishnamoorthy, The MOM of context-aware systems: A survey, Computer Communications (2019).

[6] V. Vieira, CEManTIKA: A Domain-Independent Sistema for Designing Context-Sensitive Systems, Ph.D. thesis, Centro de Informática–UFPE, Brasil, 2008.

[7] C. Bauer, S. Spiekermann, Conceptualizing context for pervasive advertising, in: Pervasive Advertising, Springer, 2011, pp. 159–183.

[8] C. Bauer, A. Novotny, A consolidated view of context for intelligent systems, Journal of Ambient Intelligence and Smart Environments 9 (2017) 377–393.

[9] C. Bauer, A. K. Dey, Considering context in the design of intelligent systems: Current practices and suggestions for improvement, Journal of Systems and Software 112 (2016) 26–47.

[10] K. Henricksen, A framework for context-aware pervasive computing applications, Ph.D. thesis, The University of Queensland, 2003.

[11] D. Hong, D. Chiu, V. Shen, Requirements elicitation for the design of context-aware applications in a ubiquitous environment, volume 113, 2005, pp. 590–596. doi:10.1145/1089551.1089658.

[12] P. Brézillon, Context modeling: Task model and practice model, in: International and Interdisciplinary Conference on Modeling and Using Context, Springer, 2007, pp. 122–135.

[13] C. Bauer, A framework for conceptualizing context for intelligent systems (CCFIS), Journal of Ambient Intelligence and Smart Environments 6 (2014) 403–417.

[14] R. Falcão, K. Villela, V. Vieira, M. Trapp, I. L. de Faria, The practical role of context modeling in the elicitation of context-aware functionalities: a survey, in: 2021 IEEE 29th International Requirements Engineering Conference (RE), IEEE, 2021, pp. 35–45.

[15] R. Falcão, Improving the elicitation of delightful context-aware features: A data-based approach, in: 2017 IEEE 25th International Requirements Engineering Conference (RE), 2017, pp. 562–567. doi:10.1109/RE.2017.42.

[16] K. Villela, A. Heß, M. Koch, R. Falcão, E. C. Groen, J. Dörr, C. N. Valero, A. Ebert, Towards ubiquitous RE: A perspective on requirements engineering in the era of digital transformation, in: 2018 IEEE 26th International Requirements Engineering Conference (RE), IEEE, 2018, pp. 205–216.

[17] R. Falcão, M. Trapp, V. Vieira, A. Vianna Dias da Silva, Using a data-driven context model to support the elicitation of context-aware functionalities–a controlled experiment, in: International Conference on Product-Focused Software Process Improvement, Springer, 2021, pp. 119–135.

[18] A. K. Dey, Understanding and using context, Personal and ubiquitous computing 5 (2001) 4–7.

[19] Object Modeling Group, Meta-Modeling and the OMG Meta Object Facility (MOF), https://bit.ly/3PAWc5K, 2017. Accessed: 2021-11-09.

[20] P. Bresciani, A. Perini, P. Giorgini, F. Giunchiglia, J. Mylopoulos, Tropos: An agent-oriented software development methodology, Autonomous Agents and Multi-Agent Systems 8 (2004) 203–236.

[21] R. Ali, F. Dalpiaz, P. Giorgini, Location-based software modeling and analysis: Tropos-based approach, in: International Conference on Conceptual Modeling, Springer, 2008, pp. 169–182.

[22] R. Ali, F. Dalpiaz, P. Giorgini, A goal-based framework for contextual requirements modeling and analysis, Requirements Engineering 15 (2010) 439–458.

[23] R. Ali, F. Dalpiaz, P. Giorgini, V. E. S. Souza, Requirements evolution: from assumptions to reality, in: Enterprise, Business-Process and Information Systems Modeling, Springer, 2011, pp. 372–382.

[24] G. K. Mostéfaoui, P. Brézillon, Modeling context-based security policies with contextual graphs, in: IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second, IEEE, 2004, pp. 28–32.

[25] K. Jeong-Dong, J. Son, D.-K. Baik, CA5W1H Onto: Ontological Context-Aware Model Based on 5W1H, International Journal of Distributed Sensor Networks (2012).

[26] O. Saidani, C. Rolland, S. Nurcan, Towards a generic context model for bpm, in: 2015 48th Hawaii International Conference on System Sciences, IEEE, 2015, pp. 4120–4129.

[27] ISO/IEC 14977:1996, ISO/IEC 14977 - Information Technology - Syntactic metalanguage - Extended BNF, Standard, ISO, 1996.

[28] J. Rumbaugh, G. Booch, I. Jacobson, The unified modeling language user guide, Addison-wesley, 1998.

[29] D. Lewis, Extrinsic properties, Philosophical Studies 44 (1983) 197–200.

[30] P. D. Costa, G. Guizzardi, J. P. A. Almeida, L. F. Pires, M. Van Sinderen, Situations in

conceptual modeling of context, in: 2006 10th IEEE International Enterprise Distributed Object Computing Conference Workshops (EDOCW'06), IEEE, 2006, pp. 6–6.
[31] M. Trapp, S. Heß, Digital villages, in: Biological Transformation, Springer, 2019.
[32] R. Falcão, R. King, A. L. Carvalho, xPACE and TASC Modeler: Tool support for data-driven context modeling, in: REFSQ'22, CEUR WS, 2022. URL: http://ceur-ws.org/Vol-3122/.
[33] C. Wohlin, P. Runeson, M. Höst, M. C. Ohlsson, B. Regnell, A. Wesslén, Experimentation in software engineering, Springer Science & Business Media, 2012.
[34] V. Venkatesh, M. G. Morris, G. B. Davis, F. D. Davis, User acceptance of information technology: Toward a unified view, MIS quarterly (2003) 425–478.
[35] J. F. Hair, W. Black, B. Babin, R. Anderson, Multivariate data analysis, Pearson, 2009.