# Designing Ubiquitous Information Systems based on Conceptual Models

Sabine Janzen, Andreas Filler, Wolfgang Maass

{sabine.janzen,andreas.filler}@hs-furtwangen.de,
wolfgang.maass@iss.uni-saarland.de

**Abstract:** Ubiquitous Information Systems (UIS) support single actors and groups by services over ubiquitous computing technologies anywhere and anytime. These systems require design approaches that keep a holistic view of situations in which single users and groups interact with one another and with accessible services. We introduce and exemplify the *Situational Design Methodology for Information Systems (SiDIS)* that uses three types of Conceptual Models (CMs) and corresponding translation procedures. This contribution focus on the specification and translation of the CMs as well as their processing by the resulting UIS.

## 1 Introduction

Design teams for Information Systems (IS) are heterogeneous; they consist of members from different fields such as domain experts, various users, and IT architects. Ideas and expertise of these members have to be brought together for building a homogeneous understanding of an IS. Although, means for communication and explication are required for building these common understandings on various levels. Whereas non-technical members intend to build an IS that supports for instance their social needs; technical members focus on engineering aspects of the technical realization of the system. Such shared understandings of design teams can be described by various conceptual models (CM) that are used during design phases of IS development [WMPW95].
The focus of conceptual modeling lies in the identification of important concepts and relations [MMG02, WMPW95] that are semantically described by shared vocabularies [MS95]. These vocabularies are either implicitly defined as part of understanding in a community or explicitly defined in forms of machine-processable representations [Cha90]. The latter have the advantage that the logic of a CM can then be evaluated and matched with other CMs which is an important aspect for re-use [PHS03]. CMs enable an abstraction from technical issues and focus on aspects of situations in which users and user groups perform activities that are supported by information and communication services [WMPW95]. A CM is represented by a conceptual modeling language (CML), such as Entity-Relationship [Che76] models or the Unified Modeling Language (UML) [BRJ05]. From an IS development process perspective, CMs can be used during analysis, design, and realization phases [WMPW95].
Within real-world situations, Ubiquitous Information Systems (UIS) support single actors

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

and groups by services over ubiquitous computing technologies anywhere and anytime [LY02a, Yoo10]. This class of IS is dominated by ad-hoc methods, e.g. "wild-west" prototyping [LY02b, Yoo10]. But, UIS require design approaches that keep a holistic view of situations in which single users and groups interact with one another and with accessible services. Little research has been done so far on design methodologies and conceptual modeling for UIS [JKM10, KL09]. In the following, a design methodology for UIS is presented that uses three types of CMs and corresponding translation procedures [MJ11]. We will exemplify the CMs as well as their translation and their processing by the resulting UIS.
After introducing the design methodology and an exemplary use case (Section 2), we will elaborate the application of the methodology regarding the three CMs according to the use case (Section 3 and 4). Next, the processing of CMs by the UIS is exemplified. Results and future work close this article (Section 6).

## 2 Designing Ubiquitous Information Systems

As mentioned before, designing UIS does not exclusively depend on technical issues but also on aspects concerning users, social interactions, and physical surroundings amongst others. Previous design science research identified seven development principles for the design of information systems that should be addressed by a design methodology [JKM10]. Based on these principles, we derived a design methodology for UIS, called *Situational Design Methodology for Information Systems (SiDIS)* [MJ11]. SiDIS consists of four

Figure 1: Physical environment of the intelligent bathroom case

phases: (1) Identification of Problem and Needs, (2) Design of Solution, (3) Development of Solution and (4) Evaluation of Solution. These phases consist of nine tasks: (1) Identification of problem and needs, (2) Derivation of situations (narratives), (3) Derivation of diagrammatic CMs, (4) Evaluation of diagrammatic CMs, (5) Derivation of formal

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

 

propositional CMs, (6) Formalization of system design, (7) Implementation of formalized system design, (8) Evaluation of solution, and (9) Product development. SiDIS was tested in various UIS development projects. Currently, the methodology is used for designing an UIS for an intelligent bathroom case (cf. Figure 1) within the EU project IKS[1]. The intelligent bathroom as UIS represents a "far out" vision of IKS for direct user interactions with embedded contents organized by a "Semantic CMS Technology Stack", and combines advanced content and knowledge management with an ubiquitous computing scenario in a place everybody is familiar with - the bathroom. The use case shows how users can interact with contents in physical environments in a way that leaves the dimension of "small windows to the infosphere" as known by the "monitor paradigm" [JMK+10]. A detailed description of the design methodology is part of other contributions [JKM10, MJ11] and beyond the scope of this paper. In the following, we will elaborate the application of SiDIS by means of the intelligent bathroom case. We will focus on the derivation of diagrammatic CMs, their translation into propositional CMs and the processing of these CMs by the system (task 3, 5, and 7).

## 3 Derivation of Diagrammatic Conceptual Models

Information systems are compositions of social system, information system, and service system that uses information technology infrastructures for realizing desired situations [LK03, LS00, OB01]. The *Abstract Information System Model (AISM)* combines these three classes for CMs of IS with the additional level of physical entities that is required for UIS [MJ11]. The model consists of Information Sphere, Social System, Service System and Physical Object System. The Information Sphere covers all information objects used within the information system. Second, the Social System consists of a set of roles described by a set of attributes, e.g., rights, obligations; as well as actions performed by role-taking actors. The Service System represents all services that are available within usage situations of the system. The specific Physical Object System covers the set of physical entities available within all situations in which a UIS can be used. According to the application of SiDIS within the bathroom case, we conducted design workshops with domain experts, more precisely with a manufacturer of high-class bathroom furniture (cf. SiDIS Task 1). Within these workshops, we identified usage situations of the future bathroom UIS.
The identified situations were translated into 12 narrative CMs (cf. SiDIS Task 2). Narrative CMs are the first CM type of our methodological approach consisting of three types of CMs. As a substitute for the 12 narrative CMs, we selected one that shall be used as an example to present the application of SiDIS:

*"It's Thursday morning. I get site-specific weather information when I am brushing my teeth in the bathroom. Based on weather information and my calendar, free-time event suggestions are given (e.g. "Today, 8 p.m. - Sneak Preview at CinemaOne."). Do you want to order tickets?"*

[1] See http://www.iks-project.eu/

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4 weitere Artikel online:
http://informatik2011.de/519.html

We assume that narrative CMs are effective means for building and understanding situations of UIS because of their ability to provide discourse information and sequences of interactions between actors [HH97, KV08]. Next, the narrative CMs were translated into the second type of CM, the diagrammatic ones - called Pre-Artifacts (cf. SiDIS Task 3). Based on the AISM, Pre-Artifacts emphasize requirements on social structure, information objects, physical objects, and services in usage situations.

### 3.1 Defining Pre-Artifacts

When deriving diagrammatic CMs based on narrative CMs, the modeling person has to combine and instantiate specific Pre-Artifact patterns similar to the notion of design patterns as used in architecture [Ale78] and Software Engineering [DA99]. Thereby, the range of modeling opportunities is restricted and canalized in the sense of a better guidance. We specified seven Pre-Artifact patterns (cf. Figure 2) whereas three shall be elaborated exemplary in the following [MJ11]:

**(P1) RoleInteraction Pattern:** This pattern describes a situation in which two or more role-taking actors interact with one another by exchanging information objects supported by an interface service, e.g., mail communication between sender and receiver. To express the difference between services that represent direct interfaces to users and services that operate on an internal level exclusively, the notion of *interface services* and *internal services* is used. The interaction between roles is described by a generic property called r-interacts. The interface service is only used as a communication channel.

**(P4) ServiceInteraction Pattern:** This pattern describes the interaction relationship of two interface or internal services with no interaction with human actors. Within this interaction that is represented by s-interacts, an information object is used. The interaction relationship between services is described by *s-interacts* while roles are connected by *r-interacts* as mentioned before. For instance, a local temperature service sends data to a central weather service. In contrast to the Role Creates Information Object pattern, this pattern supports system designs that do not use role-based designs on service level.

**(P7) RoleCreatesInformationObject Pattern:** By this pattern a service creates an information object by taking a role which links an information object to a service. This pattern supports role-based system designs. For instance, a vital sign monitoring system can take a role that allows it to create emergency alerts. Created alerts are directly linked with this service via a role.

The construction of Pre-Artifacts is guided by a method consisting of five steps regarding the instantiation and combination of Pre-Artifact patterns: (1) Definition of information objects in Infosphere, (2) Definition of user-system or user-user interactions related to information objects, (3) Definition of roles taken by services, (4) Definition of supporting internal services, and (5) Definition of user initiative. Each step proclaims specific Pre-Artifact patterns that help to achieve the objectives of the step [MJ11]. In the following, we will exemplify the five steps by means of our exemplary narrative CM. Note, that in case of high complexity of a narrative CM, multiple Pre-Artifacts are generated to avoid

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

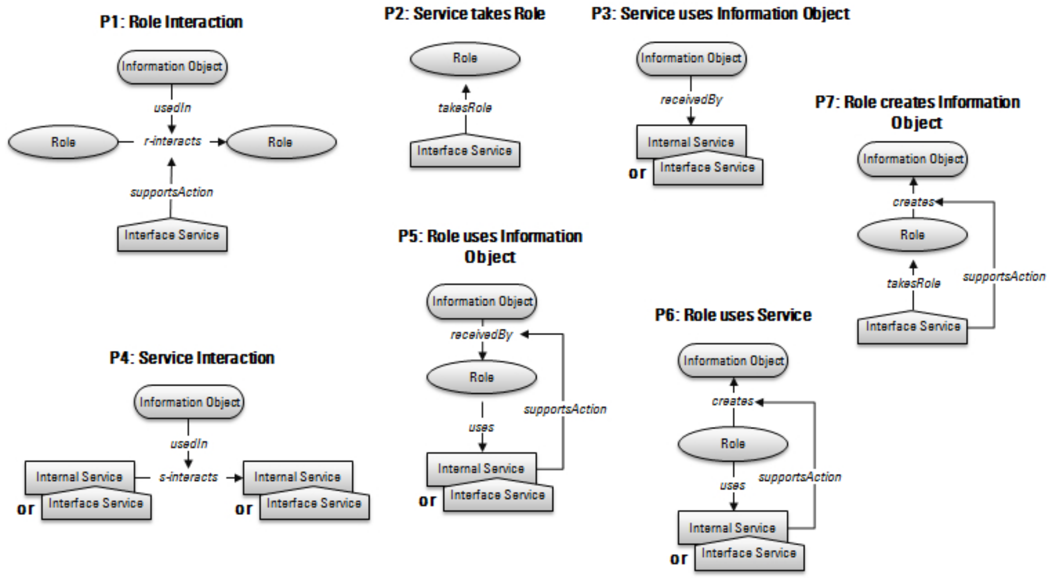


Figure 2: Pre-Artifact patterns

an overloading of a single diagrammatic structure. Our exemplary narrative CM is split into three parts; we will use the first part:
*"It's Thursday morning. I get site-specific weather information when I am brushing my teeth in the bathroom."*

### 3.2 Exemplary Translation of Narrative CM into Pre-Artifact

**Step 1: Definition of information objects in Infosphere.** All information objects that occur in the narrative CM have to be defined as information objects in the *Infosphere*. Attention should be paid to the aspect that information objects that will be created in a situation always have generic sources. Figure 3 shows that the modeling person has specified the goal "Getting weather information for user's location" that is assigned to the user in the situation. Furthermore, the information object *site-specific weather information* is defined. This information object has to be created in the situation based on the required information objects *global weather information* and *location*.

**Step 2: Definition of user-system or user-user interactions related to information objects.** Within this step, interactions between users or user and system related to new generated information objects have to be defined. These interactions take place between roles in the *Social System* exclusively. Interactions between user and system are always supported by a service of the *Service System*. The requirements of this step are fulfilled by the application of the *RoleInteraction* pattern exclusively. In our exemplary Pre-Artifact


ISBN 978-3-88579-286-4

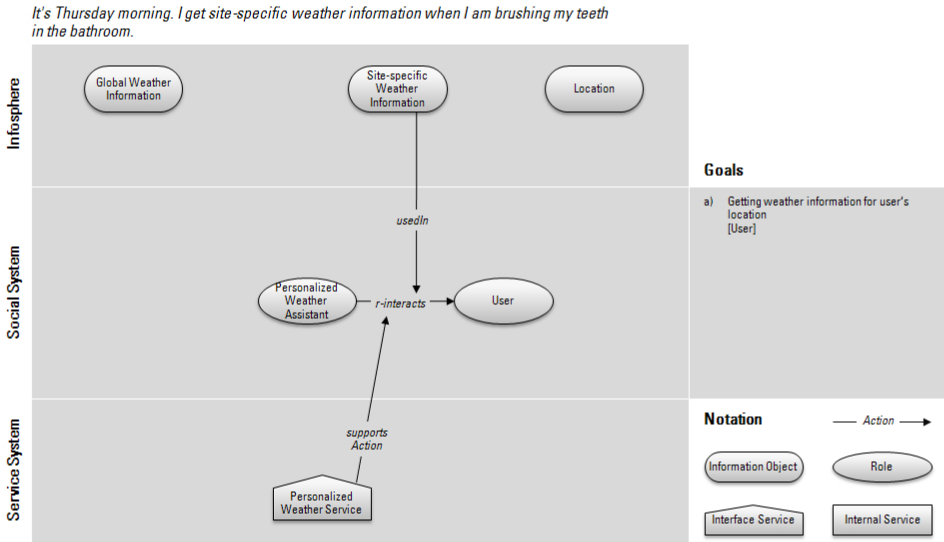


Figure 3: Definition of user-system interaction related to information objects

(cf. Figure 3) an interaction between a *Personalized Weather Assistant* and the *User* was modeled that is supported by a *Personalized Weather Service*. Subject of the interaction is the information object *site-specific weather information*.

**Step 3: Definition of roles taken by services.** Next, an interface service has to be defined that takes a role for creating the new information object that will be used in the interaction. Therefore, the service has to take a role in the interaction. To manage this step, the *RoleCreatesInformationObject* pattern is applied to define the creation of the information object by a role taken by a service. In our example, the *Personalized Weather Service* takes the role of the *Personalized Weather Assistant* that creates the information object *site-specific weather information*. The interface service supports this action indirectly (cf. Figure 4). To express the plain role-taking by a service without a creating function, the *ServiceTakesRole* pattern can be applied.

**Step 4: Definition of supporting internal services.** To create new information objects, generic information sources are needed as mentioned before. The interface service that supports the creation of a new information object needs access to these sources. Therefore, internal services for all remaining information objects in the *Infosphere* are specified. The interaction between services regarding the information objects is realized by applying the *ServiceInteraction* pattern. The exemplary Pre-Artifact (cf. Figure 5) shows the definition of two internal services *Weather Service* and *User Context Service* that feed a *Personalized Weather Service* with global weather information and location data.

**Step 5: Definition of user initiative.** If a user role initiates an interaction with the system that means using the system in a proactive way, this situation is modeled by using the *RoleUsesService* or *RoleUsesInformationObject* pattern (not required in the example). The role uses a service to create or receive an information object, for instance, the user wants to leave a message for another user. This action is indirectly supported by a service.

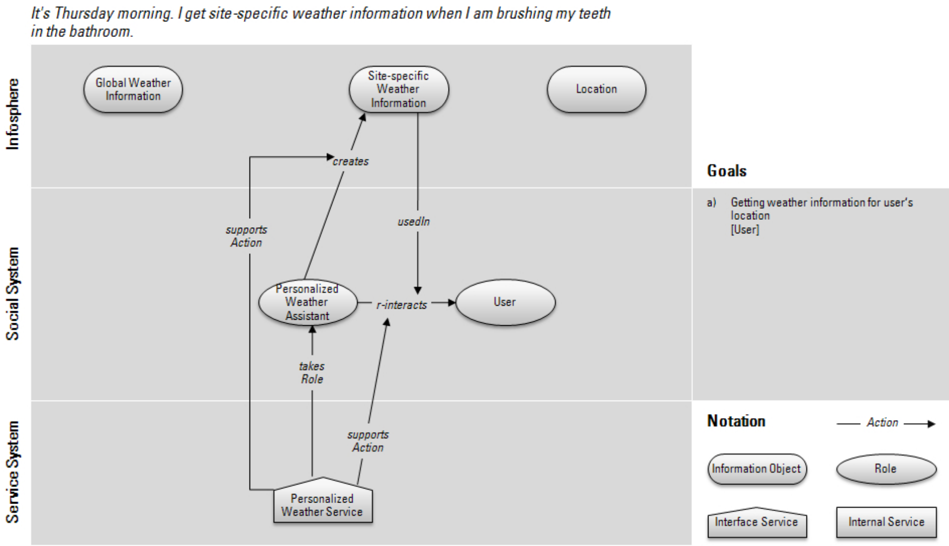


Figure 4: Definition of roles taken by services

## 4 Translation of Diagrammatic into Propositional CMs

After modeling Pre-Artifacts based on the narrative CMs by means of the aforementioned ”5 steps”, the Pre-Artifacts are translated into the third type of CM of our methodological approach - the propositional CMs (cf. SiDIS Task 5). The objective of this translation is the creation of specifications for later system designs [PHS03] as well as machine-processable CMs that can be verified [BKW10]. There are several opportunities for formalization, for instance Unified Modeling Language (UML), entity-relationship model (ER) or a formalization based on ontologies by means of RDF or OWL. Bera et al. (2010) identified some unique features of OWL that are not available in ER model and in UML, e.g., OWL is implementable that means OWL ontologies are machine-readable, and thus computational. Furthermore, OWL constructs are independent, i.e. classes can exist independent of instances or properties and properties are independent of classes. Beside these advantageous features of OWL, there are also difficulties in using OWL for the formalization of Pre-Artifacts. Bera et al. (2010) determine that there are no clear rules how to map from domain information as represented by Pre-Artifacts to OWL constructs similar to the intended propositional CMs [MJ11, MSK11].
There are at least three approaches of translating Pre-Artifacts into propositional CMs by means of OWL constructs. Each option was tested by modeling three exemplary Pre-

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

 

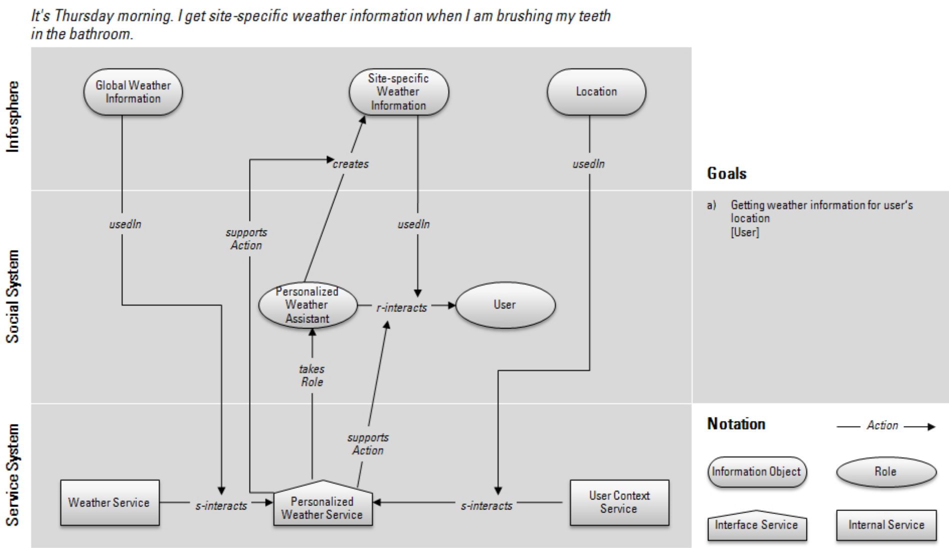


Figure 5: Definition of supporting internal services

Artifact patterns *(Role Interaction, Role creates Information Object, Service Interaction)* as pattern ontologies that were used for generating the propositional CM of our exemplary Pre-Artifact (cf. Figure 5) [MJ11]. Comparing the results of the diverse approaches, we decided to use a "philosophical ontology" [BKW10] to derive guidelines on how OWL constructs can be applied in the modeling of propositional conceptual models. A *Pre-Artifact Model* was created that represents a "vocabulary" and generic object properties of Pre-Artifact patterns. The Pre-Artifact Model consists of 12 concept types and 8 generic object properties. It represents basic entities of AISM: *InformationObject, Role* and *Service* with the sub-classes *Interface* and *Internal Service*. Furthermore, a super-class *Action* was defined that contains further sub classes that specify diverse types of pattern actions: *Creation, Receiving* and *Interaction* with the sub classes *R_Interaction* and *S_Interaction*. The decision to model most of the pattern relations by means of additive concepts is due to the fact that these relations represent three-way connections. A second opportunity would be to use property chains in OWL 2 that support transitive relationships between objects [MPSG09]. The advantage of the former opportunity lies in integrating actions as specific concepts in the social system. This allows a differentiated consideration and extensibility by further properties. Furthermore, the model consists of 8 generic object properties: *initiatesInteraction, finalizesInteraction, initiatesAction, isResultOfAction, supportsAction, takesRole, usedIn* and *usesService*. Each pattern ontology imports this Pre-Artifact Model. But, for the specification of pattern-specific object properties based on the generic properties of the model, inheritance structures of object properties are used. That means each pattern defines sub properties of the relevant object properties imported from the model. Therefore, super-properties and concepts of the Pre-Artifact Model remain unchanged. In this context, the OWL feature is used, that OWL constructs are independent, i.e. proper-

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4 

ties can exist independent of classes [BKW10]. Based on this approach, clear assignments of specified object properties to specific patterns are realized. Conceptual modelers will be supported by modeling guidelines because of a canalization of modeling options. The propositional conceptual model can be modeled in an incremental way by importing patterns step by step according to the requirements of the Pre-Artifact.

### 4.1 Exemplary Translation of Pre-Artifact into Propositional CM

When translating our exemplary Pre-Artifact (cf. Figure 5) into a propositional CM, we also use the expressiveness of Pre-Artifact patterns represented as single ontologies. After generating an empty OWL file in an ontology development tool, e.g., Protégé, required Pre-Artifact patterns can be imported by their URL. Similar to the procedure of defining Pre-Artifacts in five steps, we start with applying the *RoleInteraction* pattern and import its formalized model (cf. Figure 6). Then, we instantiate the relevant concepts of the

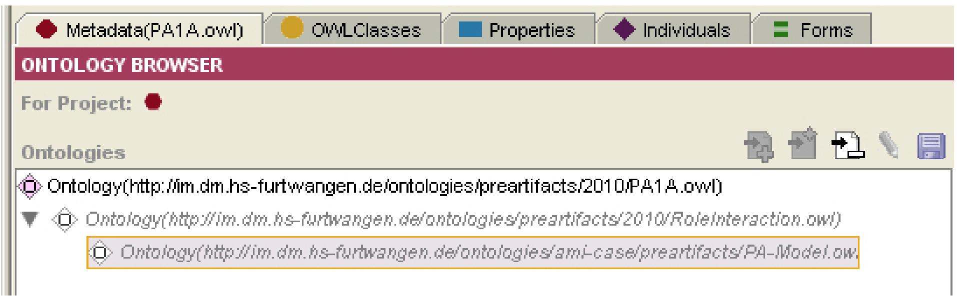


Figure 6: Import of Pre-Artifact pattern RoleInteraction

pattern, i.e., we create instances of the concept ”Role” named ”User” and ”PersonalizedWeatherAssistant” (cf. Figure 7). To represent the interaction between User and PersonalizedWeatherAssistant, an instance of the concept ”R-Interaction” has to be created. For linking both roles with the instance of R-Interaction (cf. Figure 8), the formalized pattern offers the specified object properties ”initatesR_Interaction” and ”finalizesR_Interaction” that inherit from the super-properties ”initiatesInteraction” and ”finalizesInteraction” (cf. Figure 9).The following OWL snippet clarifies this aspect and links a “User” Role with a Role Interaction object.

```
<owl:ObjectProperty rdf:about="http://im.dm.hs-furtwangen.de/
    ontologies/preartifacts/2010/RoleInteraction#
    initiatesR_Interaction">
        <rdfs:domain rdf:resource="http://im.dm.hs-furtwangen.de/
            ontologies/ami-case/preartifacts/PA-Model.owl#Role"/>
        <rdfs:range rdf:resource="http://im.dm.hs-furtwangen.de/
            ontologies/ami-case/preartifacts/PA-Model.owl#
            R_Interaction"/>
```

```
        <rdfs:subPropertyOf rdf:resource="http://im.dm.hs-
            furtwangen.de/ontologies/ami-case/preartifacts/PA-
            Model.owl#initiatesInteraction"/>
</owl:ObjectProperty>

 [...]
<owl:Thing rdf:about="#User">
        <rdf:type rdf:resource="http://im.dm.hs-furtwangen.de/
            ontologies/preartifacts/2010/PA-Model.owl#Role"/>
        <RoleInteraction:finalizesR_Interaction rdf:resource="#
            R_Interaction1"/>
</owl:Thing>
```

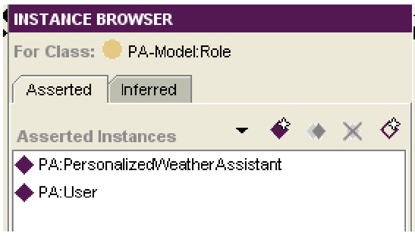


Figure 7: Instantiation of roles - User and PersonalizedWeatherAssistant

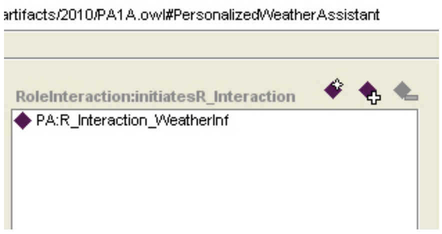


Figure 8: Instantiation of R-Interaction that is initiated by PersonalizedWeatherAssistant


Figure 9: Role Interaction pattern specifies object property

The super properties of the Pre-Artifact model are filled automatically without touching them explicitly when working with the pattern-specific relations. Within the proceeding formalization, the formalized patterns *Role uses IO* and *Service Interaction* are imported. This leads to further specifications of the super-properties of the Pre-Artifact model and thereby to an enhancement of the expressiveness without "getting lost". The following OWL snippet shows the representation of the interface service "PersonalizedWeatherService" modeled by using three Pre-Artifact patterns. The result of the formalization is an OWL description that represents the exemplary Pre-Artifact as well as the original narrative CM in a formal and computational way. According to the fact, that the steps of the formalization are on par with the steps of the definition of the diagrammatic CM, there should be no challenge to execute the formalization in an automatic way, e.g., with a graphical modeling tool based on semantic technology infrastructures.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

```
<Model:InterfaceService rdf:about="#PersonalizedWeatherService">
        <rdf:type rdf:resource="http://www.w3.org/2002/07/owl#
            Thing"/>
        <RoleUsesIO:supportsCreation rdf:resource="#Creation_1"/>
        <RoleUsesIO:interfaceServiceTakesRole rdf:resource="#
            PersonalizedWeatherAssistant"/>
        <RoleInteraction:supportsR_Interaction rdf:resource="#
            R_Interaction1"/>
        <ServiceInteraction:finalizesS_Interaction rdf:resource="#
            S_Interaction_3"/>
        <ServiceInteraction:finalizesS_Interaction rdf:resource="#
            S_Interaction_4"/>
</Model:InterfaceService>
```

## 5 Processing of Propositional CMs by the UIS

Within the aforementioned bathroom case, we translated the seven best-performing narrative CMs into Pre-Artifacts and afterwards into propositional CMs. The propositional CMs represent the structure of specific UIS situations consisting of role-taking actors, interactions, services and information objects. That means, they just "highlight" relevant entities that "float" in a large knowledge representation that covers the overall and general context of the UIS (cf. Figure 10). This knowledge representation contains all information

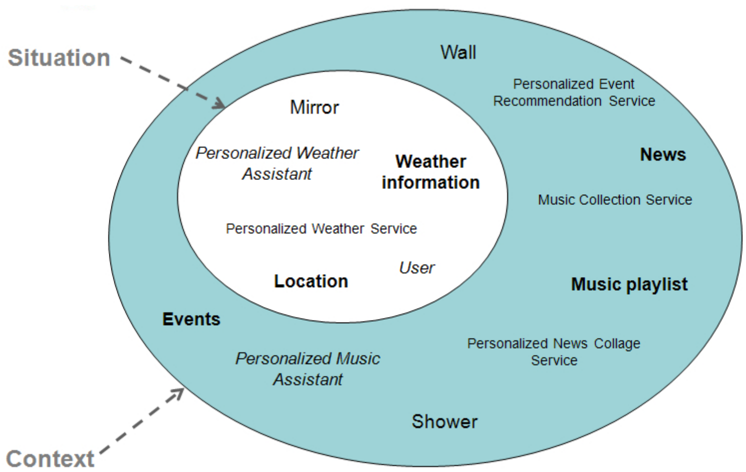


Figure 10: Context and situation regarding knowledge representations for UIS

known by the UIS in the sense of a context representation. The situational representation provided by propositional CMs focus on information that is relevant in a specific situation. Note, information objects, roles, services and physical objects within a propositional CM

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html

represent just pointers to detailed knowledge covered by the context representation. For instance, our exemplary propositional CM declares that in the specific UIS situation an information object ”location” is required and points on the detailed representation of the ”location of the system” in the context representation.

Following the division of context and situation within the knowledge representation, the resulting UIS also processes both knowledge parts in a differentiated way. On implementation level, the situational part consists of *Pre-Artifact Types* and a *Pre-Artifact Instance*. Each propositional CM represents a Pre-Artifact Type consisting of the aforementioned ontological structures classifying specific UIS situations and a related rule set. The Pre-Artifact Instance represents the real situation in the physical environment of the UIS. It is updated according to events taken place in the environment, e.g., a user enters the room. By permanently aligning the Pre-Artifact Instance with the existing Pre-Artifact Types, the UIS is able to determine which Pre-Artifact Type could be the appropriate solution to handle the current context in the environment (cf. Figure 11). The process of determining the appropriate Pre-Artifact Type goes beyond the scope of this paper and will be elaborated in another contribution. The contextual part consisting of the context representation is served by external content services that are requested as required. Furthermore, the context representation is triggered by rules of specific Pre-Artifact Types, e.g., to request the location of the system. The results that means references to specific concepts in the context representation are then stored within the Pre-Artifact Instance. Hence, the Pre-Artifact Instance grows and is adjusted to the selected Pre-Artifact Type. That means the system reacts according to the situational structure that is growing within the Pre-Artifact Instance and tries to fulfill the situation that is pretended by the Pre-Artifact Type.

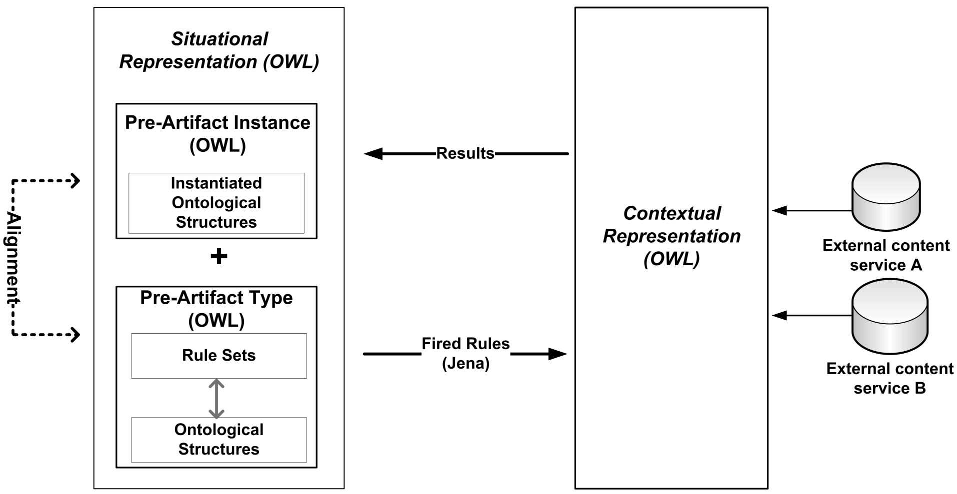


Figure 11: Processing of situations and context by the UIS

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4

## 6 Conclusion and Future Work

In recent years, Ubiquitous Information Systems (UIS) has gained attention. But, this class of Information Systems (IS) is still governed by ad-hoc development methods, e.g. ”wild-west” prototyping [LY02b, Yoo10]. Designing UIS does not exclusively depend on technical issues but also on aspects concerning users, social interactions, and physical surroundings amongst others. Therefore, UIS require design approaches that keep a holistic view of situations in which single users and groups interact with one another and with accessible services. Little research has been done so far on design methodologies and conceptual modeling for UIS [JKM10, KL09].
In this article, we have introduced a design methodology for UIS that uses three types of Conceptual Models (CMs) and corresponding translation procedures - *Situational Design Methodology for Information Systems (SiDIS)* [MJ11]. It was shown how the three types of CMs - narrative, diagrammatic (Pre-Artifacts) and propositional - are used and translated within the design process of a UIS. We exemplified all steps based on a real-world use case and elaborated the processing of the CMs by the resulting UIS. To overcome limitations of design methods for purely digital IS, SiDIS keeps the holistic structure of situations that support an integrated understanding of interactions within complex socio-technical systems. The usage of narrative CMs support joint conceptual views of design members from various domains and helps to ”build bridges”. The translation processes between the three types of CMs that means the structured pattern-based translation of narrative CMs into Pre-Artifacts, and the algorithmic translation of Pre-Artifacts into formal propositional CMs can be executed on semantic technology infrastructures.
In our future work, we will focus on three issues: (1) Verification of sufficiency of SiDIS as well as Pre-Artifacts for designing UIS in further projects in comparison to other approaches, (2) testing whether OWL is sufficient as a formal language for the automatic translation of diagrammatic CM into propositional CM of complex UIS, and (3) implementation of infrastructure to translate three types of CMs automatically or rather with technical support.

## Acknowledgement

This paper is based on work done in the project ’Interactive Knowledge Stack (IKS)’ (FP7 231527) co-funded by the European Commission.

## References

[Ale78] Christopher Alexander. *A Pattern Language: Towns, Buildings, Construction.* 1978.

[BKW10] P. Bera, A. Krasnoperova, and Y. Wand. Using Ontology Languages for Conceptual Modeling. *Journal of Database Management*, 21(1):1–28, 2010.


ISBN 978-3-88579-286-4

[BRJ05] Grady Booch, James Rumbaugh, and Ivar Jacobson. *Unified Modeling Language User Guide, The (2nd Edition) (Addison-Wesley Object Technology Series)*. Addison-Wesley Professional, 2005.

[Cha90] B. Chandrasekaran. Design problem solving: a task analysis. *AI Mag.*, 11:59–71, October 1990.

[Che76] Peter P. Chen. The Entity-Relationship Model: Toward a Unified View of Data. *ACM Transactions on Database Systems*, 1:9–36, 1976.

[DA99] Anind K. Dey and Gregory D. Abowd. Towards a better understanding of context and context-awareness. In *In HUC 99: Proceedings of the 1st international symposium on Handheld and Ubiquitous Computing*, pages 304–307. Springer-Verlag, 1999.

[HH97] Lewis Hinchman and Sandra K. Hinchman. *Memory, Identity, Community: The Idea of Narrative in the Human Sciences*. State University of New York Press, 1997.

[JKM10] Sabine Janzen, Tobias Kowatsch, and Wolfgang Maass. A Methodology for Content-Centered Design of Ambient Environments. In Robert Winter, J. Leon Zhao, and Stephan Aier, editors, *DESRIST*, volume 6105 of *Lecture Notes in Computer Science*, pages 210–225. Springer, 2010.

[JMK+10] Sabine Janzen, Wolfgang Maass, Tobias Kowatsch, Andreas Filler, Massimo Romanelli, Sebastian Germesin, Tilman Becker, Gokce B. Laleci, Ozgur Kilic, Cagdas Ocalan, Erdem Alpay, and Asuman Dogac. IKS Deliverable. D2.1 Report: AMI Case Requirements Specification. Technical report, EU project IKS, 2010.

[KL09] Scott R. Klemmer and James A. Landay. Toolkit Support for Integrating Physical and Digital Interactions. *Human-Computer Interaction*, 24(3):315–366, 2009.

[KV08] William L. Kuechler, Jr. and Vijay K. Vaishnavi. An expert system for dynamic re-coordination of distributed workflows. *Expert Syst. Appl.*, 34:551–563, January 2008.

[LK03] Roberta Lamb and Rob Kling. Reconceptualizing Users as Social Actors in Information Systems Research. *MIS Quarterly*, 27(2), 2003.

[LS00] Ulrike Lechner and Beat F. Schmid. Communities and Media - Towards a Reconstruction of Communities on Media. In *HICSS*. IEEE Press, 2000.

[LY02a] Kalle Lyytinen and Youngjin Yoo. Issues and Challenges in Ubiquitous Computing - Introduction. *Commun. ACM*, 45(12):62–65, December 2002.

[LY02b] Kalle Lyytinen and Youngjin Yoo. Research Commentary: The Next Wave of Nomadic Computing. *Info. Sys. Research*, 13:377–388, December 2002.

[MJ11] Wolfgang Maass and Sabine Janzen. Pattern-Based Approach for Designing with Diagrammatic and Propositional Conceptual Models. In *Proceedings of DESRIST 2011 (to appear)*, 2011.

[MMG02] Lynne M. Markus, Ann Majchrzak, and Les Gasser. A Design Theory for Systems That Support Emergent Knowledge Processes. *MIS Quarterly*, 26(3):179–212, 2002.

[MPSG09] Boris Motik, Peter F Patel-Schneider, and Bernardo Cuenca Grau. OWL 2 Web Ontology Language Direct Semantics. *Latest version available at httpwww w3 orgTRowl2semantics*, 2009.

[MS95] Salvatore T. March and Gerald F. Smith. Design and natural science research on information technology. *Decis. Support Syst.*, 15:251–266, December 1995.

[MSK11] Wolfgang Maass, Veda C. Storey, and Tobias Kowatsch. Effects of External Conceptual Models and Verbal Explanations On Shared Understanding in Small Groups, to appear. In *Proc. of 30th International Conf. on Conceptual Modeling (ER 2011), Brussels, Belgium*, 2011.

[OB01] Wanda J. Orlikowski and Stephen R. Barley. Technology and Institutions: What Can Research on Information Technology and Research on Organizations Learn from Each Other? *MIS Quarterly*, 25(2), 2001.

[PHS03] S. Purao, T. Han, and V. Storey. Improving Reuse-based Design: Augmenting Analysis Patterns Reuse with Learning. *Information Systems Research*, 14(3):269–290, 2003.

[WMPW95] Yair Wand, David E. Monarchi, Jeffrey Parsons, and Carson C. Woo. Theoretical foundations for conceptual modelling in information systems development. *Decis. Support Syst.*, 15:285–304, December 1995.

[Yoo10] Youngjin Yoo. Computing in Everyday Life: A Call for Research on Experiential Computing. *MIS Quarterly*, 34(2):213–231, 2010.

erschienen im Tagungsband der INFORMATIK 2011
Lecture Notes in Informatics, Band P192
ISBN 978-3-88579-286-4
weitere Artikel online:
http://informatik2011.de/519.html