Tooling of and training for
collaborative, distributed product
innovation
idSpace -2008-216199

Project Deliverable Report

Deliverable D3.1 – Description of Context Awareness in idSpace

Work Package	WP3		
Task	T3.1, T3.2, T3.3		
Date of delivery	**Contractual**: 01-10-2008	**Actual**: 28-11-2008	
Code name		**Version:** 1.0	Draft ☐ Final ☒
Type of deliverable	Report		
Security (distribution level)	PU		
Contributors			
Authors (Partner)	UCY, MORPH, AAU, UHI, EMS		
Contact Person	Aimilia Tzanavari, George A. Sielis (UCY)		
WP/Task responsible	UCY		
EC Project Officer	Mr. Christian Wilk		
Abstract (for dissemination)	This deliverable describes context awareness as it is envisaged to appear in the idSpace platform. It first offers an overview of the state-of-the-art in the area. Next context awareness in idSpace is summarized and presented in an ontology format.		
Keywords List	Context awareness, ontology, recommendations		

idSpace Project Coordination at: Open University of the Netherlands
Valkenburgerweg 177, 6419 AT Heerlen, The Netherlands
Tel: +31 45 5762624– Fax: +31 45 5762800

Table of Contents

List of Tables and Figures

1. Introduction

This document constitutes the first deliverable produced within "Work Package 3 – Context Awareness for the idSpace platform" of the idSpace project. The work behind this deliverable was carried out between months one (1) and six (6), by partners UCY (leader), MORPH, AAU, HIL and EMS.

Describing what context awareness will mean for idSpace, is the first step in the process of adding context awareness to the platform. A state-of-the-art analysis in the area is first presented. Following this, and after identifying potential problems and shortcomings, a specific proposal is formed and presented.

The literature review begins from the definition of context (Chapter 2) and moves on to the areas that context awareness is applied to (Chapter 3), the context elements used (Chapter 4), in what form(s) these elements are represented (Chapter 5) and finally what are the various models, methods (Chapter 6), architectures and frameworks (Chapter 7) encountered in the design of such systems.

Subsequently, as the idSpace platform is considered a creativity support tool, we visit the various existing tools and comment on their characteristics (Chapter 8). We particularly report on their 'context awareness' aspect, and identify shortcomings in several points. Following this, we present the description of our proposal for context awareness for idSpace (Chapter 9), which attempts to overcome most of these problems.This description starts from the distinct context elements we consider suitable for the idSpace context, first presenting them independently and finally putting them all together in an ontology format, also presenting the way they are related.

Finally the deliverable concludes with remarks on the work done thus far in this Work Package and outlines the work that will follow this deliverable, which includes prototyping the proposed context awareness.

2. Definition of Context

Context Awareness has been an active area of research in the last years. The importance of context awareness focuses on the automated services that can be offered from computing systems to users. In the literature the word ¨context¨ is defined as the set of facts or circumstances surrounding a situation or event. Being aware of the context means ¨knowing¨ the circumstances or the facts that surround an event, in order to activate it.

Context awareness in computer science could be defined as the recognition of the user's environment parameters, which will subsequently become the impulse for the activation of the corresponding functioning. Schilit, Adams, & Want (1994) attempted to define context by specifying three categories: *computing context, user context and physical context*. Chen et al. (2000) extended this definition by adding the *time context* element. The definitions given by Chen et al. (2000) and Schilit, Adams, & Want (1994) have a common limitation: they do not specifically state the boundaries for considering information as context or not (Hartmann et al. (2008). A first solution to this problem was given by Dey et al. (2001) by limiting the context to all information that is relevant to the interaction between user and application: "*Context is any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application including the user and applications themselves.*" The problem with this definition was that it was not specific enough as to what can be considered as context, thus every application could be in a sense described as context-aware. Therefore, the definition of context required an upper and a lower bound to only include information that is strictly relevant and necessary to perform context-awareness. This definition is given by Hartmann et al. (2008): *Context characterizes the actual situation in which the application is used. The situation is determined by information which distinguishes the actual usage from others, in particular characteristics of the user (user's location task at hand, etc.) and interfering physical or virtual objects (noise level, nearby resources etc.). Thereby, we only refer to information as context that can actually be processed by an application (relevant information), but that is not mandatory for its normal functionality (auxiliary information).*

3. Context Awareness Areas

Context Aware applications cover a wide spectrum of different research areas of computers science and information technology. Here we attempt to present some of the areas for which context awareness applications have been developed.

Dustdar (2004) presented a combination of ideas based on Bardram (2005) and Pascoe (1998) which is a list of features that the context aware applications may support. We conclude in three categories of context: 1) *presentation* of information and services to a user; 2) automatic *execution* of a service; and 3) *tagging* of context to information for later retrieval. Bontas (2008) introduced a list of examples that show the variety of context usage forms in computer science like Domain Classification, Natural Language Processing, Information Integration and Mobile Computing. Taking into consideration the examples given by Bontas (2008) for each form of context usage in Computer Science and the categorization of context, it becomes apparent that context-awareness is not solely a characteristic of computer applications of a particular domain; rather it can be observed/applied in several domains in different ways.

This can be shown from the multiple applications based on context-aware computing and ubiquitous computing. Even though the majority of context aware applications fall under the computer science research umbrella, there are also other applications that use context awareness for specific purposes, as can be seen in the sections that follow.

3.1 Context Awareness in Healthcare

The area of health care research appears as one of the interests for a potential testing area of tools and frameworks which support context awareness. Examples of such tools are *Vocera communication System* (Stanford, 2003) and *MobileWARD—Mobile Electronic Patient Record* Kjeldskovet (2004). Vocera (Stanford, 2003) is a communication badge system for mobile users that apart from a push to call button and a small text screen has voice dialling capabilities based on voice recognition. It allows hands free conversations, voice messages and it is biometrically secured with speaker verification. It delivers the data directly to the users without the need to use a distance device, like a phone or a pc. The MobileWARD is a hospital prototype which supports the morning tasks in a hospital ward and it is able to display patients´ profiles and information. The information and the functionality presented by the application depend on the nurse's location and the time of the day. Other context awareness examples in the healthcare sector are *Context-aware mobile communication—CICESE* (Munoz et al., 2003), *Intelligent hospital software* Mitchell et al., 2000). CICESE is a context aware mobile communication application which is used by hospital workers to carry out their tasks. Its contextual elements include: location, delivery timing, role reliance, artefact location and state.

3.2 Context Awareness in Ubiquitous Computing

As ubiquitous computing is a post-desktop model of human computer interaction, this area of computing research deals with the mobile computing, wearable computing, with the aim of such research applications to locate and serve the user. The basic mechanism of context awareness computing in this area is summarized in Abowd (1997) into four steps:

1. Collect information on the user's physical, informational or emotional state.

2. Analyze the information, either by treating it as an independent variable or by combining it with other information collected in the past or present.

3. Perform some action based on the analysis.

4. Repeat from Step 1, with some adaptation based on previous iterations.

An example application presented by Abowd (1997) is the CyberGuide application which uses the capabilities of the Personal Digital Assistant (PDA) to locate in a map the position of a user and give him directions and information for sights, restaurants or hotels that are close to him. The CyberGuide is an event-driven model where components act as event sources.

3.3 Context Awareness in e-commerce

Tarasewich (2003) presents the principles of mobile commerce and states that designing successful m-commerce applications and their interfaces, is dealing with context. As a step beyond the traditional wired web-commerce today the commerce research is taking advantage of the mobile applications, like PDAs, Bluetooth technologies, etc. With mobile computing, people might be anywhere, any time. The e-commerce area could solve the problem of mobile users by just determining the location of users. But this is something more complex. Tarasewich (2003) mentions that mobile application use can vary continuously because of changing circumstances and differing user needs. That is the reason behind the need of creating context models in mobile commerce research area. Chen et al. (2007) give a summary of context awareness application. They describe the application proposed by Abhaya (1994) which determines the location of a customer within a store, and gives to the customer information about the items, like, how to locate them, points items on sale, makes comparative price analysis.

3.4 Context Awareness in e-learning

In learning, the adoption of context awareness is not a new idea. It has been demonstrated in learning systems for quite some time. Classical methods, such as those encountered in early intelligent tutoring systems (Wenger, 1987) and student modeling (Brusilovsky & Schwarz, 1993) can all be regarded as context-aware approaches used as adaptation methods (Brusilovsky & Schwarz, 1993). They support the idea that the problem in adaptive interfaces is tracked on the complexity of the interfaces used in such applications. Adaptive interfaces can be the starting point for the determination of the significance of the existence of context awareness in e-learning applications. The usual e-learning applications are not supporting adaptation methods, thus users with different abilities, web experience, knowledge and background get the same pages in the same context (Brusilovsky & Schwarz, 1993). Recent Adaptive Educational Systems, most of them web-based (Brusilovsky, 2003), promise to offer adaptation with respect to the presentation of the learning material, the navigation support, the curriculum sequencing and support in problem solving. One of the methods used for establishing adaptivity and adaptability is context awareness. However, the proposed exploitation of contextual information from a broad spectrum as a means of enabling the best possible support in collaborative learning that aims at creative knowledge building is a novel and promising area. See also Van Rosmalen, 2005; Janssen et al., 2006; Janssen et al., 2007

4. Context Elements

From the definition of context and the categorization of context given in previous sections we can conclude that a context awareness application has two ways of using context. The first is with the automatic adaptation of context according to the discovered context and the second is the dynamic 'on the fly' updating of context in the user´ s profile for future use (Chen et al., 2007). The context awareness applications are using the human computer interaction and the interaction of the environmental conditions that the user is surrounded, to collect information and create or retrieve the corresponding context. The necessary actions, the events, or the signals that a context awareness application needs to start functioning, in order to become aware of the context are the context elements.

Context elements are dependent on the kind of context they are aware of (Karen & Hummel, 2005) separate the context of learning in three kinds: *1. Digital context, 2. Device context 3. Learner Information context.* The interaction between a context aware application and the physical environment must be expressed by a bridging model that describes the entities of the real world and their interactions (Harter et al., 2002). Simple forms of context elements are the usual input/output methods like the keyboard and the mouse. More complex context aware applications use context elements which are hardware or software implementations depending on the kind of context they present. An example of two different approaches of context awareness applications that use hardware and software context elements are presented in Harter et al., (2002) and Schmidt & Winterhalter (2004) respectively. Harter et al., (2002) present a platform that uses sensors and telemetry software to collect environmental data. The platform described contains five components, a location system, a data model for the real world entities description, a persistent distributed system for the data model representation, resource monitors for the communication between network equipment and information status and a spatial monitoring service which enables event based location-aware applications. The context elements for the contained components are implemented with sensors which are based in radio-based techniques (e.g. GPS), electromagnetic methods (e.g. interference from monitors and metal structures, imaging detectors). The elements implementation is described with three classes of resource monitor *1. Machine activity (keyboard activity), 2. Machine resources (CPU usage, memory usage) 3. Network point-to-point bandwidth and latency.* Schmidt & Winterhalter (2004) describe an e-learning ontology based platform which accomplishes the context acquisition with *workflow systems, human resources systems, web browsers* and *office applications.*

In yet another approach, Dey, Abowd & Salber (2001) proposed four different categories, focusing on the user perspective: identity, location, status, and time. Finally, Wang (2004) attempts to refine the definition context for the area of education, by specifying six dimensions: identity, spatio-temporal, facility, activity, learner and community. These six dimensions form a *context space*. *The identity dimension is commonly adopted in computer-supported learning, and the spatio-temporal and facility dimensions are specific to Context-aware Mobile Learning (CAML). The activity dimension, although having appeared in various types of computer-supported learning, contains more characteristics in mobile learning.*

5. Representation of Context

According to the definition of context in section 1, context is any information which determines and characterizes the situation of an entity. An entity is a user-person, place, an object or virtual object like noise level or a resource´ s location. Context entities can be structured into three domains: the user domain, the computer domain and the environment domain (Jun-Zhao & Sauvola, 2003). In a context-aware application the three domains are interacting between each other and the context data that can be collected as an ensemble of the context data for an entity. As a sequence of the separation of the context entity is the need of setting rules for the representation of the context. The challenge is that we have to find a way to separate the context and organize it in order to achieve the desired representation of all the kinds of context that interact with an entity. By the division of entities in domains and since the context data can be retrieved from various entities (Jun-Zhao & Sauvola, 2003) the context information can be described in internal and external features that represent the context. The internal features are used to describe characteristics that exist inside the entity or its domain and the external features are those which describe the context information that can be retrieved from the interaction of an entity with other entities. The features of an entity may have a significant type e.g. string, real, vector, etc. The value of an entity's feature can be also representing an other entity. Through the categorization and the classification of the context information, then markup languages with standard notations are used to represent the context to the clients. An example of a context aware application using the Standard Generic Markup Language (SGML), *e-sticky* is presented by Brown, Bovey & Chen (1997). They mention that the representation of context information must become as easy as the development of a web page in HTML. It also emphasizes the importance of the syntax used within these languages, and the importance of DTD (Document Type Definition), which defines the tags that can be used and how they 'fit' together. The DTD is an easy method that allows the author to define his own tags, or augment existing ones (Brown, Bovey & Chen, 1997) and the DTD can be used to represent the contextual fields.

6. Models and Methods

Key-Values Models

Key-value coding is a mechanism for accessing an object's properties indirectly, using strings to identify properties, rather than through invocation of an accessor method or accessing them directly through instance variables. Key-Values models are the simplest data structure for context modelling and they are frequently used in various service frameworks, where the key-value pairs are used to describe the capabilities of a service. Schilit, Adams & Want (1994) used the key values for modelling context. Key-value coding supports properties that are objects and non-object parameters and return types are detected and automatically wrapped, and unwrapped, as required. Key-Values Modelling is simple but not very efficient for more sophisticated data structuring purposes, because it needs exact matching to support retrieval context algorithms and it does not support inheritance.

Markup Scheme Models

All mark-up based models use a hierarchical data structure consisting of mark-up tags with attributes and content. The content of the mark-up tags are defined in other mark-up tags. The mark-up schemes are usually used to collect information for profiles. The context information profile building is usually using the SGML which is the Standard Mark-up Language the super class of all the mark-up languages like XML. There are multiple examples of such profiles given by Strang & Linnhoff-Popien (2004) like the Composite Capabilities / Preferences Profile (CC/PP) and User Agent Profile (UAProf).

Graphical Models

UML *Unified Modeling Language* is a modeling context language which is structuring the context modeling based on UML diagrams. A Context graphical model is also introduced by Henricksen, Indulska & Rakotonirainy (2003), which is an ORM extension (Object Role Extension) with differences from the classic ORM in the basic modeling concept, *fact,* and the involvement of the fact types and the roles for the modelling of a domain that an entity belongs. Henricksen, Indulska & Rakotonirainy (2003) extended the ORM to allow fact types to be categorized, according their persistence and source, either as static or as dynamic. The latter ones are further distinguished depending on the source of the facts as either *profiled*, *sensed* or *derived* types (Strang & Linnhoff-Popien, 2004).

Object Oriented Models

Object oriented context modeling approaches are using the benefits of the object oriented programming possibilities, the encapsulation and reusability. With these characteristics of object oriented programming, problems raised with the dynamic context in ubiquitous environments are covered and the details of context processing are encapsulated on an object level and hence hidden to other components.

Logic-Based Models

Logic-based models have a high degree of formality. Typically, facts, expressions and rules are used to define a context model. A logic based system is then used to manage the aforementioned terms and allows adding, updating or removing new facts. In these systems the context is defined as facts expressions and rules. The contextual information is usually added to, updated in and deleted from a logic based system in terms of facts. Logic based models have high degree of formality.

Ontology-Based Models

Ontologies represent a description of the concepts and relationships. Therefore, ontologies are a very promising instrument for modelling contextual information due to their high and formal expressiveness and the possibilities for applying ontology reasoning techniques. Due to the evaluation of the context model made by Strang & Linnhoff-Popien (2004), ontologies are the most expressive models and fulfil most of their requirements. Some of these requirements are *simplicity, flexibility and extensibility, generality, expressiveness.*

6.1 Ontologies in Context Awareness

6.1.1 Context awareness in distributed collaborative creativity and knowledge processes

Shaping and implementing context awareness in idSpace focuses on the notion of capturing and sharing new knowledge as it is being created, taking advantage of existing knowledge, group member profiles, their previous work and other information. This is a dynamic process in which minimizing the number and the size of the obstacles interfering with this intensive, creative process, is of paramount importance for creating the right conditions for creativity. This enhancement of the creativity process can be achieved through the use of ontologies and the use of the associations between the attributes within these ontologies.

6.1.2 Context in ontologies

Simply put, the context of a process or a system that facilitates that process also comprises knowledge. Specifically, it is knowledge about the user, the supporting system, the task at hand and the social environment in which it takes places. To capture this knowledge, in essence is not much different from capturing the knowledge of a certain domain. Ontologies are the most suitable instruments for capturing both types of knowledge and linking them into one associated knowledge map (Degler & Battle, 2003; Strang & Linnhoff-Popien, 2004).

6.1.3 Contextualizing knowledge

Using an ontology, the context of a creativity and knowledge intensive process, such as supported by the idSpace platform, will result in knowledge of its own. Therefore, when communicating ideas or knowledge resulting from the creativity sessions, this knowledge can be presented alongside its contextual setting. This will make the communication on ideas and knowledge, as well as the collaboration in a distributed setting, in which people can not depend on non-verbal communication and shared environments, much more effective and efficient. It can literally transport a significant part of the setting in which users ‘normally’ cooperate across the web to other users of the distributed collaborative platform.

6.2 User Modelling

Personalized or user adapted eLearning applications use user characteristics as one context dimension to adapt user interface, application behaviour, recommendation to support learning/knowledge creativity. Basic assumption is that users differ in learning, thus to make learning more effective, the system needs to understand the differences and adapt learning support accordingly (Dolog, 2006).
User modelling in eLearning applications tries to observe and deduce part of the context relevant to the learner, learning activity and learning environment. It usually comprises characteristics such as learning performance, abilities, learning style, competencies, preferences, interests and so on. Learner models can be observed or computed at different levels:

- Individual – is concerned with user characteristic of a concrete individual user and the main assumption is that in principle every user is different;
- Stereotypic – the main assumption is that users can be classified into several stereotypes which makes the personalization or adaptation strategies easier to implement as they do have to consider just couple of stereotypes as variability in learning;
- Role based – tries to base user characteristics (individual or stereotypic) based on the role in a company or learning activity and infer user characteristics based on it;

- Group based – sometimes the learning happens in a group of learners where individual or different stereotypic characteristics of participating learners may result in a conflicting recommendations or adaptation. For this reason, group based user modelling approaches have been considered.

The vast majority of currently available eLearning or intelligent tutoring systems employ so called ***overlay modelling*** for user modelling. The main idea is that an individual, stereotype, role or a group characteristics can be modelled as an overlay of an expert in particular field of training which an eLearning system offers, e.g., Personal Reader (Dolog et al., 2004), AHA (De Bra & Calvi, 1998), Interbook (Brusilovsky & Schwarz, 1993). The overlay can be represented simply as a subset of concepts which form a domain (Brusilovsky & Schwarz, 1993), sub graph of a semantic network of concepts connected by semantic relations (Dolog et al., 2004; Brusilovsky, 2003; Asnicar et al.), or *as references in a semantic models such as ontologies for a learner (Dolog et al., 2004). The main advantage of the overlay model is its simplicity, easiness to implement it and use within adaptation strategies. The main usage of the overlay models is to identify background knowledge or acquired competencies initialized either through an entry diagnosis for example through a questionnaire or an interactive dialog or through an automatic observation of performed actions on a learning material or examination test items. The overlay knowledge is then used as a filter of those items of learning material or activities which are already learned. Such an overlay model can be useful in observations of ideas the idSpace user/learner already explored in the idSpace environment.

A feature of the overlay model is that it contains only knowledge which is considered to be true. Negative knowledge or explicit false knowledge can also be sometimes useful especially in problem based learning where a problem solving and exploration of mistakes bugs and wrongly solved problems could help learners to understand the domain better. For these reason, so called ***perturbation modelling*** approaches have been considered for example with typical learner's misconceptions when learning terminologies with STyLE-OLM (Dimitrova, Self & Brna, 1999) and it is used to advice learners when a misconceptions in their learning or connecting concepts is identified. Similarly, in idSpace it can be useful to advice learners in learning about new ideas. However, creative problem solving and generating new ideas is a much more complex situation than learning terminologies. Therefore, this remains as an open issue also regarding the feasibility.

Another source of knowledge about user behaviour is a user activity or construction of knowledge. For example, ELM-ART (Weber & Brusilovsky, 2001) uses so called ***episodic learner modelling*** to overlay a learner activity in programming and identify steps which lead to both, successful as well as unsuccessful, solutions. Episodic learner models comprise besides concepts similar to the overlay models also rules which describe different ways to solve the problem together with their prioritizations. The concepts and rules help learners to explore the knowledge space more efficiently.

This is a very attractive option for the idSpace project as the rules could perhaps be used for exploration of ideas as well as for triggering appropriate creativity technique(-s). However, the applicability to the creativity domain remains an open question, since it is difficult to accurately describe the situations/problems for which a creativity technique is suitable.

The user modelling task is based always on incomplete information or observations. However, in some cases, uncertainty or probabilistic reasoning needs to be applied to properly reflect uncertainty in the facts derived from incomplete information. Therefore, so called ***probabilistic user modelling*** such as Bayesian networks have been considered in adaptive eLearning applications. Tedesco et al., (2006) employs Bayesian networks and probabilistic reasoning for inferring learning styles of a learner from distributed fragments to be used in adapting learning instructions of virtual campus of Politecnico di Milano. Mayo & Mitrovic (2001) use Bayesian networks together with decision theories as normative theories to prescribe which next tutorial step should be recommended in an intelligent tutoring system based on learner model described as a Bayesian network. Probabilistic models are interesting in the idSpace context but go beyond the scope of the project.

Constraint-based learner modelling shares the same assumption with probabilistic user modelling that the user model is always incomplete. However, it provides a different solution to this fact. It is based on the assumption that the correct solution to a problem cannot be achieved if a learner results in certain states of solving a problem. Therefore, the learner models can be reconstructed as constrains over an expert knowledge and signalled if the states are present in the model (Ohlson, 1996). This approach is also interesting for idSpace but goes beyond the project scope.

6.3 Adaptivity Methods in Context Aware Applications

The term *adaptivity* is unfortunately not well and homogenously defined in the literature; rather many different meanings are assigned to it. For us, an *adaptation* is any act by which the main trajectory of behaviour of a system or its structure is modified. In particular, modus changes of a software system are examples of this. As *adaptivity of a software system* we then define the capability of a software system to perform an adaptation. Depending on whether this adaptation is explicitly triggered externally (e.g., by a human who requests a modus change) or whether the system derives the necessary information on when and how to adapt from other data, we can differentiate *explicit adaptivity* and *self-adaptation*.
If we discuss context-awareness[1], the systems are supposed to take environmental information into account.

[1] Anind Dey defines context as any information that can be used to characterize the situation of an entity. An entity is a person, place, or object that is considered relevant to the interaction between a user and an application, including the user and applications themselves. Building on this he defines a system as being context-aware if it uses context to provide relevant information and/or services to the user, where relevancy depends on the user's task. Dey et al, 2001.

This actually points towards self-adaptivity as opposed to explicit adaptivity. However, as the definition of context-awareness (at least the one given in Dey, Abowd & Salber (2001) is deliberately broad, this also means that not every context-aware system is also self-adaptive. However, there is a significant and important overlap of the two definitions.[2]

Self-adaptation has many roots and has been widely researched in various scientific fields. In particular, research in self-adaptation has roots in fields as diverse as: fault-tolerant computing, distributed systems, biologically inspired computing, distributed artificial intelligence, integrated management, robotics, knowledge-based systems, machine learning, and control theory (Cheng et al., 2008).
When we want to describe the various behaviours a self-adaptive system may have, we may actually use variability modelling techniques (Kang et al., 1990; Schmid & John, 2004). While variability modelling techniques were initially defined in order to describe different behaviours at development time, the associated techniques are meanwhile also used to describe different runtime behaviours. This led to the introduction of the term Dynamic Software Product Lines (DSPL) (Hallsteinsen et al., 2008).

However, not all forms of self-adaptation can be well expressed in terms of a variability model. While this approach is particularly applicable for adaptation in terms of discretely different behaviours (and configurations) of a system, it is less applicable in situations where the form of adaptation is of continuous nature. Basically, variability techniques can be well used in cases where we can characterize the form of adaptation by introducing different operation modes in the system description.

In cases, where it is not possible to use such discrete formulation efficiently to describe the form of adaptation (e.g., think about a computer system dynamically allocating computing resources to different tasks), self-adaptation can be most adequately described by using some optimization function. This is in general a principle of adaptive systems. They always rely on some form of control loop. Even in the case of discrete changes (e.g. modes) this is the case (Cheng et al., 2008).
Today there are a number of challenges in developing self-adaptive systems. According to Cheng et al., (2008) these can be sub-divided into Requirements, Modelling, Engineering, and Assurances.

Requirements focuses on how to describe adaptive systems, specific challenges include here: uncertainty: when does a change in the environment necessitates a reconfiguration, as due to uncertainty the boundary may be fuzzy? (Robinson, 2003).
Requirements reflection: this means that systems should be able to reflect upon their current requirements and adapt their behaviour accordingly (Finkelstein, 2008). Goal refinement: this goes even a step further and requires that systems can identify (to some degree) their own requirements (Kramer & Magee, 2007).

[2] Note, that some authors use the terms context and adaptivity rather loosely. They, therefore, often identify the two terms.

According to Cheng et al. (2008) a major challenge, is the fact that existing modelling approaches do not support the necessary range of constructs, and they mention techniques for adaptability, for the place of the change, the impact of adaptability, etc. At least a subset of these problems however is typically addressed by variability techniques (Kang et al., 1990; Schmid & John, 2004).

With respect to engineering, the control loop is the central concept. However, there is currently a lack of support for prototypical aspects of realizing the control loop, like modelling it, providing reference architectures for these types of systems, etc. Finally, assurance is a major issue. Again this also requires modelling, but also verification and validation on top of these models are necessary. Because of the complexity of this issue sometimes designers resort to runtime issues like runtime proofing or build-in testing (Gross, Atkinson, & Barbier, 2003). Despite these challenges currently still quite a number of systems are built that are self-adaptive and context-aware, e.g. (Bencomo, 2008; Lee & Kang, 2008).

7. Frameworks and Architectures

Bardram (2005) presents the JCAF – Java Context Awareness Framework which is a java based context-awareness infrastructure and programming API for creating context-aware computer applications. JCAF is a distributed, service oriented, event based and secure infrastructure. JCAF relies on having a set of distributed context services that cooperate in a loosely coupled peer-to-peer or hierarchical fashion. JCAF´s architecture is modifiable, event based and secure. It is modifiable and extensible at runtime and not at design and compilation time. JCAF services, monitors, actuators, and clients can be added to the JCAF runtime infrastructure while running.

Dey, Salber & Gregory (1999) present the Context toolkit. The Context toolkit addresses the distinctions between context and user inputs. It enables application developers to built context aware applications, by introducing three main abstractions, *widgets, aggregators* and *interpreters.* A widget is a component that is responsible for acquiring directly from a sensor. The aggregators can be considered as meta-widgets, taking all capabilities of widgets and they also aggregate context information of real world entities and act as a gateway between applications and widgets. Interpreters transform low-level information into higher level information that is more useful to applications. All of the components share a common communications mechanism (XML over HTTP) that supports the transparent distribution.

In Burkle et al. (2006) an agent-based infrastructure is presented which provides integration and collaboration of autonomous, context aware services in a heterogeneous environment. It describes the Computers in the Human Interaction loop (CHIL) agent infrastructure and how the objectives of the proposed architecture were achieved.

Kasim et al. introduce an architecture based on the Model-View-Controller for the development of interactive context aware applications. With the MVC architecture and its capabilities the architecture attempts to make the communication between the user and application more intelligible in several ways.

The Context Fusion Networks (CFN) was proposed by Chen, Li, & Kotz (2004). It allows context aware applications to select distributed data sources and compose them with customized data-fusion operators to a directed acyclic information fusion graph. Such a graph represents how an application computes high-level understandings of its execution context from low level sensory data. CFN was implemented by a prototype named Solar, a flexible, scalable, mobility-aware, and self-managed system for heterogeneous and volatile ubiquitous computing environments which first described by Rowston, Druschel, & Pastry (2001) and it consists of a set of functionally equivalent nodes, coded Planets, which peer together to form a service overlay using a distributed hash table (DHT) based P2P routing protocol such as Pastry.

Hong & Landay (2004) proposed the Context Fabric (Confab), a toolkit for facilitating the development of privacy –sensitive ubiquitous computing applications. Confab provide a framework of privacy mechanisms that allow developers and end-users to support a spectrum of trust levels and privacy needs. This framework is designed such that personal information is captured, stored, and processed on the end user´ s computer as much as possible. Context Information is modelled as Infospaces which can contain context data as well as sources of the context data. Confab focuses on privacy, and it does not address traditional distributed system requirements such as mobility scalability component failures and deployment / configuration.

8. Adaptation in Creativity Support Tools

Honig & Rastain (2003a,b) mention that the first step of a creative process is the determination of the kinds of an idea that a team is looking for, and after that the second stage is to use creativity tools for the generation of innovative ideas. Honig & Rostain (2003a,b) separate the creativity tools in three categories: *1. Warm-ups to get the creativity juices flowing, 2. Idea generation tools to maximize creativity output and 3. Energizers that reinvigorate the group and increase group concentration.*

A warm up is an activity that initializes the problem in order to help on the creation of a mindset which will be used for generation of an innovative idea. The idea generation tools are used to create ideas, and their purpose is to maximize the creative output-answers to a problem or challenge.Energizer is an activity which is used to make the users more active and creative (e.g. when users seem to be tired or there is lack of enthusiasm).

Through the years a lot of creativity techniques were proposed. Based on these techniques, or combination of them, a lot of creativity tools were proposed and developed. Here we will give a brief description of some of these tools and their characteristics, but see D2.1 for a more extensive overview:

- **Bubbl-us** (www.bubbl.us): The application provides a shared whiteboard, where the participants can post their text contributions visualized as bubbles. (Florian et al.). Bubbl-us supports three creativity techniques, *1. Mind Mapping, 2. Brainstorming and 3. Idea organization.* It is a web based application. It does not support the real time collaboration. It supports collaboration via file exchange like importing XML files and exporting XML, HTML or images of the results.
- **Comapping** (www.comapping.com): Comapping is a web based application used for quick and intelligent problem solving. It supports team collaboration for solving a problem, even if the teams are separated by geographical or time-zones. It supports real time collaboration and asynchronous sharing. The supported techniques of comapping are *1. brainstorming, 2. Mind mapping (left to right) and 3. Problem Solving.* It supports the importation and exportation of data from and to other creativity tools formats like Freemind, Mindmanager and MeadMap, and Microsoft office formats like rich text, presentations, excel, etc. The maps can be downloaded and stored for offline usage. The metamodel supported is the *left to right mapping.*
- **Mead Map** (http://www.meadmap.com): Mead Map is almost identical to commaping but mead map is more educational oriented. The creativity techniques supported by Mead Map are: *1. Mind Mapping, 2. Taking Notes 3. Researching, 4. Planning Events and 5. Group projects.* It supports real time collaboration and asynchronous sharing.

It supports the importation and exportation of data from and to other creativity tools formats like Freemind, Mindmanager and Comapping, and Microsoft Office formats like rich text, presentations, excel, etc. The maps can be downloaded and stored for offline usage. The metamodel supported is the *left to right mapping.*

- **MindMeister** (www.mindmeister.com): MindMeister is a web based application that supports *1. Mind mapping* and *2. Collaborative Brainstorming* techniques. It supports importation and exportation from Freemind and Mind manager creativity tools. It exports data in GIF, JPG and PNG format images, in rich text and PDF formats. It also publishes password prodected maps. The metamodel used in Mind Meister is *traditional directional maps.* It supports real time collaboration.
- **Mindomo** (http://www.mindomo.com): The supported creativity techniques of Mindomo are: *1. Mapping, 2. Problem solving* and *3. Brainstorming.* It supports importation and exportation from Freemind and Mind manager creativity tools. It exports data in GIF, JPG and PNG format images, in rich text and PDF formats. The maps are published online. The metamodel used for Mindomo is *Traditional (centered) or directional maps (bottom layout/top layout).*

The above list of tools is just a subset of existing creativity-support tools, which can nevertheless be regarded as representative.

Based on the above creativity tools we examined whether they support adaptive methods, and the type of adaptation they support. We worked with the creativity support tools and we evaluated them according to: Format Adaptation, Appearance Adaptation, Encapsulation Adaptation, User Profile Adaptation and Context Adaptation. With Format Adaptation we examined the flexibility of the tools to export the created mind maps in multiple file formats. For Appearance Adaptation we examined and evaluated the modules supported by each tool, that help the user design the mind map according to his visual or aesthetic needs. Size Adaptation refers to how the mind map is changing when the window size is changing or when it is monitored in different resolutions. For User Profile adaptation we examined the information that is collected for the user and how this information is used. On this type of adaptation we focused on the user collaboration methods used in the tools. User Profile adaptation was examined in relation with Context Adaptation where we expected to see some automation methods, where the system would recommend the collaborators based on some context elements, such as the idea subject, the specialty of the participants, the content etc.

Table 1: Adaptation in Creativity Support Tools

Creativity Support Tool	Format Adaptation	Appearance Adaptation	Size Adaptation	User Profile Adaptation	Context Adaptation
1. Bubbl-us	Export as JPG or PNG mage, XML format, HTML format plain or with colors.	Zoom in/out. Multiple colors of the bubbles. Shadow effects. Custom background Positioning. Print area set up.	No size adaptation	Panel where the user can save his projects, and create a list with friends for file exchange.	None
2. Comapping	Export as web page using bullets and web page with layout, Comapping, MindManager and Freemind file format, RTF, Microsoft Project File, CSV file.	Zoom in/out, Text formatting, Hyperlinks creation, Fonts formatting, Emoticons.	The nodes of the mind map are collapsing or expanding according the size of the window.	Panel where the user can save his projects. Create friends list (There is not a list with people. The user adds his friends and sends invitation to join his team).	None
3. Mead Map	*same as Comapping*	*same as Comapping*	*same as Comapping*	*same as Comapping*	None
4. MindMeister	Export as Freemind and Mindmanager file format, RTF document, PDF, Image as PNG, JPG and GIF format.	Text formatting, use of hyperlinks and fonts formatting and use of emoticons.	No size adaptation	Presents the user's information according the project he is working on. Collaboration with people (There is not a list with people. The user adds his friends and sends invitation to join his team).	Uses the Idea title as keywords to Google, Wikipedia and http://del.icio.us/tag/
5. Mindomo	Export as image in GIG,PNG,JPG format, Plain Text, RTF, PDF, Customized Layout format of the Map style	Zoom in/out, Text formatting, Change colors of the nodes, Create hyperlinks, Move nodes and set positions manually Use of symbols	Set width and height manually, No size adaptation on maximize/minimize window	Save as public or private. If it is public other people can access it and modify it. Space for the user's maps.	None

Table 1 summarizes the types of adaptation encountered in five of the major creativity support tools that were previously described. It became apparent that the examined creativity support tools do support adaptation methods but none of them supports adaptive methods based on the context and the users in combination with the context.

Additionally, we discovered that although some support real time collaboration (users being in different geographical positions), allow file import and export in several formats, include features such as, messaging and chatting during the creativity process, selecting people that will form a team from a list of friends-contacts that the user has, customization of the tool's interface, they have several shortcomings.

The system in most cases plays a passive role and just allows the user to do things, including customizing the layout, or selecting his group members. In the plurality of the creativity support tools that we reviewed, a big emphasis is given on the interface and how the interface can be customized by the user in order to match his/her preferences. Two variables that we also took into consideration during this review, concerned the adaptive behavior of an e-learning interface, as proposed by Uruchruti & MacKinnon (2005). These were: Feedback/Advice and Interaction Techniques. We discovered that these system characteristics were not included in most tools studied.

We believe that phases of the creativity process (Schneidermann et al., 2006) such as considering alternatives, gathering information and consulting/interacting with peers or experts/mentors are not currently well supported and that they could significantly be improved in that respect and users can benefit from useful recommendations made by the system.

Frequently in organizations involved in designing innovative products, users or groups of users working on a product are not aware of other similar products or product components developed even within the same organization, or in others. This potentially leads to work being done over again, which really results in a waste of resources. Similarly users may not be acquainted with all other employees, or most importantly may not be aware of their expertise, which again leads to resources not being fully exploited.

Therefore, since software systems, in our case creativity support tools, can be 'aware' of all users/employees within an organization (and possibly beyond) and their competences, expertise, preferences, social networks, etc. and can maintain repositories of resources of any kind, they are in a position to potentially make recommendations concerning peers, mentors, similar products, useful resources, and so on.

9. Context Awareness for idSpace

Yanlin & Yoneo (2007) proposed a framework of context awareness support for peer recommendation in the e-learning context. The proposed framework is based on a three dimensional model of context elements, which define the e-learning context. The three dimensional context awareness model for the peer recommendation includes context awareness to the *knowledge potential, the social proximity and the technical access.* In the context awareness framework proposed by Yanlin & Yoneo (2007), the e-learning context is described as a result of the interaction of three key elements, which are the knowledge, human and technology.

The knowledge context deal is not only the strategic resource for learning practice. Knowledge context is the definition of the knowledge objects and the relation amongst them and the influenced complications they enfold for the e-learning, like knowledge backgrounds, needs and interests.

The social context is related to the human dimension of the framework and is defined based on the human as the subject of learning activities, human-based social network and human-based social culture. The social context deals with the social cultural, psychological and emotional influences on e-learning, such as social-cultural backgrounds, social distances, social roles/responsibilities and social prestige or status in the e-learning context.

The technical context refers to the use of the technology and how technology is used to overcome barriers of time or space, that may caused during the e-learning process. It deals with the performance of the system, so the e-learning process is reduced to a collaborative process. Technical context includes factors that may influence e-learning such as technical media, learners preferences, skills or time proximity.

The brief review of creativity support tools that was carried out demonstrates that the current tools offer only very basic forms of adaptation. IdSpace will be a platform that provides an environment that allows designers of innovative products to collaborate, but most importantly to elaborate ideas and designs that have emerged earlier on, in previously held sessions with their own group or in groups of others. idSpace supports the creativity process. Context is fundamental to creative processes. Support has to keep track of the context, primarily in the form of suitable adaptations. Only then can it be used to the team’s advantage. The idSpace platform will support context awareness and make useful recommendations to the user or the group.

Depending on the idSpace user’s competences, preferences and ideas he is working on, the system will be able to suggest suitable users to collaborate with and offer personalized assistance.

Assistance can be in the form of relevant ideas previously developed, relevant resources (e.g. literature, URLs, etc.), recommended (learning) strategies, recommended techniques to use in a particular situation, etc.
For the purposes of idSpace, we discern the following contextual elements:

User: This may include someone's competences, preferences, etc. This information defines the profile of a user. This profile can for example be used in creating balanced teams or in establishing the qualification to perform a task.

Social Environment: The social background of the users and the social environment in which the learning (as seen in idSpace context) takes place. This possibly includes information such as group composition, roles played in the group and such.

System: This may include information such as the software or idSpace platform tool used at a given time.

Task (also referred to as 'ideation'): Information about a task including which project it concerns, the specific activity, the objective, the owner and the stakeholders of a task.

The collective information under these elements forms a profile, which is used as input for setting up creative processes in learning systems. The profile defines what role a participant plays in the process and thereby what the context is in which someone functions. In order to be able to make maximum use of these contextual elements, a conceptual model must be developed for context awareness in creative processes. Ultimately, the *idSpace* platform will use this conceptual model to define how the ideas and insights from the creative process will be processed and represented. By building the conceptual model for context awareness on semantics, a *very rich model and knowledge structure* can be constructed *that offers the users the required flexibility* as well as a sound basis *for storing and representing knowledge and ideas.*

We expect idSpace users to experience context awareness, in the adaptive presentation and recommendation of ideas already generated in innovation generating sessions and in the recommendation of tools to be used in the refinement process. We also expect it to be reflected in the pedagogical design patterns. Context awareness comprehensively defined in this manner will help users derive inspiration from prior knowledge, created and preserved by other colleagues before them. In this respect, *idSpace* builds upon the state of the art, employs an interdisciplinary approach and takes advantage of techniques such as user and group profiling, semantics and ontology's. We anticipate significant advantages over other more restricted approaches and expect to significantly impact the development of future collaborative learning systems.

9.1 Contextual elements

9.1.1 User

idSpace will initially adopt a combination of episodic and overlay user model mentioned in the literature review above. It will follow an approach of standard based learner modelling suggested in Dolog & Schaefer (2005) based on combination of open specifications for learner profiles such as IEEE PAPI and IMS LIP where learners' learning activities are recorded in learner's performance and portfolios. Learner's portfolios and performance is further linked to competencies.

- **Learner's performance.** Learner's performance refers to performed learning activities in the idSpace platform. As the idSpace platform should deal with idea generation, exploration, and elaboration, the learner's performance will refer to such activities. The generated, explored, and/or elaborated ideas are recorded with an identification of idea concept preserved in the creativity techniques conceptual model. The idea is certified by a type of activity performed. Furthermore, a competence entry related to the acquired or improved skill or competence of a learner is recorded as a competence identifier. As the evolution of an idea throughout different sessions and different creativity techniques applied to the idea elaboration is traced, the assumption is that it will be possible to derive a level of competence of a learner based on that.

- **Learner's portfolio.** Learner's portfolio is an aggregation of learner's performance records. Each learner's performance contributes to the learner's portfolio. Portfolios are collections of the performance records which are further means for analysis of learner's competence evolution. Portfolios can further be structured into sub-portfolios based on activity types, competence types, as well as different domains of generated ideas.

Learner's competence. Lerner's competence is a concept which defines an ability of a learner's to perform certain activities. The competencies can be defined at several levels and can have different ontologies. In the present context competence ontology will refer to the mastering of different creativity techniques and their types. Therefore, the learner's competence recorded in the learner's performance is an overlay over creativity techniques domain in this case. Another example is a competence structure in the subject of explored area. A creativity session can be for example dedicated to brainstorming which should find a new material for a new space ship. The competence structure than refers to the different materials which a learner can explore, is able to apply in idea generation as well as ability to combine such materials. As can be seen, competencies are highly domain dependent so it is up to application partners to define them.

- **Application of the aforementioned user model.** The aforementioned model can be applied when forming a group for a creativity session based on different criteria. It can be further applied for recommendation of ideas generated by other people similar to competences and performance of a user or ideas which could improve his competences when exploring the ideas.

- **Model.** Concepts in the model are: User, Learner, Performance, Portfolio, Competence, and Idea.

9.1.2 Social Environment

"Social awareness is a mental concept where a learner becomes aware of the social network that follows him or her while moving across the different learning contexts".

IdSpace is a platform which supports collaboration, learning and the generation of innovative ideas and knowledge. The involvement of social context is mandatory. Yanlin & Yoneo (2007) mention that social context is related to human beings and they give three dimensions of the human involvement: ***1. the human as the subject of learning activities, 2. human based social network and 3.human based social culture.*** The generation of an idea is usually an individual process followed by knowledge transfer to other people, or knowledge received by others. The collaborative process is usually used as internal processes in team-groups, companies or organizations.

In the last years the collaborative work and the creation of social networks is used widely through internet technologies. The implementation of such networks is done with multiple methods like instant massaging, forums, blogs, chat etc. The difficulties rising with the existing methods are dealing with synchronous and asynchronous communication, relevance of participants on a subject. *Within social navigation we can distinguish between direct and indirect social navigation. While indirect social navigation aims to provide information about what other people do or have done, direct navigation involves direct communication between two or more people* (Dieberger, 1998). An example given by Malzahn, Zeine & Harrer (2005) is the structure and the categorization of threads inside a forum. If a forum is large, most persons know a subgroup of persons, directly involved in topics where they write themselves. Other persons are potentially interesting to get to know because they share similar interests or they are known as experts in other parts of the forum.

IdSpace as a collaborative creativity tool, and because of the existence of more than one creativity techniques aims to support the creation of social networks which will support cooperative work and learning based on the CSCW matrix introduced by Johansen (1988). But to define the social environment as a context element for the idSpace project we have to conceptualize the social environment. The social environment as a context element in idSpace platform will be used for the creation of social groups, to define the roles within a group and to define the stakeholdership within a group.

"When a computer network connects people or organisations, it is a social network based on friendship, coworking or information exchange, etc"
(Garton, Haythornthwaite & Wellman, 1997)

Based on the definition given above, we can define the *Social Environment* context element as the cluster of social networks. Social networks will consist of people with related interests and offer them the tools to exchange opinions. So in its turn, a social network will consist of "social groups". The social groups will be identified by a subject, which will describe a wide area of related topics. Each social group will have a forum, a wiki, a blog. The members of each social group will have a role inside the group For example, there are two Social Networks. *1. Computing 2.Environment.* In the Computing Social Network there are 3 Social Groups: *1.HCI interfaces, Web Design, Java Programming.* Each Social Group has its own forum, wiki and blog, The users of the social group have social roles which are defined based on the number of threads they posted, based on an evaluation of his answers from the other users, etc. That way, a user that belongs in Social Groups 2 and 3 of the example, might be "specialist" in Java Programming and "amateur" in Web Design. All members of the social group will be tagged automatically.

9.1.3 System

The idSpace platform strives to be not a one-size-fits-all model. Instead, mixing tools and automatically tweaking system functions should result in a platform which can be used in various settings. Some settings require a more formalized environment and other settings require an informal environment. In both cases, a creative process will be facilitated. One way to make a difference between the two options is to offer a set of concepts and associations as a starting point and delimiter for the creative session for formalized and restricted sessions and in the case of a more informal session offer a blank slate to be filled in by the users.
The system context element can be defined by the following discrete attributes:

- **Connection Speed:** the connection speed influences the collaborative procedure, and has importance on the proposed resources type. (E.g. multimedia, real time Skype conversations, etc.)
- **Type of device accessing:** The user may access the id Space from a PC, PDA, and Pocket PC. The interface and supported modules of idSpace must be adapted
- **3rd Party applications:** The system must search on the client´s PC for the necessary 3rd Party applications which are used in idSpace and if there are not installed, the idSpace must propose Links for Download (e.g. Skype)
- **Operating System Used:** Different plugins and 3rd party applications compatible with the OS.

9.1.4 Task

The refinement of the contextual element “task” is particularly characterized through the relation to the creativity technique that is later employed to achieve the task. Thus the context of the technique transitively defines the context of the task and vice versa. Figure 1 presents a first proposal for the “task” context model, which will potentially need to be refined in the next phase of WP3.

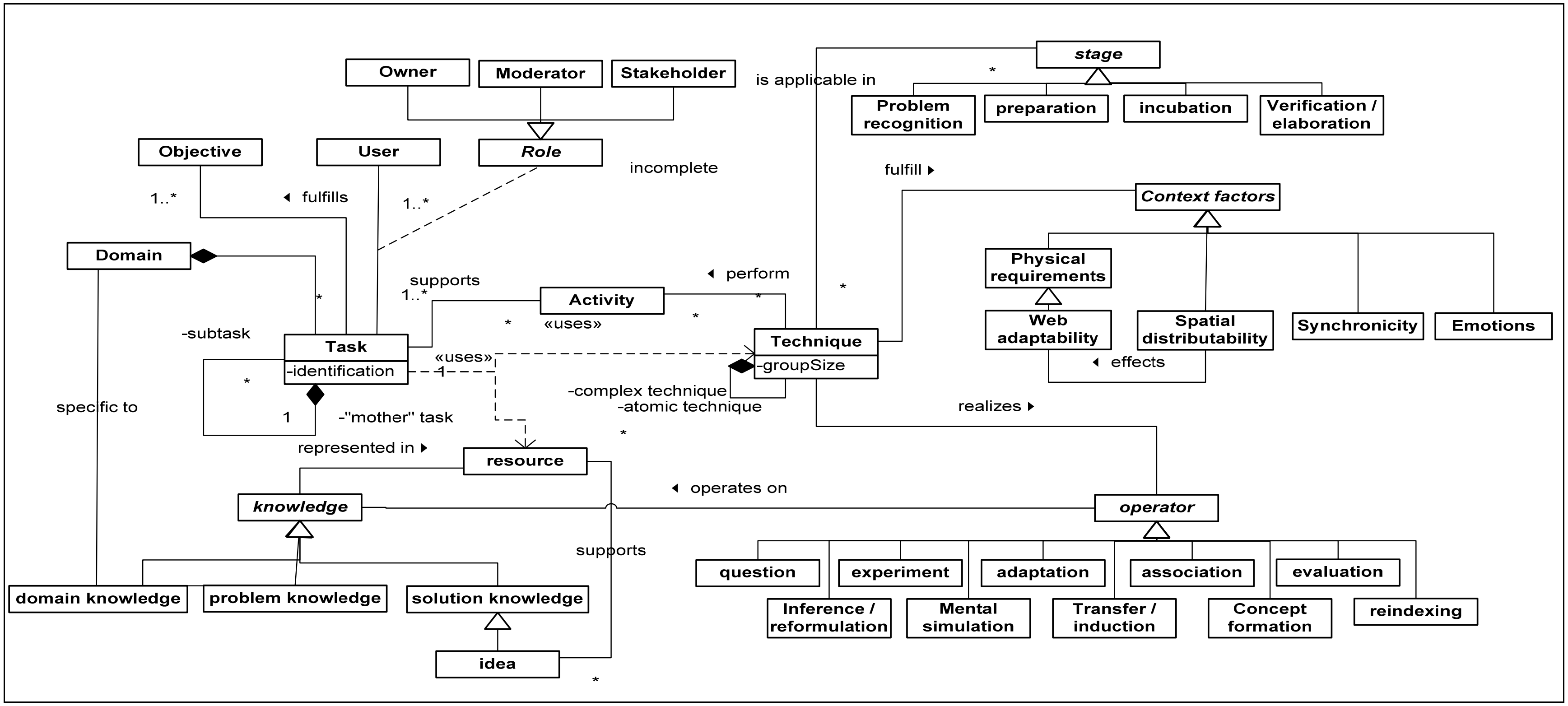

Figure 1: Task context model

Task: The task has a unique identification (here modeled as an attribute to it). As a "mother task" it can include subordinate task. (Sub tasks)

Objective: The high-level goal that shall be achieved through the execution of the task.

Technique The refinement of the contextual element "task" is particularly characterized through the relation to the creativity technique that is later employed to achieve the task. Thus the context of the technique transitively defines the context of the task and vice versa.

- A number of **context factors** determine the applicability of the technique in a given scenario:
    - **Physical requirements:** Does the technique require or include any physical (inter-) action, e.g. to provide a hands-on experience for users to literally get a hand on the idea.
    - **Spatial distribution:** Does the task allow the distribution of work over large distances as it occurs with web-based interaction or does the technique require its users to be at the same place.
    - **Synchronicity**: Does the technique allow asynchronous work on the task or do users have to work on them concurrently? In a web-based distributed environment people cooperating might work in different parts of the world and different time zones.
    - **Emotions**: Emotional expressions that influence the process of work on the task.
- A technique can be applied in different **stages** of the creative process. These are:
    - **problem recognition:** the identification of a problem. In this stage knowledge and ideas relevant to the problem solution are gathered and first problem solving attempts are made. Finding the problem (if not well-defined at the outset) is part of this stage.
    - **preparation**: Knowledge and ideas relevant to the problem solution are gathered and first problem solving attempts are made. Finding the problem (if not well-defined at the outset) is also part of this stage, too
    - **incubation:** At this point the first problem solving attempts have failed. This stage is characterized by an alternation of rest and work. This is sometimes called creative worrying.
    - **verification / elaboration**: The solution is verified for correctness, completeness, value, etc. This usually requires further elaboration in order to derive the final, full-fledged solution from the initial concepts.

Activity: The task supports certain activities that are performed by a technique.

User: Users participate in a task playing different roles such as:

- **Owner** of the task – the one who initiated it
- **Moderator**
- Other **stakeholders** who can be involved in some way

Domain: Subject area of the task

Knowledge – It can be divided into three basic types of knowledge:

- **Problem knowledge** - Knowledge regarding the problem itself
- **Domain knowledge** - Everything related to the subject area in which the problem (and the solution) is set
- **Solution knowledge** - Understanding of what can be possible (and impossible) constituents of the solution. This includes the solution itself.
    - **Idea**: An idea is a kind of solution knowledge as it proposes a certain strategy to successfully execute the task.

Resource: The intangible concept of knowledge is represented in a tangible resource. It represents a concrete knowledge item a task uses. Such a resource (e.g. relevant products, related publications) also supports the development of new ideas.

Operator: A technique realizes a number of so-called operators, basic activities or elements that a creative process can be structures into:

- **Questions**: asking for knowledge requires to identify what knowledge is needed.
- **Experiments:** sometimes active exploration and experimentation is required.
- **Association:** An association is a means of activating existing knowledge. This is usually based on aspects like similarity / contrast, cause-effect, etc.
- **Mental simulation:** mental simulation can be regarded as an "internal experiment". Similarly, it can identify knowledge. However, the experimenter cannot completely abandon his previous knowledge.
- **Concept formation:** New concepts are created (based on example).
- **Adaptation:** Existing knowledge (e.g., solution) is adapted to a new situation.
- **Transfer / induction:** Knowledge transfer is a standard technique for problem solving and restructuring knowledge. A typical example is analogical reasoning.

- **Inference / reformulation:** The reformulation and inference of knowledge relates to activities that lead to deriving new knowledge from already existing knowledge.
- **Reindexing:** Not the knowledge itself but rather the way knowledge is accessible is changed
- **Evaluation:** This is responsible to determine whether a new idea is actually a solution to the underlying problem.

9.2 The Context Awareness Ontology

In this section we present the description of the proposed ontology for Context Awareness in idSpace. The ontology will potentially need to be refined in the next phase of WP3. The ontology is based on the four context elements presented in the previous section:

- User
- Social Environment
- System
- Task

The information which is describing each one of the elements is processed into an ontology model. In processing the documents their contents was transformed to meet the demands of a proper ontology.

1. Some 'objects' in the documents did not end up as topics in the ontology, but rather as a property of another topic. For example, the Domain-object in the Task element document was dropped as a topic and instead an association to the knowledge-topic was created. For example, Learner Performance, from the User-Element document, was dropped as a topic and became a property of the Portfolio topic. The learner performance still exists within the ontology, but is the result of values for the User- and the Portfolio-topics.
2. Some 'objects' were merged into one new topic. This doesn't mean that they have disappeared, but only that they have a different ontological representation. For example, the Domain- and Knowledge-object in the Task element document were dropped as objects and instead a new topic Knowledge-item was created. The self-reference of this topic creates a network of knowledge items, which leads to specific knowledge items as well as more general domain items.

3. Specific values, such as ‘question’ and ‘experiment’ in the Task element document were kept out of the ontology model for reasons of clarity of the model. Instead they will be mentioned in the description of the model.

Specific values, such as ‘question’ and ‘experiment’ in the Task element document were kept out of the ontology model for reasons of clarity of the model. Instead they will be mentioned in the description of the model.

User vs. Learner

Learner was included as a concept. In the ontology *learner* is now one of the roles that a user might play.

Performance and Portfolio

Performance was included as a concept. This concept still exists in the ontology, but not as a topic itself. Instead it a collection of *Tasks* and *Activities* per *Creative Process* and per *Stage*. Later on, when designing and building the idSpace platform, the term Performance can be attached to this information.

The ontology itself can be seen in Figure 2 below and the definitions of the various topics and relations follow.

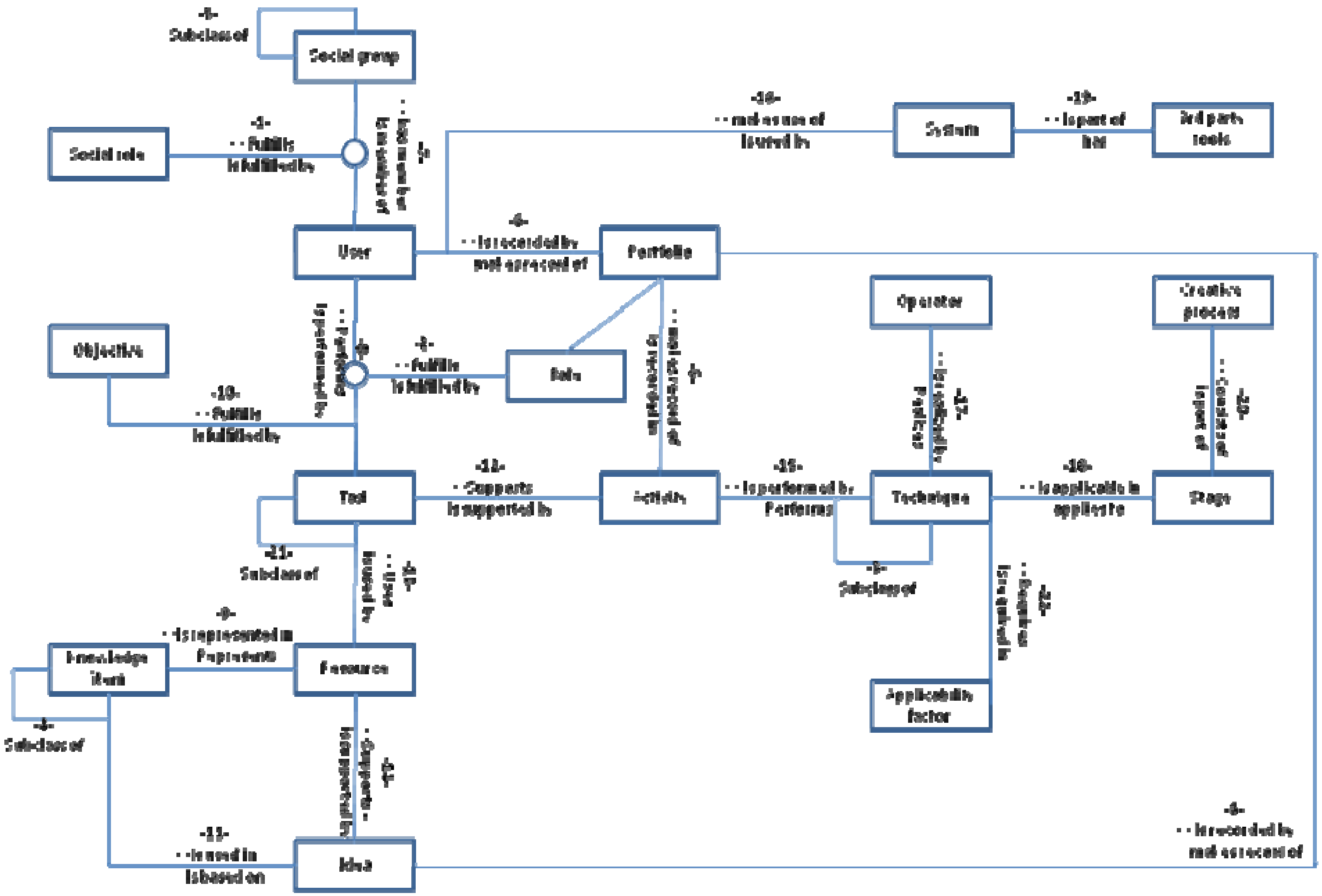

Figure 2: Context Elements Ontology

Topics

Activity

Definition: Activities are the main building block of a technique.
Property: Description

Applicability factor

Definition:
Property: Name *{i.e. physical requirement, spatial distribution, synchronicity, emotions}*
Description

Creative process

Definition: This is the main topic and core concept of the project.
Property: Description

Idea

Definition: An idea is a possible solution to a stated objective.
Property: Description

Knowledge item

Definition: This represents a piece of knowledge and may be related to other more general or
more specific knowledge items
Property: Description
Type *{'problem knowledge', 'domain knowledge', 'solution knowledge'}*

Objective

Definition: The quantified goal of a task
Property: Description

Operator

Definition: A basic activity or element that a process can be structured into
Property: Description
Type *{questions, experiments, association, mental simulation, concept formation,*
adaptation, transfer/induction, inference/reformulation, reindexing, evaluation}

Portfolio

Definition: This is the record of a user's activities and contributions in the creative process
Property: Description

Resource

Definition: This is the container of a piece of knowledge or information.
Property: Description
Type *{book, article, webpage, …}*

Role

Definition: This is the role a user may play in a task. A user can play more than one role in a task.
Property: Description

Social group

Definition: A social group, is a group of people identified by a common subject.
A social group may consist of several sub groups identified by more specific subjects.
Property: Description

Social role

Definition: This is the role a user may play in a social group.
A user can play more than one role in a group
Property: Description
Type *{local experts, answer people, conversationalists, fans, discussion artists, flame warriors, and trolls}*

Stage

Definition: A phase in the creative process.
Property: Description
Type *{problem recognition, preparation, incubation, verification/elaboration}*

System

Definition: This is the device from which a user accesses the idSpace platform.
Property: Description
Type *{PC, PDA, Pocket PC }*

Task

Definition: A task is a quantified piece of work needed to complete an activity.
It can consist of subtasks.

Property: Id
Description

Technique

Definition: This is a creativity technique that can be applied to a stage in the creative process.
A technique can consist of several atomic techniques.
Property: Description
Name

User

Definition: The user of the idSpace platform and the activities it supports
Property: Description
Name

3rd Party tool

Definition: This is a plug in or application from a third party, used with the idSpace-platform.
Property: Description
Name
Type
Property: Description
Name
Type

Associations

-1- Fulfills / Is fulfilled by

Definition: Defines the social role that a user plays as a member of a social group
Associates: ***Social role*** with ***-7- Has member / Is member of***
Properties:

-2- Fulfills / Is fulfilled by

Definition: The role a user plays in performing a task
Associates: ***Role*** with ***-8- Performs / Is performed by***
Properties:

-3- Subclass of

Definition: Creates a hierarchy of techniques being more complex or more atomic
Associates: ***Technique*** with itself
Properties:

-4- Subclass of

Definition: Creates a network of knowledge items
Associates: ***Knowledge items*** with itself
Properties:

-5- Subclass of

Definition: Creates a network of social groups, being either more general or more specific This also creates a differentiation between social group and social networks.
Associates: ***Social group*** with itself
Properties:

-6- Is recorded by / makes record of

Definition: Creates a portfolio for a user
Associates: ***User*** with ***Portfolio, Idea and Role***
Properties:

-7- Has member / Is member of

Definition: Connects a user to a social group
Associates: ***User*** with ***Social group***
Properties:

-8- Performs / Is performed by

Definition: Connects a user to a specific task
Associates: ***User*** with ***Task***
Properties:

-9- Is represented in / Represents

Definition: Indicates which knowledge items are represented in which resources
Associates: ***Knowledge item*** with ***Resource***
Properties:

-10- Fulfills / Is fulfilled by

Definition: Indicates which objective belongs to which task
Associates: ***Objective*** with ***Task***
Properties:

-11- Is used in / is based on

Definition: States which ideas are based on which knowledge items
Associates: ***Knowledge item*** with ***Idea***
Properties:

-12- Supports / is supported by

Definition: States which tasks support which activities
Associates: ***Task*** with ***Activity***
Properties:

-13- Uses / is used by

Definition: Connects a task with the resources it uses.
Associates: ***Task*** with ***Resource***
Properties:

-14- Supports / is supported by

Definition: States which resources support a certain idea
Associates: ***Resource*** with ***Idea***
Properties:

-15- Is performed by /performs

Definition: States the activities performed by a technique
Associates: ***Activity*** with ***Technique***
Properties:

-16- Makes use of / is used by

Definition: States which user uses which system
Associates: ***User*** with ***System***
Properties:

-17- Realizes / is realized by

Definition: Indicates which technique incorporates which operator
Associates: ***Technique*** with ***Operator***
Properties:

-18- Is applicable in / applies to

Definition: Connects a technique to a stage in the creative process
Associates: ***Technique*** with ***Stage***
Properties:

-19- Is part of / has

Definition: Associates a system with the 3rd party tools it can contain
Associates: ***System*** with ***3rd party tools***
Properties:

-20- Consists of / Is part of

Definition: States the stages that a creative process consists of
Associates: ***Creative process*** with ***Stage***
Properties:

-21- Subclass of

Definition: creates a hierarchy of tasks and subtasks
Associates: ***Task*** with itself
Properties:

-22- Requires / Is required in

Definition: Determines which Applicability factor applies to which technique.
Associates: ***Technique*** with **Applicability factor**
Properties:

9.2.1 Implementing Context Awareness

To implement context awareness we need to control the events and the actions of the user within the platform. A way of controlling the events within the platform is the usage of the **log events file**. The problem with a global log file is the fact that in the platform's log file, all the actions of all the users are noted. Thus it is difficult to isolate and control the actions of a user in particular. In order to resolve this issue we propose a solution that will also solve potential communication problems between the Microcosmos platform and Topic Maps. This solution will be revisited and refined in the next phase of WP3.

A common language that is recognizable from both platforms is a mark-up language like XML or RDF. The idea behind this proposal is to create personal log files for each user written in XML standard type. This file will have the user's id as a file name and it will contain all the action information in a tagged form. This information will be retrievable from the Microcosmos database and at the same time, it will be in a readable form for Topic Maps. In other words the personal XML file will be used as a proxy between the two platforms in order to create relations and associations. The XML information can be created and updated "on the fly", on the server side of the platform.

The idea of creating a personal file for catching the personal events entails the creation of a standard form of the tagged information which will be kept. The information that will be collected will obviously not be the actual data that a user will input. It will be the information regarding the location where data is saved and how this can be found (associations).

Implementation details including the methods and the procedures which will be used for the prototype of Context Awareness in the idSpace platform will be discussed and presented in the next deliverable. Here we briefly discussed a first proposal of "how" context awareness can be implemented.

9.2.2 Topic Maps Ontologies

idSpace will use ontologies to describe and implement Context Awareness, as well as to describe the semantic relationships between data, their properties and restrictions that apply. The environment where data, such as documents, notes, pictures etc. will be stored will be Microcosmos. To actually describe the metadata, like tagging the pictures, making statement on the documents etc. the Topic Maps[3] technology will be used.

A topic map is a structured semantic network in which data can be stored, using the relationships, properties and restrictions defined by the ontology. The data in the topic maps can be created, updated and retrieved using a Topic Map-specific Query Language (http://www.isotopicmaps.org/tmql/) like Tolog.

In contrast to flat file data storage where data is simply added to, for example, a database, storing or updating data in a topic map triggers a whole range of actions to ensure a valid semantic network. For example, when new data about a topic is added to a topic map while this topic is already known, the new and old data are automatically merged into one topic. Furthermore, the restrictions specified by the (meta-)ontology can be used to deduct new information. For example, if the ontology specifies that 'association A' can only exist between topic types 'B' and 'C', adding association A to topic type B and an unknown topic implies that the unknown topic is of topic type C.

Retrieving data from a topic map is different from flat file data retrieval: you cannot only retrieve the data itself, but you can also retrieve information about associations between data. Using TMQL you could even execute queries like "all topics of type A that do not have occurrence B but do have either association X or Y with topic B".

Not all the data is available to be used/ updated by any user, of any role – some restrictions apply. The Topic Maps ontology will define these restrictions and access control specifications. For this function we will use the Topic Maps Constraint Language (TMCL).

For retrieval and updating functions there are several implementations available. One of these is TMRAP, the Topic Maps Remote Access Protocol (Garshol, 2006).
In the remaining part of the section modeling guidelines for Topic Maps ontologies are shortly described.

[3] Notice that we speak about Topic Maps: which is the ISO standard and topic maps means a specific topic map, an implementation of a topic map for a particular topic map like a IdSpace topic map, see http://www.garshol.priv.no/blog/49.html

Topic Map Ontology Design

The Topic Map application life cycle consists of the following phases:

1. Define the application requirements
2. Define the ontology
3. Create the schema
4. Choose and customize tools
5. Implement the system and populate the topic map
6. Test
7. Revisit the ontology, refine the schema, modify applications

Ontology in computer science is:

- In practice the term has been used to mean almost any form of conceptual classification scheme:
  - from class hierarchies to more complex models
  - a true ontology, however, should have not just a class hierarchy, but also other "semantic relations" describing the relationships between concepts
- Usually, ontologies are considered to be a knowledge representation:
  - that is, they describe the real world
  - sometimes they describe it in ways which can be used by semi-intelligent agents
  - sometimes they describe it in ways which can be used for logical reasoning

Where we have upper and lower ontologies:

- Upper ontology
  - limited to general, abstract concepts
  - usually this means entity types, relationship types, and property types
  - entity types arranged in a type hierarchy
  - sometimes also especially prominent instances
  - some attempts have been made at creating standardized upper ontologies
- Lower ontology
  - basically a full ontology with instances minus the upper ontology

In Topic Maps this distinction is often mixed in the same model although one can separate them easily in different topic maps.

In the Topic maps standard any topic map is effectively an ontology and logical inferencing is supported through implementations of TMQL. To explain the difference between Ontologies and Topic Maps ontologies one better starts with some Topic map terminology:

- Ontology
  - topic types, association types, occurrence types, etc.
  - sometimes, some topic instances also belong here
- Constraints
  - rules governing classes of objects
- Schema
  - the combination of an ontology and constraints
- Schema language
  - a language for writing schemas
  - examples for topic maps: TMCL and several implementations like Morpheus, Ontopia and Networked Planet has.

Important to notice is that often:

- The ontology itself is part of the topic map
- That is, the topic types and association types etc are also topics
- Topic maps have no strict separation between ontology and instances
- The ontology can be described with topic map constructs

and it is important to express the constraints in TMCL:

- The ontology only tells you what types of things exist
- In ISO 13250 there is no way to express constraints on those types
  - No way to state the intention to let a certain topic define a class
  - No way to express the constraints

For the whole modeling process the following fundamental aspects will be applicable:

- There are many incorrect ways to model a domain, but no single correct way
  - There are often viable alternatives
  - The best solution depends on the application
  - The result may have an element of subjectivity
  - Modeling is an esthetic activity
- Ontology development is an iterative process;
  - As your ontology develops you will want to revisit earlier decisions and modify some of them
  - Fortunately, the topic map model is sufficiently flexible that this does not usually have a major impact on applications
  - Prototyping is a Good Thing

And for each Topic Maps Ontology design one can say that:

- It is a form of knowledge engineering or data modeling
- Often starts from a set of information resources
- one needs a necessary skill set:

- domain expertise
- understanding of the topic map paradigm
- communication skills
- data modeling skills

- It is normally a team effort involving:
  - subject matter experts, documentalists, topic map specialists, project managers, developers.

10. Conclusion

This deliverable aimed at describing the proposed employment of context awareness in the idSpace platform, having first offered an insight in the area of context awareness through the relevant State of the Art analysis. The State of the Art described and defined context awareness in the context of idSpace, after offering an overview from existing literature and research work in the area. We described the scientific research areas where the context awareness is implemented; we described the existing models and architectures of context awareness tools, middleware, and frameworks. We used the literature to describe the meaning of a context element and how the context elements are used for the implementation of context awareness applications.

Based on the definition of the context awareness and based on the use of context awareness for the idSpace, we finalized the State of the Art with the definition of the context elements for the creation of the context awareness Ontology. We analyzed the context elements for the platform, and we defined their attributes. With this definition, we managed to design the context awareness ontology in a schematic way, which will be used for its implementation in the idSpace platform. A first proposal concerning the implementation of the context awareness prototype was also presented. The Topic Maps technology will be used for the description and manipulation of the ontology (both for Context Awareness and in other aspects of idSpace) and therefore a relevant presentation was offered.

The research that was conducted for the creation of this deliverable helped us realize the necessity of context awareness in the idSpace platform. An important outcome of the work in WP3 up to this point is the fact that the methods to be used for the implementation of context awareness became clearer. Through the State of the Art for the context awareness and its description for the idSpace platform we are, now, in a position to proceed to the next step: to propose the prototype for the implementation of context awareness, which will in fact form the next deliverable. For proposing the context awareness prototype we will firstly consider the resulting requirements of the platform given the context awareness proposed in this document, and also evaluate the possibilities offered by the two basic platforms (Microcosmos and Topic Maps) for the implementation of this prototype.

References

Abowd, G., D., Atkenson C. G., Hong J., Long S., Kooper R., & Pinkerton M. Cyberguide: amobile context-aware tour guide, *ACM Wireless Networks,* 1997

Abhaya, A., Cravatts, M., & Krzyzanowski, P., An indoor wireless system for personalized shopping assistance. *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications,* pp. 69-74, 1994.

Bardram., J. E., , The Java Context Awareness Framework (JCAF) - A Service Infrastructure and Programming Framework for Context-Aware Applications. In *Proceedings of the 3rd International Conference on Pervasive Computing (Pervasive2005)*, 2005

Bencomo N., P. Grace, C. Flores, D. Hughes, & G. Blair, Genie, , Supporting the Model Driven Development of Reflective, Component-based Adaptive Systems, *Proceedings of the 30th International Conference on Software Engineering*, pp. 811-814, 2008

Bontas, P., E. (2008). Context Representation and Usage for the Semantic Web: A state of the Art, *Technical Report-Technical Report Freie University Berlin,* (B-04-30)

Brown, J., P., Bovey, J., D., & Chen, X., Context-Aware Applications: From the Laboratory to Marketplace, *Personal Communications, IEEE,* 4(5), pp.58-64, 1997

Brusilovsky, P., Eklund, J., & Schwarz, E. Web-based education for all: A tool for developing adaptive courseware, *Computer Networks and ISDN Systems Proceedings of Seventh International World Wide Web Conference,* 30 (1-7), pp. 291-300, 1998

Brusilovsky., P. (2003) Developing adaptive educational hypermedia systems: From design models to authoring tools, A*uthoring Tools for Advanced Technology Learning Environment*, pp. 377-409, 2004

Brusilovsky, P., & Schwarz, E., Student as user: Towards an adaptive interface for an intelligent learning environment. *Proceedings of World Conference on Artificial Intelligence and Education, AI-ED'93,* pp. 386-393, 1993

Bricon., N., & Souf, Conrad., R., N., Context awareness in health care: A review, *International journal of medical informatics,* volume 76, pp. 2–12, 2007

Burkle A., Muller., W., Pfirrmann., U., Schenk., M., Dimakis., N., Soldatos., J., & Polymenakos., L., An Agent based Architecture for context aware services supporting human interaction, *Proceedings of the 2006 IEEE/WIC/ACM international conference on Web Intelligence and Intelligent Agent Technology*, pp.146-152, 2006

Chen., C.,T., Cheng., Y., C., & Hsieh., C.,Y., Towards a Pattern Language Approach to Establishing Personal Authoring Environments in E-Learning, *Proceedings of the sixth conference on IASTED International Conference Web-Based Education - Volume 2,* pp.13-18, 2007

Chen., G., & Kotz., D., A survey of context-aware mobile computing research. *Technical Report TR2000-381*, 2000

Chen., G., Li., M., & Kotz., D., Design and implementation of a large-scale context fusion network, *Proceedings of the First Annual International Conference on Mobile and Ubiquitous Systems (MobiQuitous), IEEE Computer Society*, pp. 246-255, 2004

Cheng., B., Giese H., Inverardi., P., Magee., J., & Lemos., R., Software Engineering for Self-Adaptive Systems: A Research Road Map, *Software Engineering for Self-Adaptive Systems, Dagstuhl Seminar Proceedings*, 2008

Degler., D., & Battle., L., Can Topic Maps describe context for enterprise-wide applications?, *Proceedings of the Extreme Markup Languages 2003 Conference, Extreme Markup Languages*, 2003

Dey, A., K., Abowd, G., & Salber, D., A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications.In: *Human-Computer Interaction*, pp. 97-166, 2001

Dey., A., K., Salber., D., & Gregory., D., A., The Context Toolkit: Aiding the Development of Context-Enabled Applications, *Proceedings of CHI'99*, pp. 434-441,1999

Dieberger, Andreas, Social Connotations of Spatial Metaphors and their Influence on (Direct) Social Navigation, In Höök, Munro and Benyon (eds.) “Workshop on Personalised and Social Navigation in Information Space”, Stockholm, March, 1998.

Dimitrova., V., Self., J., A., & Brna., P., STyLE-OLM - an interactive diagnosis tool in a terminology learning environment. *Proceedings of Workshop on Open, Interactive, and Other Overt Approaches to Learner Modelling at AIED99*, pp. 25-34, 1999

Dolog P., Henze N., Nejdl, W., & Sintek., M., The Personal Reader: Personalizing and Enriching Learning Resources using Semantic Web Technologies. *Proceedings of AH2004 - International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, 2004

Dolog., D., Knowledge Representation and Reasoning in Personalized Web-Based e-Learning Applications, *Proceedings of Znalost*, 2006

Dolog., P., & Schaefer., M., A Framework for Browsing, Manipulating and Maintaining Interoperable Learner Profiles. *Proceedings of UM2005 - 10th International Conference on User Modeling*, 2005

Dustdar., S., A Survey on Context-Aware Systems, *International Journal of Ad Hoc and ubiquitous Computing*, 2004

Finkelstein., A., Requirements reflection, *Dagstuhl Presentation*, 2008.

Garshol L., M., TMRAP – Topic Maps Remote Access Protocol, *Lecture Notes in Computer Science*, ISSN: 0302-9743, 3873(2006), 2006

Garton, L., Haythornthwaite, C. & Wellman, B., Studying online social networks. *Journal of Computer-Mediated Communication, 3*(1), pp.75-105, 1997.

Gonzalez-Campos., J., S., A Comparison of Context-Modeling Representations for Context-Aware Applications, *Seventh Mexican International Conference on Computer Science (ENC'06)*, 0-7695-2666-7/06, IEEE, 2006.

Gross., H., G., Atkinson., C., & Barbier., F., Component Integration through Built-in Contract Testing, *Component-Based Software Quality, LNCS 2693*, pp. 159-183, 2003

Hallsteinsen., S., Hinchey., M., Park S., & Schmid., K., Dynamic Software Product Lines, *Computer*, 41(4), pp. 93–95, 2008

Harter., A., Hopper., A , Steggles., P, Ward., A., & Webster., P., The Anatomy of a Context-Aware Application, *Wireless Networks*, 8(2), pp.187-197, 2002.

Hartmann., M., & Austaller., G., Context Models and Context Awareness, *Handbook of Research on Ubiquitous Computing Technology for real time Enterprises, ISBN:1599048329*, pp. 235-277, 2008.

Henricksen., K., Indulska., J., & Rakotonirainy., A., Generating Context Management Infrastructure from High-Level Context Models, *Industrial Track Proceedings of the 4th International Conference on Mobile Data Management (MDM2003)* , pp. 1–6, 2003

Hong., J., I., & Landay., J.A., An Architecture for Privacy-sensitive Ubiquitous Computing, *Proceedings of the Second International Conference on Mobile Systems, Applications, and Services (MobiSys)*, 2004

Honig., B., & Rostain., A., Tools in support of creative collaboration, *ISBN:1560526874*, 2003a

Honing, B., & Rostain, A., Creative Collaboration: Simple Tools forInspired Teamwork, *Thomson Crisp Learning*, ISBN:1560526874, 2003b

Janssen, J., Tattersall, T., Van den Berg, B., & Koper, R., Navigational support in lifelong learning: enhancing effectiveness through indirect social navigation. *Proceedings of International Workshop in Learning Networks for Lifelong Competence Development, TENCompetence Conference*, 2006

Janssen, J., Tattersall, C., Waterink, W.,Van den Berg, B., Van Es, Bolman, C., & Koper, R., Self-organising navigational support in lifelong learning: how predecessors can lead the way. *Computers & Education*, 49(3), 2007

Johansen, Robert, Charles, Jeff, Mittman, Robert, & Saffo, Paul. Groupware: Computer Support for Business Teams (Series in Communication Technology and Society) ISBN**:** 0029164915, 1988

Jun-Zhao., S., & Sauvola., J., Towards a Conceptual Model for Context-Aware Adaptive Services, *The 4th International Conference on Parallel and Distributed Computing, Applications and Technologies PDCAT'2003*, pp. 90-94, 2003.

Kang K., S., Cohen, J. Hess, W. Novak, & A. Peterson, Feature-Oriented Domain Analysis (FODA), *Feasibility Study, Software Engineering Institute, Carnegie Mellon University, Technical Report CMU/SEI-90-TR-21 ESD-90-TR-222*, 1990.

Karin., A., & Hummel., M., Modelling Context-Aware Learning Scenarios, *Proceedings of the 3rd Int. Conf. on Pervasive Computing and Communications Workshops (PerCom 2005 Workshops)*, 2005.

Kramer., J., & Magee., J., Self-managed systems: an architectural challenge. *FOSE '07: 2007 Future of Software Engineering*, pp. 259-268, 2007.

Lee., J., & Kang., J., A Feature-Oriented Approach to Developing Dynamically Reconfigurable Products in Product Line Engineering, *Proceedings of the 10th International on Software Product Line Conference,* pp. 131 – 140, 2008.

Malzahn, N., Zeini, S., & Harrer, A., Ontology Facilitated Community Navigation- Who Is Interesting For What I am Interested In?, *Lecture notes In Computer Sciences*, 3554(2005), pp. 292-303, 2005.

Mayo, M., & Mitrovic, A., Optimising ITS behaviour with Bayesian networks and decision theory, *IJAIED*, 12(2), pp. 124-153, 2001.

Mitchell S., M. Spiteri, J. Bates, & G. Coulouris, Context aware multimedia computing in the intelligent hospital, *Proceedings of the Ninth ACM SIGOPS European Workshop, Denmark*, 2000

Munoz M, Rodriguez M., Favela J., Martinez-Garcia A., & Gonzalez V., Context-aware mobile communication in hospitals, IEEE Comput. 36 (9) (2003) 38–46, 2003.

Ohlson., S., Learning from performance errors. *Psychological Review*, 103(2), pp. 241-262, 1996

Pascoe, J., Adding Generic Contextual Capabilities to Wearable Computers. *2nd International Symposium on Wearable Computers*, pp. 92-99, 1998

Rowston, A., & Druschel., P., Pastry: Scalable, Decentralized Object Location, and Routing for Large-scale Peer-to-peer Systems, *Proceedings of the International Middleware Conference*, pp. 329-350, 2001

Robinson., W., N., Monitoring web service requirements, *Proceedings of International Requirements Engineering Conference (RE03)*, pp. 65-74, 2003

Rehman., K., Stajano., F., & Coulouris., G., An architecture for interactive Context-Aware Applications, *IEEE Pervasive Computing*, 6(1), pp.73-80, 2007

Schilit, B., Adams, N. & Want, R., Context-Aware Computing Applications. *1st International Workshop on Mobile Computing Systems and Applications*, pp. 85-90, 1994

Schneiderman, B., Fischer, G., Czerwinski, M., Resnick, M., Myers, B., Candy, L., Edmonds, E., Eisenberg, M., Giaccardi, E., Hewett, T., Jennings, P., Kules, B., Nakakoji, K., Nunamaker, J., Pausch, R., Selker, T., Sylvan, E., & Terry, M., *Creativity Support Tools: Report from a U.S. National Science Foundation Sponsored Workshop*, International Journal of Human- Computer Interaction, 20(2). Pp. 61-77, 2006

Schmid K., & I. John, A Customizable Approach To Full-Life Cycle Variability Management, *Science of Computer Programming*, 53, pp. 259-284, 2004

Schmidt., A., & Winterhalter. C., User Context Aware Delivery of E-Learning Material: Approach and Architecture. *Journal of Universal Computer Science*, pp. 28-36, 2004

Stanford V., (2003), Beam me up, Dr. McCoy, IEEE Pervasive Comput. Mag. 2 (3), pp. 13–18, 2003

Strang T., & Linnhoff-Popien C., , A context modelling survey, *First International Workshop on Advanced Context Modeling, Reasoning And Management (UbiComp '04)*, 2004

Tarasewich., P., Designing Mobile Commerce Applications. *Communications of the ACM*, 46(12), pp. 57-60, 2003

Tedesco R., Dolog P, Nejdl, W., & Heidrun, A., (2006), Distributed Bayesian Networks for User Modeling. *ELearn'2006: World Conference on E-Learning in Corporate, Government, Health Care, and Higher Education,* 2006

Uruchrutu E., MacKinnon L., & Rist R., User Cognitive Style and Interface Design for Personal, Adaptive Learning. What to Model? , Book Chapter of Lecture Notes in Computer Science,ISBN: 978-3-540-27885-6, p. 154-163, 2005

Van Rosmalen, P., Adaptation in AlFanet, *Presentation at the 4th Review Meeting aLFanet, April 29th, 2005 in Luxemburg, IST 2001-33288,* 2005

Wang, Y-K , Context awareness and adaptation in mobile learning, *Proceedings Second IEEE International Workshop on Wireless and Mobile Technologies in Education* (WMTE'04), pp. 154, 2004

Weber., G., & Brusilovsky., P., ELM-ART: An Adaptive Versatile System for Web-based Instruction. *International Journal of Artificial Intelligence in Education,* 12, 351-384, 2001

Wenger, E., (1987), Artificial intelligence and tutoring systems. *Computational approaches to the communication of knowledge,* pp.486

Yanlin, Z., & Yoneo, Y., A Framework of Context Awareness support for peer recommendation in the e-learning context, *British Journal of Educational Technology,* 38(2), pp.197-210, 2007