

Pattern-Based Approach for Designing with Diagrammatic and Propositional Conceptual Models

Wolfgang Maass^{1,2} and Sabine Janzen¹

¹Research Center for Intelligent Media
Furtwangen University, Germany

²Institute of Technology Management,
University of St. Gallen, Switzerland

Abstract. A conceptual modeling approach for Ubiquitous Information Systems (UIS) is presented as a central part of a UIS design methodology. Three conceptual models are used for step-wise derivation of machine-executable design models for distributed service infrastructures: narratives, pattern-based diagrammatic conceptual models (Pre-Artifacts), and formalized propositional conceptual models.

Keywords. Conceptual modeling, design methodology, Ubiquitous Information Systems, semantic technologies, patterns

1. Introduction

The class of Ubiquitous Information Systems (UIS) has recently gained attention [25, 40] but is governed by ad-hoc methods, e.g. “wild-west” prototyping. 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. In the following, a design methodology for UIS is presented that is centered around three types of conceptual models and corresponding translation procedures.

Design teams for Information Systems are heterogeneous, with members from different fields such as domain experts, various users, decision makers, IT architects, analysts, developers, and Marketing experts. All ideas, expertise, experience, and expectations of these members are brought together for building a homogenous understanding of a future Information System. Explication and communication means are required for building these understandings on various levels. Non-technical members intend to build an Information System that supports their business, social, and communication needs while technical members focus more on engineering aspects of the technical realization of the system. Shared understandings of design teams are described by various conceptual models (CM) that are used during design phases of Information System development [38]. Central to conceptual modeling is the identification of important concepts and relations [28, 38] semantically described by shared vocabularies [26]. Shared vocabularies are either implicitly defined as being part of a mutual understanding in a community or explicitly defined in forms of machine-processable representations [6]. In the latter case, the logic of a CM can be evaluated and matched with other CMs which is important for re-use [33]. CMs abstract 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 [38]. A CM is represented by a conceptual modeling language (CML), such as Entity-Relationship [7] models or the Unified

Modeling Language (UML) [3]. From an IS development process perspective, CMs are used during analysis, design, and realization phases [38].

Ubiquitous Information Systems (UIS) provide means for supporting single actors and groups in real-world situations by services over ubiquitous computing technologies anywhere and anytime [24, 40]. Little research has been done so far on dedicated design methodologies and conceptual modeling for UIS [18, 19]. Klemmer and Linday investigate CMs for designing tangible user interfaces based on physical input devices [19]. Janzen et al. propose a design methodology for UIS that models situations by narratives and semi-structured representations [18]. In this article, we discuss a conceptual modeling approach based on three aligned types of conceptual models that (1) supports holistic and explicit representations of communication and collaboration situations for UIS and (2) uses service infrastructures as a means for supporting social functions. Next, the role of conceptual models is discussed for Information Systems and UIS in particular. This sets the scene for our design methodology for UIS followed by a detailed discussion of a conceptual modeling approach based on diagrammatic and formal propositional CMs. Results and future work close this article.

2. Conceptual Modeling

Conceptual modeling is a key topic for design science theories [38]. Several CML are proposed with a focus on (1) business process modeling (e.g., [35]), (2) general software engineering (e.g., [3]), (3) semantic data models (e.g., [7]), and meta-data models and computational ontologies [2]. Grammars provided by CMLs require ontologies for defining the fundamental entities and structures that shall be focused by CMs [38].

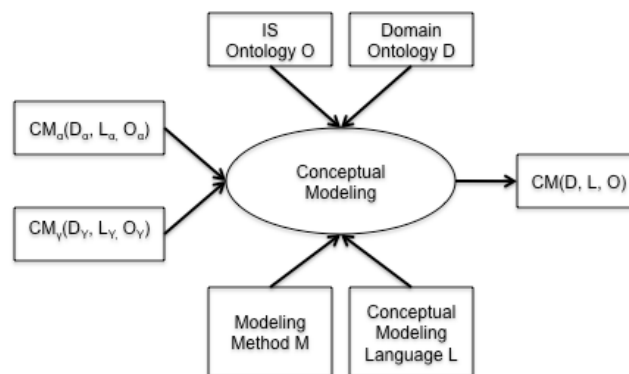


Fig. 1: Generic model of conceptual modeling

In an idealized form, conceptual modeling transforms existing explicit CMs or implicit mental models of members of a design team into integrated CMs, $CM(D, L, O)$, by means of a modeling method M and a conceptual language L based on a domain ontology D and a fundamental information systems ontology O (Fig. 1). Hence, CMs are a type of shared mental model that support mappings from application domains to CMs and from CMs to views of an information system. Useful conceptual modeling approaches “should enable both mappings without loss of information” [38]. The distinction between CMs and design models for information systems gets blurred if CMs can be executed [38] as intended by CMs based on formal ontologies [11].

In different phases, IS design teams use different types of CML. Consistency, syntactic, and semantic interoperability are major obstacles for working with different CMLs. For instance, UML and the Unified Software Development Process (USWDP) provides a set of CMLs for representing different conceptual aspects and a qualitative procedure for iterative and incremental software development [3]. But UML and USWDP do not give clear guidance for building integrated CMs caused by cognitive misdirection, semantic inconsistency, inadequacy, and ambiguity of modeling concepts [36]. Some CML of the UML are qualitative, such as use cases, while others are formal, such as state transition diagrams. For instance, the Rational Unified Process (RUP) provides 159 key resulting artifacts that are created and used during the software development process that are managed by at least four systems [20]. Use case centered development focuses on single observable results of value for a particular actor [3, 17]. For fixing this atomistic view, the concept of a *summary use case* was introduced that textually describes how various instances of a use-case combine to achieving an overarching goal [9]. No structure and guidance for writing summary use cases are given. Thus, use case modeling conveys localized perspectives, which often atomizes overall understandings of a target system [36].

Conceptual modeling frameworks for UIS have to cope with more complex requirements than the more strongly constrained Information Systems for office settings [25]. Contents shall be seamlessly provided by any kind of mobile or embedded device based on loosely coupled service infrastructures while users are moving in physical environments [25]. Hence, situated communication and collaboration of user groups in physical environments are far more complex than well-structured online environments, highly dynamic, and context-dependent on various dimensions [10]. This requires that contents can effortlessly move over loosely coupled and distributed service infrastructures, for instance, supported by semantically annotated contents. Next, a design methodology for UIS is briefly described before the underlying conceptual modeling approach is presented.

3. Design Methodology for Ubiquitous Environments

Designing UIS does not exclusively depend on technical issues but also on aspects concerning, for instance, users, social interactions, and physical surroundings. Environments of UIS cannot be fully specified, i.e. UIS designs should be flexible enough to cope with a range of unpredictable events and entities. This is in contrast with fully specified, artificial digital environments of traditional IS. A basic hypothesis for the design of UIS is that this general requirement for flexibility can be supported by strongly modularized computing environments and dedicated design principles for composing computational modules [40]. Thus, the following limitations of design methods for purely digital IS (e.g., [3, 16, 28]) should be overcome by a design methodology for UIS: (1) consideration of physical objects (e.g., [34]) and (2) contextualized computational modules (e.g., [10]). Contextualized computational modules describe logically coherent interactions not only by its functionality but also with respect to requirements on contents, social organization, interactions, and supporting services. Previous design science research identified seven development principles for the design of information systems, which should be addressed by a design method [27, 32]. Based on these principles, we derived a design methodology for UIS, called *Content-Centered Design of Ambient Environments (CoDesA)* [18]. CoDesA consists of four phases: (1) Identification of Problem &

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 CM (Pre-Artifacts), (4) Evaluation of Pre-Artifacts, (5) Derivation of formal propositional CMs, (6) Formalization of system design, (7) Implementation of formalized system design, (8) Evaluation of solution, and (9) Product development. CoDesA was tested in various UIS development projects. In the following, definition of diagrammatic CM and derivation of formal propositional CM are focused (task 3 and 5). In particular it is presented how diagrammatic conceptual patterns help to solve problems with ambiguities of qualitative CMs (task 3) and how formal web-based CML can be used for deriving executable CMs (task 5).

4. Conceptual Modeling of UIS

4.1 Fundamental Information System Ontology AISM

Conceptual models are strongly influenced by basic conceptualizations of Information Systems [38] (Fig. 2). In general, information systems are compounds of social systems, information, and service systems that use information technology infrastructures for realization of desired situations [22, 23, 31]. With the Abstract Information System Model (AISM), we bring together these three conceptual classes for conceptual models of Information Systems with the additional dimension of physical entities that is required for UIS (cf. Fig. 2):

- 1) Social system: the set of roles available with a set of attributes, such as rights, obligations, and prohibitions, and actions performed by role-taking actors
- 2) Information sphere: all information objects used within the realm of an IS
- 3) Physical object system: the set of physical entities available within all situations in which a UIS can be used
- 4) Service system: the set of all digital and physical services available within all situations in which an IS can be used

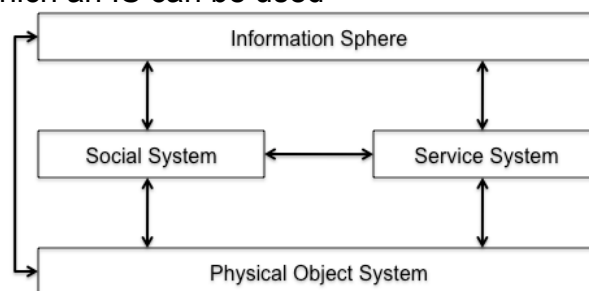


Fig. 2: Abstract Information System Model (AISM)

Information objects that are used in situations by the social system of role-taking actors are defined within an information sphere based on supporting services. Actors use roles, information objects, and services for implementing situations in work and other contexts. An organization consists of structural elements, in particular role systems that describe the capabilities of roles and attributes, and dynamic structures based on interactions that describe procedural aspects of an organization. Interactions are explications of task requirements that are described by directed relations between roles. Communication, as a sub-class of interactions, transfers

information that, in turn, refers to information objects. Information objects are abstract conceptions of any kind of content, such as speech, written language, graphics, or digital contents. Information objects generally require support given by a role-taking actor or a service. In the other case, external services can create information objects and make them available to other services or role-taking actors. Services provide functional capabilities to roles and other services. Services that support roles are called interface services. Interface services provide graphical, tangible, speech or other interfaces by which role-taking actors access services for achieving some situation-specific goals. Mobile, Pervasive, and Ubiquitous computing [24, 30, 37, 39] are approaches by which Information Systems (a) extend from spatially restricted access to information spheres to temporally and spatially unrestricted access forms and (b) invisible embedding of information technologies in physical environments. The physical object system encompasses all physical objects that are relevant for the design model of an Information System. In the following, AISM takes the role of the IS ontology O (Fig. 1).

4.2 Pre-Artifacts

Situations are instance-based descriptions of interactions between entities in an environment that use concepts and relations defined by shared vocabularies. CMs described by a CML capture situations and frame design discussions [15]. Our methodological approach is based on three CM types: narrative CM, diagrammatic CM (Pre-Artifacts), and propositional CM. In task 2, narratives describe identified situations. The mode of narrative thinking is highly context-sensitive, anchored in situations, articulated in temporal sequences, around individual and group intentions and actions [41]. Narratives are effective means for building and understanding situations of future Information Systems because of their capacity to provide discourse information and sequential orderings of interactions between actors [14, 21]. Next narratives are translated into diagrammatic CM, called Pre-Artifacts (task 3). Based on the AISM, Pre-Artifacts conceive usage situations by emphasizing requirements on social structure, information objects, physical objects, and services in a coherent structured manner. All core entities are identified in narratives that fit to these conceptual categories [18].

Similarly, relations are extracted that connect these entities. Analogue to use cases, Pre-Artifact are described on instance level but are used as prototypes for class descriptions [38]. The concept of a Pre-Artifact resembles the basic concept of use cases because it also describes logically consistent parts of a situation. But Pre-Artifacts are structured by an underlying IS model (AISM) with a set of defined concepts (information object, role, services, and interactions) while use cases are neutral with respect to IS models. Another differentiating factor is that the Pre-Artifact model is geared towards role-based designs that explicitly demand relationships between information objects and roles.

For heterogeneous information spheres this is important for qualification of information with respect to, for instance, reputation, responsibility, and copyrights and also for explanation-based systems. For these kind of typical IS requirements, use cases are not specific enough and require external guidelines that, in turn, increases the complexity for their application in design situations. In the following, we focus on modeling with diagrammatic CM patterns, called Pre-Artifact Patterns, and translation of resulting diagrammatic CM into formal propositional CM.

4.3 Pre-Artifact Patterns

Analysis of Pre-Artifacts in several UIS development projects showed re-occurring structures similar to the notion of design patterns as used in architecture [1] and Software Engineering [12]. They represent means for reusable CMs for Information Systems. We identified seven Pre-Artifact patterns (cf. Fig. 3) that are elaborated in the following.

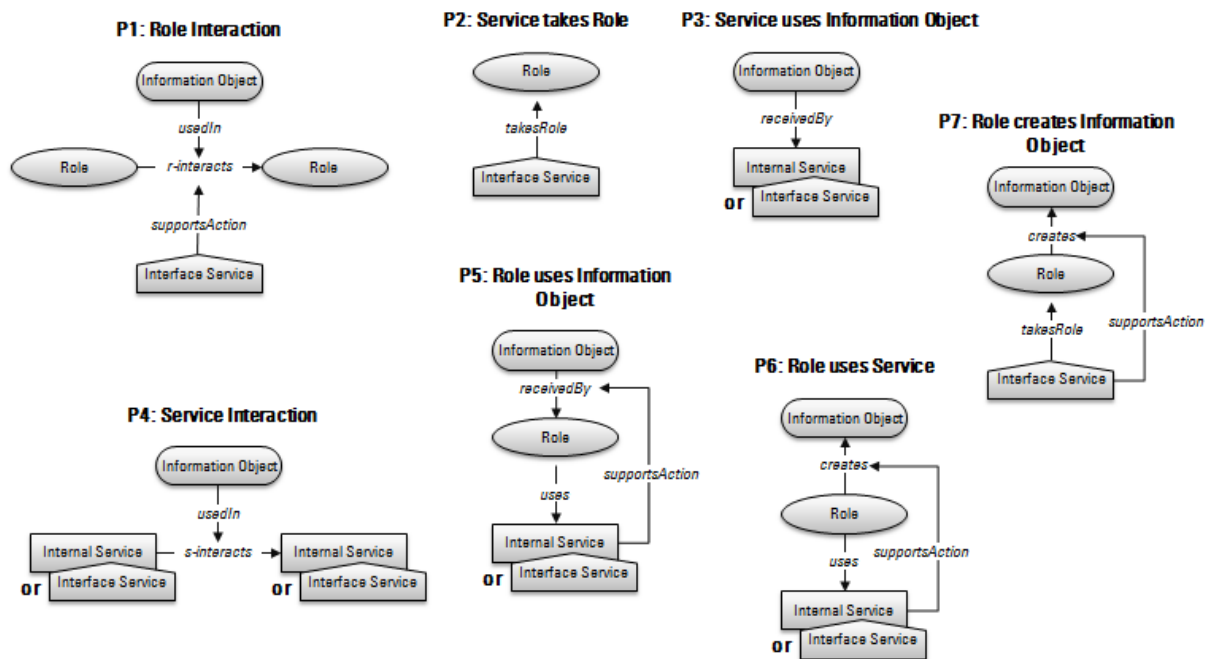


Fig. 3: Pre-Artifact patterns

Role Interaction Pattern (P1): 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. The interaction between roles is described by a generic property called *r-interacts*. The interface service is only used as a communication channel.

Service takes Role Pattern (P2): This pattern represents a situation in which a role is taken by an interface service. For instance, Wikipedia provides information and takes a role with connotated social attributes, such as reputation and credibility.

Service uses Information Objects Pattern (P3): expresses that an internal or interface service receives information objects without human interventions. This is a simplification of the Service Interaction pattern. It is used when a providing service is not important for a CM. For instance, stock information used by a local service and received from a cloud infrastructure.

Service Interaction Pattern (P4): 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 on service level.

Role uses Information Object Pattern (P5): In situations with direct manipulation of information objects, this pattern allows to express that a role receives an information object by using an internal or interface service. This means a role-taking actor can actively receive an information object supported by a service. For instance, a CEO who uses a business intelligence service for accessing corporate sales information.

Role uses Service Pattern (P6): This pattern describes a situation with a role-taking actor creating an information object. Therefore, the actor uses a service that supports the creation of an information object, e.g., a nurse who creates a status report for a patient by a healthcare reporting service.

Role creates Information Object Pattern (P7): 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.

4.4 Method for Conceptual Modeling with Pre-Artifact Patterns

Conceptual modeling of Pre-Artifacts is improved by Pre-Artifacts patterns because they provide conceptual structures as basic building blocks for IS designs. The construction of Pre-Artifacts is guided by a method with five steps based on instantiation and integration of Pre-Artifact patterns. Each step pursues a sub goal for constructing the Pre-Artifact step by step and proclaims specific Pre-Artifact patterns that help to achieve the objectives (cf. Tab. 2).

Step 1: Definition of Information Objects in Infosphere. All information objects that occur in a narrative are defined as Information Objects (IO) in the Infosphere.

Tab. 1: Appliance of Pre-Artifact patterns within steps of defining Pre-Artifacts

Patterns P / Steps	Step 1	Step 2	Step 3	Step 4	Step 5
<i>P1: Role Interaction</i>	-	x	-	-	-
<i>P2: Service takes Role</i>	-	-	x	-	-
<i>P3: Service uses Information Object</i>	-	-	-	x	-
<i>P4: Service Interaction</i>	-	-	-	x	-
<i>P5: Role uses Information Object</i>	-	-	-	-	x
<i>P6: Role uses Service</i>	-	-	-	-	x
<i>P7: Role creates Information Object</i>	-	-	x	-	-

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 newly generated information objects have to be defined. These

interactions take place between *Roles* in the *Social System* exclusively. Thus step 2 connects infosphere and social system. Interactions between user and system are always supported by a service of the *Service System* that is defined in Step 3. The requirements of this step are fulfilled by the application of the *Role Interaction pattern* exclusively.

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. Either a service is linked to a role that was already defined in step 2 or it adds a new role. Thus step 3 connects infosphere and social system with the service system. To manage this step, the *Role Creates Information Object pattern* is applied to define the creation of the information object by a role taken by a service.

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 IO 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 *Service Interaction pattern*.

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 *Role uses Service* or *Role uses Information Object* pattern. 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.

5. Example

Next, an example will be given for deriving a diagrammatic CM from a narrative CM based on the Pre-Artifact CML, Pre-Artifact patterns, and CM method. Finally, three approaches of translating Pre-Artifacts into propositional CMs are discussed and exemplified.

5.1 Defining Pre-Artifacts

Step 1: Definition of *Information Objects* in *Infosphere*. Fig. 4 shows this modeling step by means of an exemplary Pre-Artifact that shall represent the narrative: “*It's Thursday morning. I get site-specific weather information when I am brushing my teeth in the bathroom.*” The figure 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*. In the exemplary Pre-Artifact (cf. Fig. 4) 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 IO *site-specific weather information*.

Step 3: Definition of Roles taken by Services. In our example, the Personalized Weather Service takes the role of the Personalized Weather Assistant that creates the IO site-specific weather information. The interface service supports this action indirectly (cf. Fig. 4). To express the plain role-taking by a service without a creating function, the *Service takes Role* pattern can be applied.

Tab. 2: Appliance of Pre-Artifact patterns within steps of defining Pre-Artifacts

Patterns P / Steps	Step 1	Step 2	Step 3	Step 4	Step 5
<i>P1: Role Interaction</i>	-	x	-	-	-
<i>P2: Service takes Role</i>	-	-	x	-	-
<i>P3: Service uses Information Object</i>	-	-	-	x	-
<i>P4: Service Interaction</i>	-	-	-	x	-
<i>P5: Role uses Information Object</i>	-	-	-	-	x
<i>P6: Role uses Service</i>	-	-	-	-	x
<i>P7: Role creates Information Object</i>	-	-	x	-	-

Step 4: Definition of supporting Internal Services. The exemplary Pre-Artifact (cf. Fig. 4) 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. Step 5 is not required in this example because the user does not initiate interactions with the UIS (cf. Fig. 4).

Next, Pre-Artifacts are translated into propositional CMs (task 5). Currently this translation is a manual task but we work on an automatic translation mechanism so that designers are not required to deal with formal logics. The objective of this translation is the creation of specifications for later system designs [38] as well as machine-processable CMs that can be verified [2]. Considering the method of formalization, there are several opportunities, for instance Unified Modeling Language (UML), entity-relationship model (ER) or a formalization, for instance, based on ontologies by means of RDF (<http://www.w3.org/RDF>) or OWL (<http://www.w3.org/TR/owl-ref>). The use of computational ontologies for conceptual modeling by means of a pattern-based approach has already been investigated [8, 13]. Bera et al. (2010) identified some unique features of OWL that are not available in ER model and in UML. Amongst others, OWL is implementable, which 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. Concerning the verification aspect, OWL allows inferences and automated reasoning support. 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.

There are at least three approaches of translating Pre-Artifacts into propositional CMs. Each option was tested by modeling three exemplary Pre-Artifact patterns (Role Interaction, Role creates Information Object, Service Interaction) that were used for generating the propositional CM in Section 5.1.

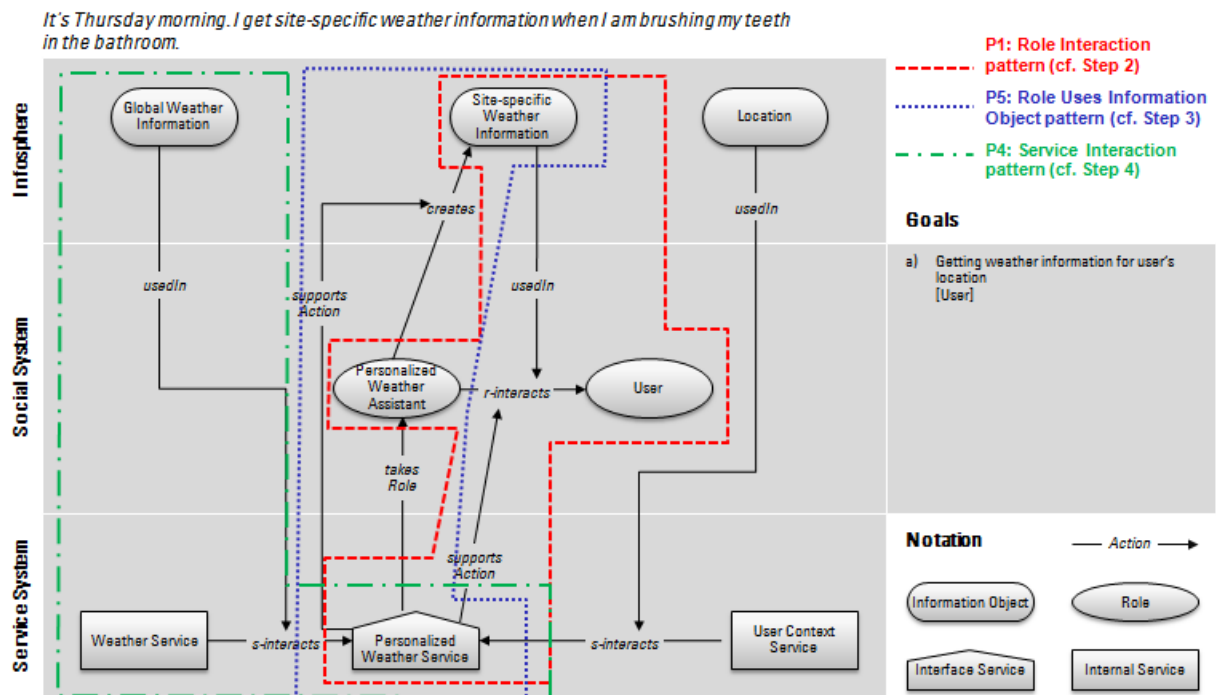


Fig. 4: Translation of a narrative in a Pre-Artifact based on three Pre-Artifact patterns

5.2 Translation of Pre-Artifacts into Propositional Conceptual Models

Approach 1. This option was realized by representing each Pre-Artifact pattern by a unique formal propositional model. The approach leads to redundant concepts when integrating propositional models into a complex propositional CM. A modeling person has to specify equivalences to resolve these redundancies; e.g., in our example (Fig. 4) the concept type “Role” occurs three times because three of the imported patterns contain this concept type. This procedure demonstrates the aforementioned lack of modeling guidelines and constraints [2].

Approach 2. Following the approach by Bera et al. (2010) of using a “philosophical ontology” to derive guidelines on how OWL constructs can be applied in the modeling of propositional conceptual models, a Pre-Artifact Model is 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 sub-classes *Interface* and *Internal Service*. Furthermore, a super-class *Action* is defined that contains further sub-classes that specify diverse types of pattern actions: *Creation*, *Receiving* and *Interaction* with 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 [29]. The advantage of the former opportunity lies in adding actions as specific concepts to 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 the Pre-Artifact Model. Afterwards, the pattern ontologies specify relevant generic object properties with additional concepts. When integrating the pattern ontologies in the propositional CM, ambiguous assignments of object properties to specific patterns occur. Because of lack of clear results and statements, the second approach is not a proper solution for handling the lack of modeling guidelines.

Approach 3. In this approach, the notion of the Pre-Artifact Model as well as the integration of this “vocabulary” into pattern ontologies is adopted. 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. properties can exist independent of classes [2]. 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.

5.3 Example

Considering CoDesA Task 5 – the formalization of propositional Pre-Artifacts -, the aforementioned third approach was applied. To model the propositional Pre-Artifact based on the diagrammatic Pre-Artifact, the expressiveness of the semantic pool of Pre-Artifact patterns is used. This approach is based on the Pre-Artifact Model, the pool of Pre-Artifact Patterns, and propositional CMs. For derivation of propositional CM tools for modeling formal web-based representations are required, e.g., Protégé (<http://protege.stanford.edu/>). After generating an empty OWL file in Protégé, required Pre-Artifact patterns are imported by their URL. According to the procedure of defining Pre-Artifacts (cf. Section 5.1), the formalized model of the pattern *RoleInteraction* is imported. Then, the relevant concepts of the pattern are instantiated, e.g., by creating an instance of the concept “Role” named “User”. To represent the interaction between *User* and *PersonalizedWeatherAssistant*, an instance of the concept *R-Interaction* is created. For linking both roles with the instance of R-Interaction, the formalized pattern offers the specified object properties *initiatesR_Interaction* and *finalizesR_Interaction* that inherit from the super-properties *initiatesInteraction* and *finalizesInteraction*. Within the proceeding formalization, the formalized patterns *Role uses IO* and *Service Interaction* are imported. Note, each pattern automatically imports the Pre-Artifact Model. The result of the formalization is an OWL description that represents the exemplary narrative in a formal and computational way.

6. Discussion

Explication and integration of individual understandings of different members of a design team are central tasks of conceptual modeling for Information Systems.

CMLs with an origin in Computer Science are technical languages that are difficult to use by non-technical design team members [36]. Therefore additional CMLs are required that support capturing different aspects of CMs. We presented a design methodology with three types of CMs for translating individual mental models, into narratives, diagrammatic CMs (Pre-Artifacts), and finally formalized propositional CMs. Thus shared understandings are incrementally supported from qualitative, textual descriptions of complete situations into, semi-structured representations with consolidated conceptual structures, and machine-processible, propositional representations based on formal ontologies. We have already evaluated by initial empirical studies that narratives and Pre-Artifacts are useful tools for modeling complex Information Systems. The Pre-Artifact modeling task is supported by a pattern-based approach that provides generic conceptual modules. Translation of Pre-Artifacts into formalized propositional CM support IT experts in rapid prototyping of Information Systems based on distributed service infrastructures, as common for UIS.

We have applied this modeling approach to a real world situation. Resulting formal propositional CMs are directly used as design models that can be executed on semantic technology infrastructures [5]. Thus, the three-step conceptual modeling approach of CoDesA supports rapid prototyping for complex UIS by

- (1) keeping the holistic structure of situations that supports an integrated understanding of interactions within complex socio-technical systems
- (2) providing narratives as qualitative CMs that support integrated conceptual views of design members from various domains,
- (3) structured pattern-based translation of narratives into Pre-Artifacts, and
- (4) algorithmic translation of Pre-Artifacts into formal propositional CM that can be executed on semantic technology infrastructures.

This research is a key step towards a structured conceptual modeling process in particular for complex UIS leveraging distributed service infrastructures in the sense of a utility computing model [4]. Whether CoDesA and Pre-Artifacts are sufficient means for modeling CM for UIS has to be proven in further projects. Furthermore it is an open issue whether OWL is sufficient as a formal language for automatic translation of diagrammatic CM into propositional CM of complex UIS.

7. Conclusion and Future Work

A three-step conceptual modeling approach was presented as an integral part of a design methodology for UIS. It was argued that a pattern-based modeling approach provide structure and guide conceptual modelers without loss of too much freedom. We have shown the importance of IS ontologies and how they fundamentally affect conceptual models. The model of conceptual models provides a concise roadmap for future research. For instance, mapping of narratives derived by different modelers as input for a Pre-Artifact modeling task. Currently we work on empirical evaluations of the effectiveness of our conceptual modeling approach.

8. Acknowledgements

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

9. References

- [1] C. Alexander, *A Pattern Language: Towns, Buildings, Construction*, (Oxford University Press, 1977).
- [2] P. Bera, A. Krasnoperova, Y. Wand, Using Ontology Languages for Conceptual Modeling, *Journal of Database Management*, 21(1) (2010) 1-28.
- [3] G. Booch, J. Rumbaugh, e. al, *The Unified Modeling Language User Guide*, (Addison-Wesley, Redwood City, CA, 1999).
- [4] N. Carr, The End of Corporate Computing, *MIT Sloan Management Review*, 46(3) (2005) 67-73.
- [5] J.J. Carroll, I. Dickinson, C. Dollin, D. Reynolds, A. Seaborne, K. Wilkinson, Jena: implementing the semantic web recommendations, in: *Proceedings of the 13th international World Wide Web conference on Alternate track papers and posters*, (ACM, New York, NY, USA, 2004), pp. 74-83.
- [6] B. Chandrasekaran, Design Problem Solving: A Task Analysis, *AI Magazine*, 11(4) (1990) 59-71.
- [7] P. Chen, The Entity-Relationship Model--Toward a Unified View of Data, *ACM Transactions on Database Systems*, 1(1) (1976) 9-36.
- [8] P. Clark, J. Thomson, B. Porter, Knowledge Patterns, in: *In Proc. of KR-2000*, (Morgan Kaufmann, 2000), pp. 591--600.
- [9] A. Cockburn, *Writing effective use cases*, (Addison-Wesley, Upper Saddle River, NJ, 2001).
- [10] A. Dey, G. Abowd, Towards a better understanding of context and context-awareness, in, (College of Computing, Georgia Institute of Technology, 1999).
- [11] J. Evermann, A UML and OWL description of Bunge's upper-level ontology model, *Software and Systems Modeling*, 8(2) (2009) 235-249.
- [12] E. Gamma, R. Helm, R. Johnson, J. Vlissides, *Design Patterns. Elements of Reusable Object-Oriented Software*, (Addison-Wesley, 1994).
- [13] A. Gangemi, Ontology Design Patterns for Semantic Web Content, in: Y. Gil (Ed.) *Proc. of the International Semantic Web Conference (ISWC)*, (2005), pp. 262 - 276.
- [14] L. Hinchman, S. Hinchman, *Memory, identity, community: the idea of narrative in the human sciences*, in, (State University of New York Press Albany, 1997).
- [15] L. Hult, M. Irestig, J. Lundberg, Design Perspectives, *Human-Computer Interaction*, 21(2006) 5-48.
- [16] E.B. Jackson, R.L. Jackson, *Industrial information systems*, (Academic Press, 1978).
- [17] I. Jacobson, M. Griss, P. Jonsson, *Software Reuse: Architecture, Process and Organisation for Business Success*, (Addison-Wesley and ACM Press, Reading MA, 1997).
- [18] S. Janzen, T. Kowatsch, W. Maass, A Methodology for Content-Centered Design of Ambient Environments,, in: R. Winter, J. Zhao, S. Aier Eds. *DESRIST 2010*, (St. Gallen, Switzerland, 2010), pp. 210-225.
- [19] S. Klemmer, J. Landay, Toolkit Support for Integrating Physical and Digital Interactions, *Human-Computer Interaction*, 24(2009) 315-366.
- [20] P. Kruchten, *The Rational Unified Process-An Introduction*, Addison-Wesley, 3rd ed., (Addison-Wesley, 2003).
- [21] W. Kuechler, V. Vaishnavi, On theory development in design science research: anatomy of a research project, *Eur J Inform Syst*, 17(2008) 489-504.
- [22] R. Lamb, R. Kling, Reconceptualizing Users as Social Actors in Information Systems Research, *MIS Quarterly*, 27(2) (2003) 197-235.

- [23] U. Lechner, B.F. Schmid, Communities and Media -- Towards a Reconstruction of Communities on Media, in: 34th Annual Hawaii International Conference on Systems Sciences (HICSS-34), (2001).
- [24] K. Lyytinen, Y. Yoo, Issues and Challenges in Ubiquitous Computing, *Communication of the ACM*, 45(2002) 62-65.
- [25] K. Lyytinen, Y. Yoo, Research Commentary: The Next Wave of Nomadic Computing, *Inform Syst Res*, 13(4) (2002) 377-388.
- [26] S. March, G. Smith, Design and natural science research on information technology, *Decision Support Systems*, 15(1995) 251-266.
- [27] M.L. Markus, M. Keil, If We Build It, They Will Come: Designing Information Systems That People Want to Use, *Sloan Management Review*, 35(4) (1994).
- [28] M.L. Markus, A. Majchrzak, L. Gasser, A design theory for systems that support emergent knowledge processes, *Mis Quart*, 26(3) (2002) 179-203.
- [29] B. Motik, P. Patel-Schneider, B. Cuenca Grau, OWL 2 Web Ontology Language: Direct Semantics, in, (W3C, 2010).
- [30] E.W.T. Ngai, T.C.E. Cheng, S. Au, K.-h. Lai, Mobile commerce integrated with RFID technology in a container depot, *Decis Support Syst*, 43(1) (2007) 62-76.
- [31] W.J. Orlikowski, S.R. Barley, Technology and Institutions: What can Research on Information Technology and Research on Organizations Learn from each others?, *Mis Quart*, 25(2) (2001) 145-165.
- [32] J. Pries-Heje, R.L. Baskerville, The Design Theory Nexus, *Mis Quart*, 32(4) (2008) 731-755.
- [33] S. Purao, V.C. Storey, T.D. Han, Improving Reuse-Based System Design with Learning, *Information Systems Research*, 14(3) (2003) 269-290.
- [34] P. Ross, D. Keyson, The case of sculpting atmospheres: towards design principles for expressive tangible interaction in control of ambient systems, *Personal Ubiquitous Computing* 11(2) (2007) 69-79.
- [35] A.-W. Scheer, ARIS. Modellierungsmethoden, Metamodelle, Anwendungen, 3 ed., (Springer, Berlin, 1998).
- [36] A. Simons, I. Graham, 30 Things that Go Wrong in Object Modelling with UML 1.3, in: H. Kilov, B. Rumpe, I. Simmonds Eds. *Behavioral Specifications of Businesses and Systems*, (Kluwer Academic Publishers, 1999), pp. 237-257.
- [37] U. Varshney, R.J. Vetter, R. Kalakota, Mobile commerce: a new frontier, *Computer*, 33(10) (2000) 32-38.
- [38] Y. Wand, D. Monarchi, J. Parsons, C. Woo, Theoretical foundations for conceptual modelling in information systems development, *Decision Support Systems*, 15(1995) 285-304.
- [39] M. Weiser, The computer for the 21st century, *Pervasive Computing*, first published: *Scientific American*, 1991, 1(1) (2002) 18-25.
- [40] Y. Yoo, Computing in Everyday Life: A Call for Research on Experiential Computing, *Mis Quart*, 34(2) (2010) 213-231.
- [41] H. Zukier, The Paradigmatic and Narrative Modes in Goal-Guided Inference, in: R. Sorrentino, E. Higgins Eds. *Handbook of Motivation and Cognition: Foundations of Social Behavior*, (Guilford Press, New York, 1986), pp. 465-502.