How to identify the relevant elements of “context” in Context-aware Information Systems?

Kurt Sandkuhl, Ulrike Borchardt
Institute of Computer Science, University of Rostock, 18051 Rostock, Germany
[kurt.sandkuhl, ulrike.borchardt]@uni-rostock.de

Abstract. Context-awareness is a feature of more and more applications, which adds further requirements to be taken into account in the implementation process. Though accepted approaches for software development exist, no accepted way for the inclusion of context has been established yet. An essential part of developing context based systems is to analyze and conceptualize the elements of the specific context required for the application under development, including their dependencies and mechanism of use. This activity of context modeling forms an important part of the system’s specification, since it identifies relevant aspects of the application environment in a representation adequate for the modeling purpose. Within this paper we aim at closing this gap by introducing an approach for context modeling for the utilization in context-aware applications, providing a structure guiding through the process and illustrating it by examples as a reference for further projects.

1 Introduction

„Context-awareness“ has emerged from a special and innovative feature of niche applications to a characteristic of many IT Systems in modern enterprises. At the beginning of the century, Dey’s seminal work about context as information characterizing the situation of an entity [1] paved the way for context-aware ubiquitous computing and assistive systems. Nowadays, enterprise portals, groupware systems, assistive systems or control systems are including mechanisms to adapt to the users’ situation on demand – to just name a few examples.

However, design and development of context-awareness in information systems still require substantial engineering work, i.e. there is no general development methodology for context-based systems. One reason for this probably is the variety of interpretations of the term context in the area of computer science (see section 2). An essential part of developing context based systems is to analyze and conceptualize the elements of the specific context required for the application under development, including their dependencies and mechanism of use. This activity of context modeling forms an important part of the system’s specification, since it identifies relevant aspects of the application environment in a representation adequate for the modeling purpose. Furthermore, the context as such is also required during runtime of a context-aware information system, i.e. the context model is not only a conceptualization but has to be reflected in appropriate information structures and instantiated in the actual system.

In this paper we focus on context modeling for IT-application cases in enterprises aiming at the support of human actors, i.e. we do not address context modeling for purely technical systems or cyber-physical systems. Based on experiences from several cases (see section 3) we propose to use techniques from enterprise modeling for context modeling and derive recommendations for context modeling activities. The main contributions of the paper are (1) an analysis of past context modeling cases with respect to commonalities in development processes, (2) an approach defining development steps of a context model, and (3) first experiences with the new context modeling approach.

As already indicated above, the question guiding our research is “What steps has a ‘good practice’ procedure for identifying and modeling the elements of context to include?” Good practice in this context has the meaning of a proven procedure for reliably completing a defined task, which originates from knowledge management [18]. The research process used in our work includes a deductive and an inductive phase. In the deductive phase, we analyze previous context modelling cases with respect to the sub-question “what commonalities and differences do context modeling procedures show?” The “data” available from these previous cases consist of project reports, notes of the developers and our own (undocumented) experiences. Based on these case data, an initial approach for a context modeling procedure was developed. The inductive phase includes usage and improvement of the initial approach in (new) application cases, which also serves validation purposes in order to reach the envisioned “good practice”.

The remaining part of the paper is structured as follows: Section 2 provides a brief overview to context modeling in computer science and defines the term context. Section 3 presents context modeling projects from the past. These projects serve as experience basis for deriving our good practice approach for context modeling introduced in section 4. Section 4 describes the approach and relates it to the cases. Section 5 applies the new approach in an e-learning project. Section 6 summarizes the achievements and gives an outlook to future work.

2 Context Modeling in Computer Science

The term “context” has been used and still is subject of research in various application areas and sectors of computer science, which will be briefly summarized in this section. In the most general meaning, context describes what relates the entity under consideration to the environment surrounding this entity. What an “entity” is depends on the actual interpretation of context. Hoffmann [2] provided a way to classify these interpretations as follows:

- Linguistic context is used for disambiguating the meaning of words in texts and denotes the words surrounding the word of interest. An example is the approach presented in [3] to disambiguate keyword-based search using the paragraph surrounding a keyword of interest.
- Situational context includes any information characterizing the state or situation of a person, object or location. This information has to support the purpose of understanding or being relevant for the interaction between user and application. Situational context models are often used in ubiquitous computing [1].
- Relational context includes any information pertinent to characterizing the relation of an entity to other entities, where this information is judged according

to a given purpose. An example from problem solving is given in [4] using contextual graphs for this purpose.
- Formal representations of a perception or part of reality are like a model of an individual's viewpoint, which expresses a local view of the reality.

In this paper, we use the term context according to Dey, who defines context as "*any information that can be used to characterize the situation of an entity, where 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 the application themselves.*" [1]

Although the term context is widely used in computer science, there is no general procedure how to develop context models. Many authors of context-based systems describe the way of developing the context model for their specific application, but do not provide a general view. [5] and [6] show examples for UML-context development in pervasive computing and OWL-based context for reasoning applications. Mena and colleagues [8] sketch a development process for context –aware systems and identify invariant characteristics of context as part of their work. These characteristics are (a) context relates always to some entity, (b) is used to solve a problem (c) depends on the domain and (d) is a dynamic process.

The development of a procedure for context modelling should be anchored in experiences from past projects in order to exploit the lessons learned. In addition to our own experiences presented in section 3, surveys analysing previous context work form an important contribution. Bazire and Brézillon [7] analyzed the state of research in the area including more than 150 definitions. One of the conclusions from this work study is that "*the context acts like a set of constraints that influence the behaviour of a system (a user or a computer) embedded in a given task.*" Furthermore, in [7] it is concluded that context definitions should be analysed using six parameters: constraint, influence, behaviour, nature, structure and system.

3 Context Modeling Cases

The context modeling approach presented in section 4 is based on experiences from a number of projects where adaptivity to situations was a decisive feature. Two of these context modeling application cases are presented in this section with their background or application scenario, the development process for the context model, and the context model as such. The selected context modeling cases are from information demand context and context-based ontology matching. The other cases analyzed but not included for brevity reasons are context-modeling for capability-as-a-service [14], situation-based messaging [15] and decision support [16].

3.1 Information Demand Context

Background. The first context modelling case originates from the field of information logistics which focuses on improving information flow in enterprises and on demand-oriented information supply. A core subject of demand-oriented information supply is to capture the needs and preferences of a user in order to get a fairly complete picture of the demand in question. This requires an understanding of what information demand is and a method for capturing and analysing information demand.

Information demand has a strong relation to the context in which such a demand exists [8]. The organisational role having the demand, the task the information is demanded for as well as the setting in which such tasks are performed are important aspects for understanding information demand. Thus, the concept of information demand context has been defined as follows: "*An Information Demand Context is the formalised representation of information about the setting in which information demands exist and comprises the organisational role of the party having the demand, work tasks related, and any resources and informal information exchange channels available, to that role.*"

Development process of context model. The approach used for developing an information demand context model is information demand analysis; the main characteristics are published in [4] and summarized in the following.

Scoping is the process of defining the area of analysis and is done with the purpose of selecting parts of an organisation to be subjected to analysis. This phase also includes the identification of the roles (individuals) relevant for the continued information demand analysis. *Information Demand Context Modelling* is mainly performed through participative activities such as joint modelling seminars where the participants themselves are involved in the actual manufacturing of different models. This process is usually supported and facilitated by a method expert who could be an internal or external person. The key to context modelling is to identify the interrelationship between roles, tasks, resources and information. No regard is given to the sequence of activities, resource availability, etc. *Information Demand Context Analysis and Evaluation:* Once the necessary knowledge about the information demand contexts is obtained, it can be used for a number of different purposes. One purpose is evaluation where different aspects of information demand can be evaluated in relation to roles, tasks, resources and information. It is also suitable to address the results from the modelling session with respect to motivation and purposes expressed during scoping activities. Focusing on information demand contexts provides only an initial view of information demand without any consideration given to such aspects as individual competence, organisational expectations and requirements in terms of goals, processes etc. Depending on the intentions behind the analysis further activities might be required. The method provides a number of method components supporting such activities. If the method user wishes to investigate such additional aspects of information demand, he or she can do this by using subsets of the other methods, notations and languages.

Context model. An information demand context model basically is an excerpt from an enterprise model for a specific role showing the processes the role is involved in, the co-workers in the organization structure, the resources used and the IT-systems applied.

Figure 1 illustrates the relation between enterprise model and information demand context. For the considered role, the information demand context is derived from an existing enterprise model by extracting (a) all tasks the role is supposed to perform or is responsible for, (b) the co-workers, superiors and sub-ordinate positions, (c) the resources required for the tasks gained in step a. The information demand context at design time is instantiated with the active tasks, persons assigned to the positions and resources in use and forms the actual information demand at the point of runtime.

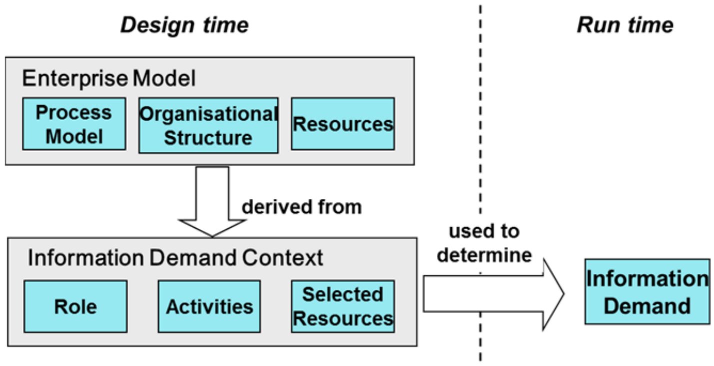

Figure 1: Information Demand Context

3.2 Context-based Ontology Matching

Background. Although ontologies are developed for various purposes and domains, they often contain overlapping information. Ontology matching aims at finding similar entities or translation rules between two ontologies. Ontology matching is an important technique for creating a collaborative semantic web. However, currently existing approaches for automatic ontology matching do not sufficiently take into account context dependencies in the process of matching. This leads to situations where the results of automatic matching are of limited or no use for the task or application at hand. An increased user involvement can be a way to improve the quality of matching results [9]. The second context modeling case aims at facilitating a new way for user involvement by using a context ontology capturing both, tasks of the user and user preferences.

Development process of context model. The context aims at reflecting the information demand of a role in an enterprise. Role here means a part of a larger organizational structure clearly defined by the responsibility it has within that structure [10]. The context is modeled in two levels: abstract context and operational context. Abstract context is an ontology-based model integrating information about the role. Operational context is the instance of the abstract context for a specific role. Normally the context consists of three parts:

- The information about the tasks of a role included in the enterprise ontology.
- Information about tasks of the role that is related to the enterprise but not inside the enterprise ontology. This is additional information provided by the role based on his/her knowledge.
- Additional information about the role, every individual having the role, for example, the competence of the individual having the role.

Context model. An example for a context model for context-based ontology matching is shown in Figure 2. The example shows the context model for a person (concept "expert=person(CODISPLAY)") with the projects and training courses this person was involved in. This context model is supposed to represent the competence of the person which should be taken into account when searching for experts and competences complementary to the person's competence. The context model is used in ontology matching during relevance calculation of matching concepts in two ontologies identified for expert finding and competence supply purpose.

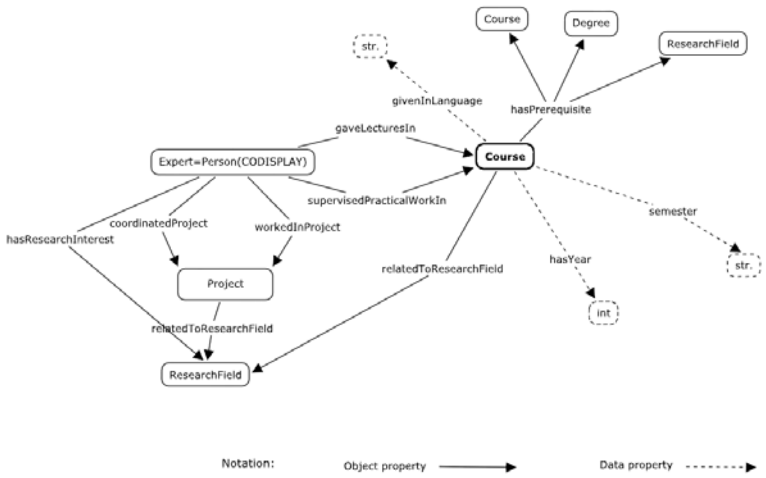

Figure 2: Ontology Matching Context

4 Context Modeling Approach

Based on the cases presented in section 3, an approach for context modeling has been devised. The cases were analyzed in order to find commonalities and differences in the development process and in structure and content of the context models. The most significant commonalities were (a) all cases required to understand what user activities were supposed to be supported using context, (b) the context models were reflecting the potential variations to be supported and (c) the common phases of engineering projects (e.g. requirements elicitation, design, implementation, test) were represented in all cases. The most significant differences were the context models, i.e. their structure and content showed hardly any commonalities. Our conclusion is that we have to support both, developing context models for applications (i.e. context model types) and context models for specific usage cases of these applications (i.e. actual context models). We propose a context modeling approach consisting of six steps, which are discussed in the following:

1. Scenario modeling for the future context-based application
2. Variability elicitation
3. Initial context model development
4. Implementation of context-based application
5. Alternating model-instance improvement
6. Theory and practice validation

These six steps form our hypothesis for a context modeling approach and require evaluation regarding completeness, practicability and refinement needs. They reflect the typical phases of engineering projects found in all analyzed cases with a specialization for accommodating variability aspects and different context types.

4.1 Scenario modelling

The purpose of the first step is to identify user groups and intended scenarios of use for the future context based system. This step is similar to the first phase of information system development or software projects. In order to understand which user groups exist and how their ways of using the future system differ from each other, the process supported by the system, the information input and output, possible connections to other systems and processes, or the integration of resources have to be analyzed and described. This may be done using conventional use case modeling (e.g. from RUP), business modeling in UML 2.0, goal-process-actor modeling in 4EM or other techniques. The result of this step are scenario representations, e.g. as diagrams or visual models. In order to be suitable for context modeling, the scenario descriptions have to include and identify:

- The different user groups of the future context-based IT system
- The tasks the users are supposed to perform with the future system. This should at least include the primary scenario (often referred to as “success scenario” or “happy flow”) with steps to perform
- Information input or conditions which cause branching in the flow of actions during the tasks

4.2 Variability elicitation

The second step is probably the most important one. A context model has to include in what situations and on what inputs or events what kind of adaptations in the context-based system should be made. Adaptations can concern functionality, behavior, output or appearance of the system. Since the results of these adaptations of the system can be considered as variations of the use of the system, the system’s behavior or even the system’s configuration, it is decisive to understand the cause and kind of the variation. In order to determine cause and kind of variation, two aspects have to be investigated: the variation aspects and the variation points.

Variation Aspects. Variations in behavior, functionality or content of context-aware systems can be caused by different aspects, like the user groups, the task performed, the information input, etc. In order to identify the relevant aspects, the scenario models developed in step 1 have to be analyzed. In principle, different strategies of doing so are possible, like investigating all tasks in the scenario models and their variations or focusing on causes for different branches in the scenario. Since deciding on the best strategy would require more cases and data collected from analyzing them, we propose a “brute force” strategy based on the scenario models:

The modeling languages mentioned in step 1 “Scenario modeling” include different model component types, like the “process” or “actor” types in 4EM [11]. For each of these model component types, it has to be examined whether different instances of this component type would require an adaptation in the context-aware system. For those component types causing an adaptation it has to be investigated what characteristic of the component type actually is decisive for the adaptation. If, for example, “process” component types would cause adaptation, it has to be investigated whether this is due to process input, process output, process duration or other characteristics. The identified component types and their decisive characteristics are called variation aspects.

Variation Points. Within each variation aspects, the variation points define under which conditions or for which events an adaptation in the context-aware system has to happen. Often even the kind of adaptation can be identified together with the variation points. In order to identify the variation point, all variation aspects identified in the above procedure have to be examined. It is recommended that this is done based on the scenario models by assuming alternatives in the scenarios regarding the validation aspects under consideration.

4.3 Develop initial context model

According to the definition of context, the context contains all information characterizing the situation of an entity. We assume that this information consists of different elements and that each element has different attributes. An example would be a context element "user group" with the attribute of "list of user groups to be distinguished" and "individuals assigned to the user groups". For developing the initial context model, the first task would be to define a context element for each of the identified variation aspects and to decide on the attributes for the context model.

The second task aims at investigating what type of adaptation of the context-based application is related to each context element. For this purpose, we assume that a context-aware application not only has to adapt its own behavior with respect to functionality or what information is provided (active role) but also needs to provide information to other "context-aware" components outside the context-based application to be developed (passive role). An example would be a context element "current user location" which can be used to adapt the context-based application under development, but which also serves as input for other applications using location information.

When investigating the type of adaptation related to context elements, this passive vs. active role of the context and the content vs. application orientation of the context can be used as aid. For all context elements identified, the following questions should be answered using the variation points from step 4.2:

- Does the context element influence the behavior of the context-based application only or also an external "context-aware" applications? (active role; internal and/or external)
- Does the context element influence the information provision or the application behavior or both?
- Do the attributes of the context element have to be updated by the context-based application only or could there be a need to also use external "context-aware" applications?

The above questions would result in a classification for each context element on the one hand side into internally relevant and updated in the context-base application or (also) externally relevant and updated. On the other side there is a classification into relevant for behavior adaptation or relevant for information provision. This classification helps during software design of the context-aware application for deciding on operations on context elements and their external visibility and related interfaces.

4.4 Implementation of context-based application

The next step in our approach is the implementation of a context-based application using the initial context model from 4.3. This step basically is not elaborated in our approach, since it usually includes a software development process and many software development approaches exist which could be integrated (see [17] for an overview). From the software development process viewpoint, context modeling can be considered as part of the requirements specification task or as part of the early software design task.

However, using the context model for implementing an application based on it is part of the validation of this model and will give valuable and necessary feedback regarding required improvements and utility of the model. This is why the “link” to software development was included as an explicit step in our approach. As a result of this step, experiences from using the context model including improvement requirements or a confirmation of the context model’s utility are expected.

4.5 Iterative improvement

As discussed in section 2, context models are used in various application domains with different needs, i.e. different context types exist, some of these types are used in several applications and we expect more types and applications for these types to emerge in the future. Thus, development processes for context models include the development of the type of a context model – in case this does not yet exist – and development of actual context models of this type. In order to reach a high “fitness for purpose” of the context model types and utility of the actual context models we propose an iterative approach, which resembles Boehm’s well-known “spiral model” [12] for development of software systems and was inspired by this work. Advantages attributed to the spiral model are early validation and continuous improvement of artefacts developed in the process. From our perspective, these characteristics are very useful for collecting feedback from developers of context-based applications and reaching a high maturity of context models. Our proposal is to develop context models in an iterative way consisting of alternating development and validation steps for both, model type and actual model. Figure 3 illustrates the overall approach.

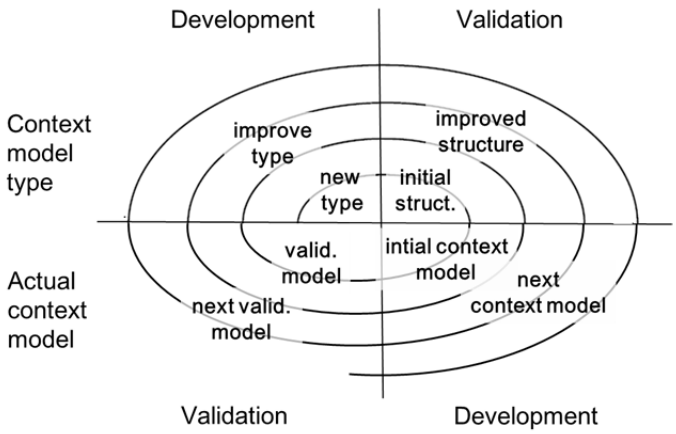

Figure 3: Iterative process of development and validation

In this context, the initial development of a context model type (step 1; marked as “new type” in Figure 3) is performed according to the steps described in sections 4.1 to 4.3. The next step should be the validation of the initial structure (step 2; “initial struct.” in Figure 3) consisting of checking internal consistency and soundness. The next step recommended is to apply the context model type by building an initial version of the context-based application, i.e. the next step includes developing an actual context model (step 3, “initial context model”). Afterwards, the initial model also has to be validated by using the context-based applications (step 4, “valid. model”). Steps 1 to 4 form the first iteration. The second iteration would then improve both, type and actual model.

4.6 Theory and Practice Validation

The validation of context model type and actual model has to be performed including developers and users of context-based applications, and encompassing both theory and practice. Among the many scientific approaches for validating qualitative research results, we base our proposal for validation activities to be performed on the work of Lincoln and Guba [13, p. 289 ff.] on “naturalistic inquiry”. On the one hand, we distinguish between theoretical and practical validation. Theoretical validation means assessing an approach within the theories of the domain the approach is part of or supposed to contribute to. For context type validation, this means to assess the soundness, feasibility, consistency within the body of knowledge in, for instance, computer science and information systems. Practical validation encompasses all kinds of application of the context model for validation purposes, which requires defined procedures and documenting results. This could be simple lab examples illustrating the approach, controlled experiments in a lab setting, application in industrial cases, etc.

On the other hand, we consider the context of validation and distinguish between validation by the developers of the approach in their internal environment, validation by the developers outside the internal environment, and validation by other actors than the developers. Combining these two perspectives leads to a two by three matrix, which is depicted in Table 1. The cells of this table show typical ways of validation for the different combinations of the two perspectives.

Table 1: Proposed validation steps for context models

	Theory	**Practice**
Internal, development team	Validation against state of research	Prototype implementation, test in lab environment
External, in validation context	Peer-review, comparison to known best practices	Case studies for evaluation purposes
External, in application context	Development of extensions by external actors	Use of the artifacts developed for solutions

Using the above matrix, the different iterations of the context model development described in section 4.5 should proceed from theory to practice and internal to external validation. Thoroughly validated context types will include all parts of the matrix and involve several iterations.

5 Application in KOSMOS project

The context modeling approach presented in section 4 has been applied in the KOSMOS project in order to validate feasibility and usefulness, and to gather first experiences and hints how to improve it. The KOSMOS project aims at attracting new target groups to university education and to develop and explore new study formats. New target groups and formats need an adjusted or different kind of support by learning management systems compared to the traditional target groups, since didactic and pedagogical concepts also differ. In order to facilitate this adaptivity requirement, our approach is that learning management systems (LMS) should be flexibly adaptable to the learner's individual demands when it comes to contents and applications supporting the learning process. We consider a context-aware LMS a suitable technical implementation of this requirement. In KOSMOS, this LMS is supposed to be a portal integrating existing and future learning objects and tools supporting the different learning phases. This portal is called "myKOSMOS". The development process of myKOSMOS is performed according to the steps introduced in section 4 and is described in the following. However, the application of myKOSMOS is not subject of this paper.

Step 1: Scenario Modeling

For this first step of scenario modeling we used an approach from enterprise modeling based on Troux Architect as a tool and Troux Semantics as notation. Consequently, we modeled the different planned ways how myKOSMOS would be used by the future users. This resulted in process model-like scenarios, as depicted in **Figure 4** showing the example "assignment work in distance learning".

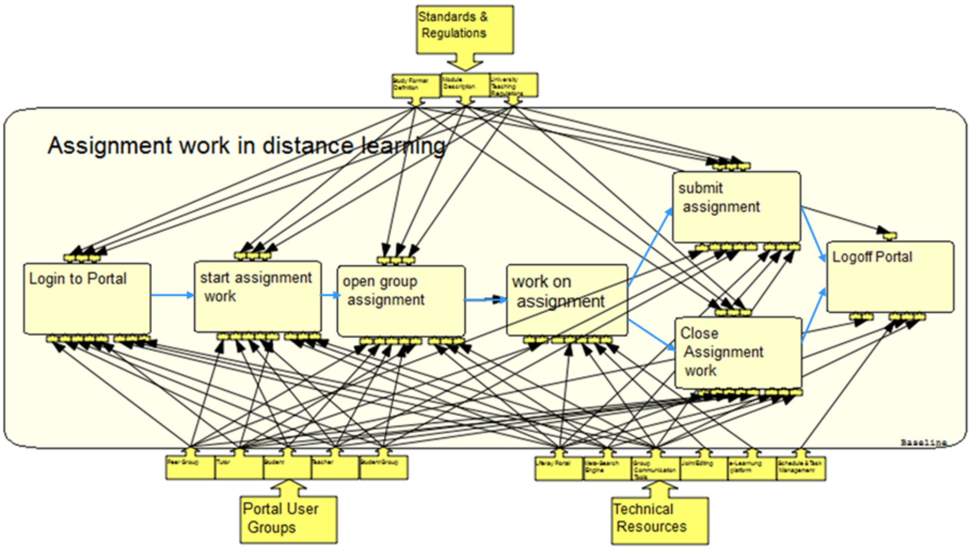

Figure 4: Scenario model for "assignment work in distance learning" to be supported by the context-based portal myKOSMOS

Step 2: Variability Elicitation

The second step is to identify variation aspects and variation points. As a starting point for this activity, we briefly describe the scenario in order to show the variations: the scenario starts with the student logging in. According to his profile he is provided with an individually configured entry page, making offers for his learning process. The variation here is due to his study format as well his individual preferences captured during different sessions. Following his course of study, completing different modules within the study format, the student choses to open or proceed with his assignment work for a certain module, which is loaded presenting the recent state of his work in progress. Once having caught up with his recent results, the student is confronted with different tasks to be fulfilled in order to fulfill the assignment, however is free to choose which task to pick. A regular assignment the designed study formats includes information research the portal supports providing the appropriate sources for the study format. In addition many assignments also involve the communication with fellow students since they are assigned group work. In the process the work should be documented to be handed in, where the kind of documentation being determined in the assignment description. During the work process coordination issues between the team members should be resolved as well, which might be due to the individual time tables and working hours, as well as the specific interests or responsibilities within the task assignment. At the end of each session the user has the choice between submitting his work for the correction process and simply closing it to proceed in another sessions. The consequence at the end of the session would be a log-off which is accompanied by a profile update due to the user's behavior during the session.

Variation aspects. According to the above description, the following component types caused variations: activity (for capturing the portal use processes), user group (like assignment group, study format group), documentation type (Word document, interactive document, learning journal entry etc.), application type (communication support, groupwork support, editing support, search, etc.). For all component types, the characteristics of the type decisive for the variation is the actual instantiation, i.e. what user group logged in, what type of documentation is used etc.

Variation Points. The scenario model for myKOSMOS included the following variation points: login to portal (variation due to user group and study format of the user logging in), start assignment work (variation due to the type of assignment given in the module description), open assignment (variation due to the actual student group working on a specific assignment), work on assignment (variation due to status of the actual work) and open group assignment (variation due to the tool support available).

During analysis of the entities we discovered that also the need to distinguish the type of learning task (e.g. assignment, exercise, lecture, etc.). These tasks were represented by the different scenario models.

Step 3: Develop initial context model

After identifying the variation aspects and variation points we combined them into the necessary context elements and their classification within the scenario as described in section 4.3. The result of this step is provided in Table 2.

Table 2: Context Elements for myKOSMOS

Context element	*Context element attributes*	*Type of adaptation*	*Originated from variation aspect*	*Originated from variation point*
Study format	Name, modules assigned	Active, content and behavior	Type: portal user group, char.: study format	“Login to Portal”
Module	name, assignments	Active, content and behavior	Type: standards & regulations, char.: module description	“Start assignment work”
User Group	Name, members, assignments, Preferences	Active/passive, behavior and content	Type: Portal user group, char.: student group	“Open group assignment”
Assign-ment	Description, deadline, type	Active/passive, content and behavior	Type: learning tasks, char.: assignment	“start assignment”
Appli-cation support	Description	Active/passive, content and behavior	Type: activities, char.: kind of activity	“work on assignment”
Prefe-rences	Application type	Active/passive, behavior	Type: technical resources, char.: group commu-nication tools	“Open group assignment”

Step 4: Implementation of context-based application

Using the initial context model with its identified elements, we started implementation phase of the portal myKOSMOS. The main effort of this phase lies in transferring the context model into a data model. The implementation takes places via a Liferay development which will be extended by a context processing component.

Step 5: Alternating model-instance improvement

Following the initial implementation of myKOSMOS a validation is necessary. Already in the process implementation minor adjustments are done due to implementation specifics, as e.g. the preferences as such have to be refined to be captured from the behavior of the user. This part of the improvement means successive refinements by the concretization of the scenarios under the implementation process.

Step 6: Validation phase

Finally the rigorous validation of the context model is necessary. Referring to the validation phases as mentioned in section 4.5 the internal and theory related validation is finished with the end of the modelling. The internal and practice related validation is ongoing being closely connected with the successful implementation of the finished context-aware application.

6 Conclusion

The goal of the paper was to develop a ‘good practice’ procedure for identifying and capturing all necessary context elements for context-based applications. The approach presented in section 4 was derived from an analysis of previous context modeling cases and has gone through an initial validation in the myKOSMOS project. The approach is based on scenarios and can be run in different manners. One example is to start with one initial scenario, and build a context following this one scenario only. Another example would be to add further scenarios, which makes the context more complex from the beginning, but certainly allows for a more extensive validation.

Our experiences during the validation of the method by application in the KOSMOS project included that the differentiation between variation aspects and points turned out to be most difficult for the creation of the context model. Furthermore the transfer of the context model to an explicit data structure suitable for implementation was labor-intensive using the style of modeling with Troux Semantics as shown here.

Future work will have to include theoretical and practical aspects. From a theoretical perspective, we aim at further formalizing the concept of variation points and variation aspects and how the variation has to be reflected in the context model. The classification in content and behavior aspects and in internal and external effects seems useful, but is not yet clear enough. Furthermore, the transition from the context model into an implementation of the model has to be further investigated with the objective to support the design of software components implementing the context concept and the envisioned behavior captured in the context model. Furthermore, the integration of the context modeling approach and software engineering processes should be further investigated. Since a model representation of context can be part of the early design of the information system to be developed, it might also influence the architecture of the overall system. Furthermore, the model carries requirements which need to be taken into account during the development process. These aspects need further exploration.

From the practical perspective, implementation of the myKOSMOS context component, using it in selected study formats and collecting improvement potential and experiences during the usage will be an important future activity. In accordance to steps 4 and 5 of our context modeling approach, we aim for finishing the first complete iteration of type development and implementation before starting an improvement cycle and we expect to collect sufficient experiences from internal practical validation to be able to continue with external validation.

Acknowledgements

The work presented in this paper was supported within the project KOSMOS (Konstruktion und Organisation eines Studiums in Offenen Systemen) funded by the BMBF (Federal Ministry of Education and Research, Germany) and the European Social Funds of the European Union.

References

[1] Dey, Anind K.: Understanding and Using Context , Personal and Ubiquitous Computing 5, no. 1 (February 20, 2001): 4-7.

[2] Hoffmann, P. (2008): Context-based Semantic Similarity across Ontologies. PhD-thesis. Claude Bernard University of Lyon, December 2008.
[3] Finkelstein, L., E. Gabrilovich, Y. Matias, E. Rivlin, Z. Solan and G. Wolfman (2001): Placing search in context: the concept revisited. Proceedings of the 10th international conference on World Wide Web (2001): 406-414.
[4] Brézillon, Patrick (2007): Context Modeling: Task Model and Practice Model. LNCS 4635, pp. 122- 135. Springer Berlin / Heidelberg.
[5] Henricksen, K., Indulska, J. and Rakotonirainy, A. (2002): Modeling con context information in pervasive computing systems; Pervasive ’02: Proceedings of the First International Conference on Pervasive Computing; Pervasive ’02: Proceedings of the First International Conference on Pervasive Computing; 167–180; Springer-Verlag, London, UK .
[6] Wang, X., Zhang, D., Gu, T. and Pung, H. (2004): Ontology based context modeling and reasoning using owl; 18–22.
[7] Bazire, M. and Brézillon, P. (2005): Understanding Context Before Using It, in Hutchison, D., Kanade, T., Kittler, J., Kleinberg, J.M., Mattern, F., Mitchell, J.C., Naor, M., Nierstrasz, O., PanduRangan, C., Steffen, B., Sudan, M., Terzopoulos, D., Tygar, D., Vardi, M.Y., Weikum, G., Dey, A., Kokinov, B., Leake, D. and Turner, R. (Eds.), Modeling and Using Context, Lecture Notes in Computer Science, Vol. 3554, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 29–40.
[8] Mena, T., Bellamine-Ben Saoud, N., Ahmed, M. and Pavard, B. (2007): Towards a Methodology for Context Sensitive Systems Development, in Kokinov, B., Richardson, D.C., Roth-Berghofer, T.R. and Vieu, L. (Eds.), Modeling and Using Context, Lecture Notes in Computer Science, Vol. 4635, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 56–68.
[9] Shvaiko, P., Euzenat, J. (2008): Ten challenges for ontology matching; OTM Conferences (2); 1164–1182.
[10] Levashova, T., Lundqvist, M., Sandkuhl, K. and Smirnov, A. (2006): Context-based modelling of information demand: Approaches from information logistics and decision support; Proceedings of the 14th European Conference on Information Systems.
[11] Sandkuhl, K.; Stirna, J.; Persson, A. and Wissotzki, M. (2014): Enterprise Modeling–Tackling Business challenges with the 4 EM method. Springer Verlag, Berlin Heidelberg.
[12] Boehm, B. W. (1988) A spiral model of software development and enhancement. IEEE Computer, 21 (5), 61-72.
[13] Lincoln, Y. and Guba, E. (1985): Naturalistic Inquiry. Sage Publications.
[14] Sandkuhl, K.; Koc, H. and J. Stirna (2014) Context-aware Business Services: Technological Support for Business and IT-Alignment. 5th International Workshop on Business and IT-Alignment @ BIS 2014, Larnaca, Cyprus. Accepted for publication in LNBIP Workshop proceedings. To be published in October 2014.
[15] Meissen U., Pfennigschmidt S., Sandkuhl K., Wahnfried T. (2004) Situation-based Message Rating in Information Logistics and its Applicability in Collaboration Scenarios. Euromicro 2004 Special Session on "Advances in Web Computing", August 31- September 3, IEEE Computer Society Press, 2004.
[16] K. Sandkuhl, U. Seigerroth, V. Tarasov (2012) Context Modelling for Decision Making in Competence Supply: Experiences from a Military Case. In: L. Villa-Vargas, L. Sheremetov, and H.-D. Haasis: Operations Research and Data Mining ORADM 2012 workshop proceedings. March 2012, Cancun, Mexico. Mexican Petroleum Institute, Mexico City, 978-607-414-284-6.
[17] Sommerville, I. (2010) Software Engineering, Edition 9. Pearson. ISBN: 9780137053469.
[18] Davenport, T.H. ; Prusak, L.: Working Knowledge. Boston:Harvard Business School Press, 1998.