INSEMTIVES
FP7-ICT-2007-3
Contract no.: 231181
www.insemtives.eu

INSEMTIVES

Deliverable 1.2.1

Methodology for Semantic Content Authoring (Initial Version)

Editor:	Katharina Siorpaes, University of Innsbruck
Deliverable nature:	R
Dissemination level: (Confidentiality)	PU
Contractual delivery date:	31.03.2010
Actual delivery date:	31.03.2010
Version:	1.0
Total number of pages:	31
Keywords:	Semantic content authoring, methodology, game theory, implementation and introduction methods, participatory design, end user development, action research, organization and technology development, laboratory experiments, field studies, evaluation methods

Abstract

This document presents an initial version of a methodological framework for semantic content authoring by end users. We first summarize semantic content authoring tasks including typical instances and tasks in which a human participation is required. The framework integrates methods for (requirements) analysis, design and introduction of technological tools, adaptation and appropriation in the social/organizational practice and evaluation, keeping as central the problem of user-involvement.

The deliverable focuses on a set of intertwined decision processes and their implications for organizational practice. The main dynamic relationships between those processes are identified, theoretical approaches for motivation support are introduced and an appropriate methodology for processes of participatory design and integrated organization and technology development are presented.

The presented framework allows for a) an analysis of work environments of a social nature in which one aims to introduce annotating tasks and b) the conduction of a design process for the tools that serve as vehicle for the tasks themselves taking the previous analysis of the social and technological context as an input. These two phases are mirrored in the two deliverables as follows: the deliverable D 1.2.1 contains all of the elements that can guide the design process *strictly speaking*, whereas the present deliverable D1.3.1 concentrates on the analytical steps that logically precede the design process. This approach stresses a *value-chain outlook* on the design process, clearly distinguishing problems regarding motivation/participation in the design process *strictu sensu*, on the one hand, and motivation/participation of users once the tool is in place, on the other hand.

Naturally, we expect the reader to proceed iteratively between the two layers that, as far as we are concerned, can only be analytically separated but that are carried out simultaneously in the design practice. Whereas the

deliverable D.1.2.1 provides a comprehensive methodological approach allowing for the analysis, design and testing of the tools which will be used in the case studies for semantic content creation tasks, the deliverable D1.3.1 outlines a general framework which aims at guiding designers with respect to which design goals or requirements to be considered as well as allowing them to map the analysis results into a possible set of solutions.

[Full project title] INSEMTIVES – Incentives for Semantics

[Short project title] INSEMTIVES

[Number and title of work-package] WP1: Incentives Framework for Semantic Content Creation

[Document title] Methodology for Semantic Content Authoring (Initial Version)

[Editor: Name, company] Katharina Siorpaes, UIBK

[Work-package leader: Name, company] Katharina Siorpaes, UIBK

Executive Summary

The success of the INSEMTIVES is highly dependent on motivational concepts, which encourage users to get involved in the process of annotating a huge amount of unstructured information and generating valid ontologies for non-textual sources. Following Herbert Simon we can distinguish between two types of motivation: (a) the motivation to participate, that is the willingness of a subject to take part in a task, and (b) the motivation to produce, that is the intention to perform competently once he commits to the task. These types of motivations and incentives aspects are not only investigated within sociological and economical research with the aim to develop theories and experimental insights, but also within the more IT and design oriented research, including those dealing with the design of environments for online communities or cooperative works.

The insights from previous IT and design research suggest that enabling participation and motivating users to contribute in online contexts require the design of both technical functionalities and social mechanism as features of a system. In other words, the system needs to be first of all usable, i.e. it should allow users to work efficiently, effectively and with satisfaction. In addition, the system should support sociability, i.e. it should invite and support its users to engage in social activities online as well as offline. Yet, the success of the users involvement in the process of content creation is not only dependent on the features of the tools (or design artefacts) to be used, but also the design process that created such tools can have also a significant effect on the acceptance of such tools by the users.

Therefore, we first summarize semantic content creation tasks and describe use case scenarios. Then, this deliverable describes methods for enabling user participation in design process and for evaluation of designed artefacts with respect to the goals and requirements for motivating users to participate.

Whereas this deliverable provides a comprehensive methodological approach allowing for the analysis, design and testing of the tools, the deliverable D1.3.1 outlines a general framework which aims at guiding designers with respect to which design goals or requirements to be considered as well as allowing them to map the analysis results into a possible set of solutions. We expect the reader to proceed iteratively between the two layers that, as far as we are concerned, can only be analytically separated but that are carried out simultaneously in the design practice.

List of Authors

Company	Author
University of Trento	Marco Zamarian
University of Siegen	Fahri Yetim
University of Siegen	Markus Rohde
University of Siegen	Torben Wiedenhöfer
University of Trento	Oksana Tokarchuk
University of Trento	Roberta Cuel
University of Innsbruck	Katharina Siorpaes

Table of Contents

1 Introduction

According to the Description of Work of the project D1.2.1 should focus on methodologies for semantic content authoring, such as a detailed description of the process model (activities, participants and roles, decision to be made, best practices and guidelines) and a set of participatory design and integrated organization and technology development methods. D1.3.1 should focus on an initial set of incentive models and on guidelines for tool design, usability engineering and community support to foster the motivation of users to engage and participate in semantic content creation.

In other words, the task of D1.2.1 and D1.3.1 is that of providing a guide to incorporating into software artefacts tools and solutions developed in the social sciences with the aim of improving the level of effort exhibited by users in both ontology creation and population tasks, mainly by leveraging on different motivation processes. This ambitious overall goal presents many challenges due to both the complexity of the task itself and to the heterogeneity of contributions that many disciplines can bring to this particular application.

This general introduction will serve the purpose of presenting in an abridged fashion these challenges and will also present the rationale for organizing the materials between the two deliverables in their present arrangement. In this document, we first provide an overview of semantic content creation tasks. Then, we describe those tasks that require substantial human contribution. Subsequently, we present and analyze the processes by which we can contribute to the design of those tasks, taking their special features in terms of human motivation and commitment into account.

Generally speaking, our goal is to identify a set of relationships between different analytical kinds of decision processes, located at different levels that can explain how software artefacts, and their design, are related to the social processes relevant to users, with a focus on their motivation. We want to emphasize that a special focus on decision processes does not necessarily imply an over-rationalistic approach. On the contrary, our framework is based on the concept of bounded rationality [70], [47]. Our concept of decision process implies all those informational, social and cognitive limitations and biases that are included in the concept of bounded rationality.

When the interplay between artefacts and organization is considered, three different sets of homogeneous relevant decision processes need to be distinguished: design, adoption and use decisions. In this paragraph, we will first describe individually these sets of decision processes, and then we will describe how these processes dynamically interact.

Design decisions: Most artefacts have to be conceived (either implicitly or explicitly), designed and concretely produced. At this stage, several decisions are involved, concerning, for example, the general goals related to the use of the artefacts, the operational functions, and the interfacing modalities between artefacts and potential users; in other words, these decisions concern all the physical, technical and operational features of the artefact

Design decisions can take place both internally and externally to the work process in which the artefact will be used. In both cases, design decisions have always some influence on the activities. Sometimes this influence can be very relevant – for example, when design decisions put significant constraints on users' actions.

Adoption decisions: artefacts have to be connected with the actual work activities and operations. Only artefacts that have been integrated within the work processes can be considered as organizationally relevant. Thus, a number of adoption or implementation decisions become crucial for the change potential that artefacts bring into the organization: e.g., decisions concerning in which macro-area of activities the artefact will be used, in which work processes, in which specific operations, by which operators, in which ways it will be connected with other activities or other artefacts, and so forth.

Use decisions: Use decisions concern the way end-users interact with the artefact in the actual work processes. For example, users of software systems normally have different choice options about what kind of operation to perform, with what kind of procedure, what kind of information to input in the system; sometimes they can customize the application itself.

Thus, use decisions are organizationally relevant not only for the obvious reason that they concern the actual actions within the organization, but also because they are not always and not necessarily aligned with designers' and adopters' expectations. In many cases, users take use decisions that can be rather divergent from design and adoption decisions, although they are necessarily and at least partially influenced by them. Indeed, we believe that it is very important to consider the alignment of users' appropriations to adoption and design decisions: the relevance of artefacts as structuring devices becomes more interesting, in organizational terms, when the three decision processes (design, adoption, use) become carriers of different – and sometime conflicting - goals and expectations about the meaning of artefacts.

As we stated before, our approach focus on a set of intertwined decision processes and their implications for organizational practice. Thus, even though design decisions are usually taken by external and / or internal engineers, adoption decisions are usually taken by managers and use decisions are usually taken by workers and operators, every actor can make design choices (if and when he influences the technical features of the artefact), adoption choices (if and when he influences the way the artefact is integrated with the work processes) and use choices (if and when he actually uses the artefact), either in different moments or even at the same time. Our framework identifies the main dynamic relationships between those choice processes, and it can be graphically summarized as in Figure 1. In the rest of this section, the letters within parenthesis refer to the related elements of figure 1.

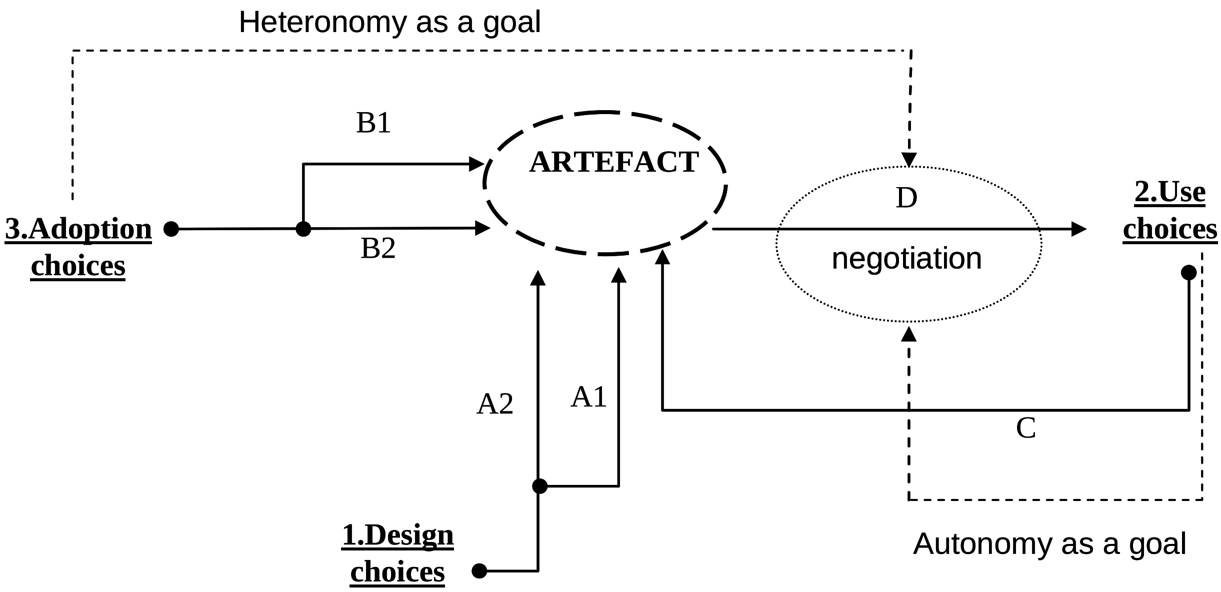

Figure 1: The analytical framework

1. **Design choices**. Designers conceive the artefact according to their technical knowledge, their goals, their customers' needs and requests. Design choices define (A1) what we call the artefact's core, consisting of those elements of the artefact that are not directly connected to the users' choices. Design decisions also define (A2) what we call the artefact's interface, consisting of those elements that are directly connected to the use choices. When considered jointly with other decision processes, both types of design decisions are very relevant in organizational terms. First, they both are the outcome of the designers' interpretations of the organizational setting in which the artefact will be used. It is important to explicitly consider this in order to avoid technological determinism: technology does not "impact" on work processes from "outside", since it is the outcome of specific (bounded rational) choices. Second, the design choices that define the artefact's interface have a big influence on actual users' behaviour. Artefact's interface represents the "means" through which users can appropriate the artefact; in other words, it gives users constraints and opportunities about their own work processes.
2. **Use choices.** Users appropriate the artefact in work processes through use decisions (C). More specifically, use decisions enact what we call the use-in-use, that is, the users' interpretations and the cognitive filters defining "what is" the artefact for the users. In every work process it is possible to distinguish between rules and procedures that are generated by actors that are directly involved in the same work process (autonomous rules), and rules that are given or imposed by actors that are external from the work process itself (heteronomous rules) [63], [26]. Use decisions depend on

subjective knowledge, goals, and values of individual users. Thus, the use-in-use depends directly from use decisions, and indirectly from design decisions (crystallized in the artefact's core and interface). However, the use-in-use depends also on adoption decisions.

3. **Adoption choices.** Adoption decisions influence the use-in-use by demanding users to enact certain specific use modalities and interpretations (B2), and by influencing the artefact's interface and core (B1). For example, managers often try to encourage certain skills and abilities while discouraging others, they require the users to have certain types of knowledge, they require the users to input some kinds of information and not other kinds, and so forth. They can also influence directly the artefact's interface by not allowing users (either formally or informally) to access to certain parts or elements of the artefact's interface. Thus, the general goal that is implicit in adoption decisions is that of introducing heteronymous rules in the structuring process.

These three choice processes can diverge in significant ways, and this divergence represents a very important point. Indeed, different forces interact in a negotiation process (D), that can be either implicit or explicit, resulting in the actual, local and transitory structural arrangements [63], [26], [49] of work processes.

As a whole, our framework emphasizes the complex mediation role of artefacts between different decision processes, taking place at different analytical levels, with different time frames, by different kinds of actors and how they can be shaped by, and interfere with, their belief systems. Defining mechanisms such as motivation and implicit and explicit incentives are clearly a central part of this goal.

As a methodological framework we introduce the approach of Integrated Organization and Technology Development (OTD), which is based on an action research perspective and offers a process model for that analysis, design, adaptation/appropriation and evaluation of socio-technical systems.

As we have just seen, we are not examining merely a design process allowing for the incorporation of some received wisdom into technology; rather we have the task of proposing a methodology that encompasses the ability to a) analyze any work environment of a social nature in which one aims to introduce annotating tasks; b) conduct a design process for the tools that serve as vehicle for the tasks themselves taking the previous analysis of the social and technological context as an input.

These two phases are mirrored in the two deliverables as follows: the present deliverable D 1.2.1 contains all of the elements that can guide the design process "strictly speaking", whereas the sister deliverable D1.3.1 concentrates on the analytical steps that logically precede the design process.

This approach stresses a "value-chain" outlook on the design process, clearly distinguishing problems regarding motivation/participation in the design process *strictu sensu*, on the one hand, and motivation/participation of users once the tool is in place. The most glaring advantage of this approach is the ability to consolidate theory, method and applications in distinct units, avoiding redundancy.

The rest of the document is organized as follows. Section 2 presents the analytical dimensions preceding the core design process. Section 3 presents the core design process and its sub phases; section 4 details the diagnostic tools associated with the design process allowing for an evaluation of the results attained. Section 5 closes with some conclusions and a link to D1.3.1.

2 Analysis

In this section we present a set of analytical considerations that should be made at a design stage with regards to human involvement into semantic content creation. We start with the analysis of the need of human involvement in a semantic content authoring. We describe the semantic content creation tasks and the type of contribution humans should provide. Next, we consider how to promote human contribution in these tasks. We present game theory – an analytical tool used by economists to study incentives in different settings. While D.1.3.1. focuses mainly on an initial set of incentive models and guidelines to foster the users' motivation to engage and participate in semantic content creation, in section 2.2.1 we explain what is game theory. We present how this tool can be used to analyze and design incentive system for distributed production settings (the typical setting of semantic content creation tasks). Analysis provided by the tools of game theory is normative and is done under some theoretical assumptions. There is a need to test recommendations obtained by game theory. In section 2.2.2. we present laboratory and field experiments that are used testing game theoretical recommendations. This testing is an integral part of the design process with the tools of game theory as during experiments a whole set of details that are not included in the general game theory concepts are collected.

2.1 Semantic content creation: human contribution and use cases

In this section we provide an analysis of the most important semantic content creation activities which maily refers to two different activities:

- To develop ontologies and other semantic based conceptualizations (e.g. partonomies, taxonomies, classifications), to modify, evaluate, align and evolve them
- To populate ontologies adding instances or tagging various resources.

In both these activities, human contribution is strongly required.

Following, we first provide an overview of of semantic content creation tasks and we describe those that require substantial human contribution.

Ontology designates an explicit specification of a shared conceptualization that holds in a particular context [28], [29]. In other words, ontology provides an explicit conceptualization, which describes semantics of data, providing a shared and common understanding of a domain. Creating and developing ontologies requires domain expertise and the ability to capture this knowledge in a clean conceptual model. Various tools have been developed to help people in creating manually, or semi-automatically, categories, partonomies, taxonomies, and other organization levels of ontologies. Behind these tools and techniques, different (domain-independent) approaches and methods are used to develop numerous heterogeneous ontologies [11].Some methodologies are designed for a team-oriented approach, i.e., a team of ontology engineers and domain experts produce an ontology. However, as presented in [28], 7 out of 17 tools need user intervention throughout the entire process, 8 can be run semi-automatically, and only 2 are fully automated. As such, and despite existing semi-automated approaches of ontology creation, alignment, and evolution, it remains a costly and time-consuming human-driven process, whereas other approaches suggest that ontology building is performed within a community of contributors in a collaborative fashion. This topic is also well described in "D2.1.1 Report on single-user and community-based annotations - state of the art, requirements and specification" provided by the project members for month 6.

The alignment of heterogeneous ontologies is regarded as one of the major challenges for making the Semantic Web a reality and semantic technologies a success [16], [20], [19], [57]. There is a wide range of algorithms that aim at semi- or complete automation of the mapping tasks, but ontology matching cannot yet be done fully automatically [19], [21]. Many approaches attempt to develop automatic ontology matching tasks, but only very few can run automatically. Most of the ontology alignment tools require the humans' involvement to either provide training data, define rules to carry on mappings, or give feedback on the suggestion of the system.

Ontology evaluation aims at providing a critical technical judgment on the quality of the ontology. It is a very broad topic, and so far, no fully automatic approaches have emerged and semi-automatic approaches are rare [8]. In its nature it is still human-driven, as it has to evaluate what a human user initially provided in the conceptual modeling phase. As explained in [71], [37], only few methods, which support automatic or semi-automatic evaluation, are available (i.e. OntoClean, EvaLexon).

Concluding, ontology building, alignment and evaluation are mainly human-driven tasks. Even if automation support is partly possible for collecting relevant terms for ontologies or proposing properties to concepts derived from semi-structured knowledge corpora (folksonomies), the final modeling decision is mainly taken by human actors.

The population of ontologies is a task within the semantic content creation process as it links abstract and concrete knowledge. It usually refers to the association of instances (resources) to ontology concepts or more easily to the annotation or tagging of resources. The tags might refer to concepts of ontologies, partonomies, taxonomies, or simple classifications. This knowledge acquisition can be carried on manually, semi-automatically, or fully automatically.

There is a wide range of approaches that propose semi-automatic annotation of text: most of the approaches make use of natural language processing and information extraction technologies. These already have a long tradition and reached a good level of maturity. Even though they require training, a large share of the work can be automated [62], [75]. The nature of semantic annotation of text is not easily defined because it requires training but can then continue autonomously. Due to time-consuming training activities (which is required in the first place or when domains continuously evolve), semantic annotation requires substantial human contributions [62], [75]. Summing up, semantic annotation of text is advanced and allows for automation in many cases but requires a substantial amount of training by human users.

The situation is slightly different with the annotation of multimedia content. A large share of approaches aims at extraction of low-level semantics (such as simple annotation or tags that refer only to taxonomies or cloud keywords). However, the real challenge is the provision of high-level semantics, i.e., semantic content descriptions. This can only be done to a limited extent, e.g., by applying machine learning with a vertical focus for a specific domain. Approaches for the annotation of media objects, be that manual, semi-automatic or automatic ones, aim at closing the so-called “semantic gap”, i.e., the discrepancy between low-level technical features, and the high-level meaning-bearing features a user is typically interested in. More precisely semantic gap means the difference between, the reproducible and computational representation of knowledge in a formal language (e.g. programming language) and the ambiguous formulation of contextual knowledge in a powerful language (e.g. natural language). For practical application, any formal representation of real world tasks requires the translation of the contextual expert knowledge of an application (high-level) into the elementary and reproducible operations of a computing machine (low-level). Since natural language allows the expression of tasks which are impossible to compute in a formal language and existing approaches aim at a high degree of automation, only a limited number of specific domains and types of media are automatically translated [71]. In the following table, we describe some of the semantic content creation tasks and the type of human contribution is required to fulfill them. In some relevant cases, we describe the concrete case study in which such task can be applied. A More in depth analysis of case studies will be reported in the deliverables: D5.2.1, D6.2.1, and D7.2.1 planned for M20.

Table 1: Semantic content creation tasks

Name	**Description and Objective**	**Human Contribution**	**Use Case Summary**
Ontology Development	The activity of ontology development aims at producing conceptual models. The activity includes various sub-steps (see below). The outcome is a model of varying expressivity (heavy- vs. lightweight ontologies) depicting some domain of interest. Depending on the setting the activity can be carried out individually, in an organized or in an open community.	Largely human-driven. Only the collection of very standardized terms and definition of class hierarchies can be automated.	The ontology is a shared conceptualization and it usually refers to a specific domain. In that case, domain experts and knowledge engineers can build and maintain the information system and the respective ontology (e.g. Telefónica OKenterprise). The ontology is built by an open community (e.g. on the Web) as a collaborative effort (cf. Wikipedia or myOntology).

The ontology development task can be divided in many subtasks which might reflect one or more ontology creation methodologies. In this deliverable we have adopted the very famous methodology developed by Noy and McGuinnes called “Ontology Development 101” available at http://protege.stanford.edu/. In this methodology the following tasks have been identified: determine the domain and the scope of the ontology making use of competency questions, consider reusing existing ontology, enumerate important terms in the ontology, define the classes and the class hierarchy, define the properties of classes—slots, define the facets of the slots, create instances. These tasks are described in the table 2.

Table 2: Semantic content creation tasks: Ontology Development 101 steps

Name	Description and objective	Human contribution
Description of Domain and Scope	This task results in a thorough description of what the ontology is expected to cover. It outlines the domain, scope and aim of the ontology and lists the intended uses and users.	The domain and scope (requirements, motivating scenarios) of the ontology must be defined by one or more human actors.
Competency Questions	This task can be seen as a continuation of the previous one: competency questions are a means to further specify the domain and scope of the ontology and to evaluate the ontology in a later stage. They capture the queries that the ontology is expected to answer based on the identified usage scenarios.	The formulation of informal or formal competency questions is complex and requires a deep understanding of the whole project, which can be hardly done automatically.
Re-use of Existing Ontologies	Based on the state of the art (D1.1.1), automation support for the re-use of existing ontologies seems feasible when searching potential re-use candidates.	Discovery of suitable ontologies can be performed automatically. The selection of re-usable ontologies and the actual re-use rely on human input.
Collection of Relevant Terms	From a corpus of knowledge (documents, interviews, etc.) relevant terms for the ontology are extracted. The outcome is a list of terms the ontology should cover.	Given an appropriate corpus that can be used for term extraction, automation is possible. The final selection and check of relevant terms must be done by a human actor.
Typing of Terms	This task answers the question whether a term should be modeled as a concept, an individual, a relationship, or an attribute.	Deciding whether a term falls in a particular category can be supported automatically to a very limited extent. It is largely human-driven.
Building a Hierarchy	The goal of this task is to define the class hierarchy (is-a relationships).	Building a hierarchy requires human input at least for validation purposes.
Define Properties	This task includes the definition of cross-taxonomical relations, as well as attributes further describing a concept.	The definition of properties can be supported by automatic methods, however to a less extent than the definition of the class hierarchy.
Define Axioms	Defining axioms involves specifying precise logics-based rules, such as cardinality constraints, disjointness, etc. that apply to concepts.	Extending an ontology with axioms requires a human author.
Create Ontological Instances	The creation of instances is also referred to as semantic annotation or ontology population: see below.	The distinction between classes and instances is very specific and requires the human in the loop.
Documentation	The documentation of an ontology is an essential component of ontology engineering to facilitate maintenance and re-use, and to ensure a smooth operation of a collaborative process.	Ontology documentation is heavily human-driven, as it refers to decisions made during the engineering process by the participants.
Ontology Alignment	This activity aims at aligning different ontologies in order to enable data integration.	Tools are all designed for automation, but the majority required human contribution.
Ontology Learning	This activity aims at providing semi-automatic support when creating, extending, and maintaining ontologies.	Learning tools are automatic or semi-automatic (requiring human intervention). Activities around learning are human-driven.

Ontology Evaluation	This activity produces an assessment of an ontology and validates its compliance with the requirements initially specified.	Some aspects can be evaluated automatically. However, whether an ontology is suitable to be used in a domain can to a large extent only be evaluated by a human actor.
Annotation of Text	This activity involves the population of an ontology with textual resources, or – in other words – attaching ontological elements to text (portions). With more lightweight ontologies, the annotation can also be done via simple tagging.	Semi-automatic with human contribution. User creates text and validates annotations done by system. User searches for documents/text and provides annotations to the found resources. User annotates existing content.
Annotation of Multimedia	This activity aims at describing multimedia content (image, sound, video) with ontology concepts and relationships. Similarly to text annotation, tagging based on ontology concepts is a way to facilitate this with lightweight domain models.	Largely human-driven with little automation. User creates and uploads multimedia and annotates it. User finds existing content and annotates it. User changes annotation of a file. User searches content and annotates found files.

As we can see from the table, there are many types of human contribution which are required to fulfill semantic content creation tasks. Some of these tasks are carried on in the semantic content creation processes of the three case studies, and will be analyzed according the task matrix described in D1.3.1.

2.2 Economists' Tools To Analyze and Design Incentive Systems for Distributed Production Settings

In this subsection we present methodology that economists use to address problems in which incentives are involved. Incentives are any factor (financial or non-financial) that enables or motivates a particular course of action, or counts as a reason for preferring one choice to the alternatives. Ontology building and semantic annotation requires significant human contribution. The best technology is worthless if no one uses it. Thus, there is a need to motivate human users to contribute to the process of semantic content creation. Game theory is a standard economic tool for the analysis of situations from the point of view of incentives. We briefly explain what game theory is and how it can be used for design of incentive systems for semantic content authoring in section 2.2.1.

The result of the design process with the use of game theory is a simplified representation of a real life situation. It includes a set of recommendations about the rules and actions for participants. Game theory is a useful tool for the analysis of interaction among individuals. However, game theory is a mathematical tool and provides theoretical predictions. It puts strong requirements on abilities of people to predict actions of their game partners. Not always theoretical recommendations suggested by the game theory framework obtain good results in their application to reality. It is necessary to test the recommendations to understand if they suite the concrete real case.

In economics the test of game theory is performed with experiments. A group of people is asked to perform a task following experimental rules. Experimentalist observes actions and behavior of experimental subjects and compares outcomes of the experiment with expected outcome predicted by the game theoretic solution concept. Experiments provide an important insight into human behavior in a representative situation that often can be extended to a more generic case. In section 2.2.2 we present two types of experiments that economists use to test game theoretical predictions, laboratory and field experiments.

2.2.1 Game Theory

Game theory is a widely used tool in economics to study behavior of individuals when more than one individual is involved and there is strategic interdependency. In these situations individual recognizes that payoffs that are received depend not only on the single actions of the individual but also on the actions taken by other individuals involved in the situation. Therefore, the best action to be undertaken may depend on the actions already taken by other individuals, actions that individual expects that others may take or actions that may be taken by other individuals involved in the context.

From the point of view of game theory semantic content creation is classified as a task in which multiple individuals are involved and that is characterized by strategic interdependency. The success of the creation of semantic content largely depends not only on the quality and quantity of a single contribution of a single user. A vital importance in this task belongs to the collective effort of each single user. Although technical aspects are significant for the realization of the project of content creation it is crucial to provide correct incentives to the individuals involved in the task.

“A game is a formal representation of a situation in which a number of individuals interact in setting of strategic interdependency” as it is defined by [48]. The following features define situation of strategic interdependency:

- The players: who is involved in the situation
- The rules: who moves when? What do they know when they move? What they can do?
- The outcomes: for each possible set of moves what is the outcome of the game?
- Payoffs: what are the players’ preferences over the outcomes of the game?

For example, let’s consider a simple game of meeting at New York:

Players: Tow players, Mr. Thomas and Mr. Schelling

The rules: The two players are separated and cannot communicate. They are supposed to meet in New York at lunch but forgot to specify the place. Each should decide where to go (each can make only one choice).

Outcome: If they meet each other, they will enjoy each other’s company at lunch, if not they must eat alone

Payoffs: They each attach a monetary value of 100 dollars to the other’s company at lunch (their payoffs are 100 dollars if they meet, 0 otherwise).

In this example, players’ interests are completely aligned. They simply need to coordinate their actions to obtain the preferred outcome, meeting each other for lunch. Nevertheless, each player’s payoff depends on what the other player does. Game theory studies a whole set of situations that involve strategic interdependency, games that range from pure conflict to cooperation.

In game theory, the solution of the game, the solution concept, is a formal rule for predicting how the game will be played. These predictions describe which strategies players will adopt, therefore predicting the result of the game.

By the design of incentive system we understand development of a game by playing which players reach the desired by the designer outcome as equilibrium of the game. To develop such a game we need to define players, rules, outcomes and payoffs of this game.

Game theory provides a simplified representation of the problem that permits to indentify main agents involved in the situation, rules they are following and their motivation to complete the task. This approach permits to classify seemingly unrelated situations into general classes of problems and to provide recommendations in each single situation. Correct recognition of the concrete situation in game theoretic terms permits to design incentive structure that makes all players do their best to complete the task.

Therefore, designer needs to analyze the situation at hand. It is important to define which the parameters of the process are. Very often the set of players and rules of the game are defined by the project from the very beginning and it is impossible to intervene by changing these components of the game. In this case it is crucial to obtain correct description of the existing situation and its classification.

Frequently, the design of the incentive system does not start from zero but there is already a platform that is functioning. Also in this case the first step to take is to analytically describe the game that is involved, define the players, their set of actions and payoffs that they assign to possible outcomes. Departing from this description of the game the action is made to achieve desirable game solution.

Analysis of the pre-existing situation can be done with the tools that are discussed in section 3.

2.2.2 Laboratory and Field Experiments

Design of incentive system in game theoretic terms provides a theoretical concept of the game that represents the analyzed situation in general and the solution to it. Application of this concept in real life requires providing a whole set of details that are not included in the general concept that can be studied with tools of game theory. Dealing with this creates complications not foreseen by standard tools of game theory. To solve these complications several tools are available to the designer.

Once the incentive structure is developed a good practice suggests testing it with real people before its implementation on a large scale. Two approaches are generally used to address this task. Both of them refer to experimental testing of the incentive design.

The first approach permits testing incentives' structure in a controlled environment of laboratory setting. In this case experimental subjects are invited to the laboratory. They are presented with a simplified representation of the problem, i.e., they are asked to play a game that is characterized by the rules, outcomes and payoffs as defined by the designer. At this stage it is possible to discover that, when faced with a game, subjects may behave in a different way compared to what was expected by the designer. There can be some problems with it:

- The game was defined in a wrong way. In this case the designer needs to step back and, based on the observed behavior, to re-define the game that is being studied
- The game was defined correctly but the outcome of the interaction of experimental subjects is different from the solution concept. There are several situations in which behavior of real people leads to a different solution concept compared to the one provided by the theoretical predictions of the model. It is important to study behavior of subjects in similar experimental settings at the design stage.

Experimental subjects can be every kind of individuals that are considered suitable for the completion of the experimental task. Normally, these subjects are university students, but they can also be real users of the system that are involved in the task of content creation under analysis.

The task that subjects are asked to perform during the experiment can be the real task for which the model is designed or it can be a simplified representation of this task. Using the real task is more desirable for the experiment. If the incentive model is developed for a platform aimed at general public that is not expected to have prior knowledge for contributing to the task, it is recommended to use the real task in the experiment. In this case the experiment not only tests the incentive model but also provides a test of the portal in general.

In some cases the aim of the experiment does not allow the use of the real task. For example, if the aim of the experiment on meeting in New York game discussed earlier in the section is to study the role of communication and the extent of interpersonal knowledge in solving coordination problems, it is better to avoid making people actually meet in New York. Instead, experimental subjects, that do not know each other, meet in the laboratory. Their task is to choose one of the proposed meeting places in a way to match the choice of a randomly assigned to them partner. Experimental conditions control for communication and interpersonal knowledge. In similar cases the experiment allows testing subjects' performance under proposed incentive system. However, this experiment cannot provide a reliable test of the portal as subjects may not possess necessary knowledge for using it.

It is advisable that experimental subjects are rewarded for their participation in the experiment [9], [39]. This reward should be based on their performance during the experiment in accordance with the developed incentive scheme. The easiest form of reward is material benefit that can be translated in cash or other similar payments. It is possible that the main component of the incentive scheme is non-material. In this case it is important that the experiment matches the main features of this scheme.

Experiments in the laboratory are a relatively low-cost tool for the first testing of the system of incentives. In a controlled setting of a laboratory it is easy to fine-tune the rules of the game, the structure of strategic interaction among players, explore reaction to the change of the outcome structure or its magnitude.

Experiments in the laboratory are an important tool for the first test of the incentive system. They permit, if needed, to test the influence of a chosen set of parameters on the whole system in a controlled environment. However, laboratory experiments simplify real life situation too much. Experimental task may be artificial compared to the real task or it may not completely reflect the nature of the real task. It may be impossible to collect as many experimental subjects in a laboratory experiments as it is required in a real life setting. Field experiments help to address these problems.

A second approach to testing incentive design refers to field experiment. Field experiment is aimed to solve the problems present in laboratory experiments due to their artificial nature. It permits to test the incentive model in an environment that captures important characteristics of the real world. The extent to which the field experiment brings the test closer to a live setting can be classified based on Harrison and List (2004) [34]. This survey lists six factors that determine the field context of the experiment: the nature of the subject pool, the nature of the information that the subjects bring to the task, the nature of the commodity (not applicable in our case), the nature of the task, the nature of the stakes, and the environment in which the subjects operate. These factors are interconnected and can all be present in the field experiment.

A first step towards to the real environment is the test of the incentive model with subjects that will be final users of the platform, for whom the incentive structure was designed. These subjects are believed to possess more information and skills related to the task compared to volunteers in the experiment, especially if the platform is case specific as in the case of corporate portal.

When a new tool is developed and is being introduced to the existing environment, it is a natural tendency to involve the most active part of the population for its testing. An obvious advantage of doing so is that it is easier to motivate these individuals to participate in the test. However, what works with the most motivated part of the population may not work with the rest of it. Moreover, the fact that subjects are informed that they are taking part in an experiment may be an important factor and alter their behavior. It was demonstrated that awareness of being part of the research at a working place makes people feel more important and thereby makes them improve their performance [2].

Therefore, for the best outcome of the experiment it is advisable to randomly select experimental participants from the existing pool of future users. Experiment usually compares several treatments. It is important that during the experiment people assigned to different treatments do not communicate to each other. These requirements are satisfied when the test is performed with subjects that do not know each other and they are located in different places. This condition is met when the platform is developed for general public and prospective users of the platform do not communicate and do not know each other. In this case it is possible to randomly select subjects from the pool of existing or future users. It is harder to meet these requirements when the platform is developed for people that work together before the experiment takes place, like in the case of corporate portal. Individuals assigned to different treatments may communicate to each other and alter their behavior. This problem can be avoided by performing the test in laboratory setting but in this case subjects know that they participate in a test.

In the field experiment's testing of incentive model subjects may be faced with the real platform for which the model is developed and perform the real task. In this case the outcomes of the experiment may be due to testing of the platform with real users and testing of the incentive model. When the real task is used in the field experiment it is important to evaluate the results taking into account both sides of the test.

In the field experiment subjects are rewarded according to the incentive system that is being tested. When subjects are faced with real task in the experiment it is possible to test also non-material aspects of the incentive system. However, if the experiment is conducted in the laboratory and incentive system is based exclusively on non-material rewards, there is a need to present subjects with some tangible reward to motivate their performance in the laboratory. This additional reward may alter subjects' behavior.

Field experiment can be conducted in the laboratory or at the usual place where the final users normally perform the task. Conducting the experiment in the natural for the user environment increases external validity of experimental results. In this case it is possible to conduct the experiment without communicating to subjects that they are taking part in the study. When subjects are in their natural environment the experimenter normally cannot control their working conditions, information that is being used, the degree of communication among subjects and many other important aspects of external environment.

The test of the incentive model in the field is a complex task. To obtain more reliable results of the experiment it is desirable to intervene in all the mentioned factors. However, the closer experimental conditions reflect the real environment, the more complex is the interpretation of the results. Thus, the experimenter is called to balance the eagerness for reliable results with the clearance of their interpretation. This stage faces the experimenter with a whole list of important design choices; each of them may have a dramatic influence on the outcome.

To sum up, field experiment is hard to conduct, as it is difficult to disentangle whether the obtained result is given by the parameter that is being varied or if it is caused by a combination of some factors that the experimenter does not influence upon. However, it is an important tool that permits to individuate complications that may rise during implementation of the system to real life environment.

Field experiment may lead to different results compared to the results observed in preliminary laboratory experiments. This can happen for several reasons:

- different subjects may behave in a different way;
- environment can have an important effect;
- the fact that subjects know that their behavior is observed and that they know what the experimenter is looking at can alter their behavior in the laboratory setting compared to the field.

It is difficult, if not impossible, to study the effect of a single factor with the tools of only field experiment. The best strategy is a joint use of both laboratory and field experiments. Let's consider, for example, the effect of communication among subjects on the overall performance. It is believed that communication increases performance in considered here tasks. By conducting only field experiment in the natural environment it is impossible for the experimenter to control for the degree of communication among subjects. Laboratory experiments may help to decide whether this factor is important for the concrete task and how to make best use of it in the concrete case (i.e., the ideal size of the group, the kind of communication, etc.). The field experiment is useful to address whether the platform is able to create conditions to induce communication on the desired level and whether it has an effect on the overall performance.

3 Design and Implementation

While the previous sections deal with human contribution for semantic content creation (and tasks/objectives in ontology development) and with the development of supporting reinforcement mechanisms for (external) motivation, the third section focuses on methodologies to include users directly into processes of software development/tool design and the development of appropriate organizational structures for collaborative content creation activities. This inclusion of users in the development processes aims to increase the (internal) motivation of end users and to design needs-oriented tools for the tasks at hand. Since section 2.2 focuses on an economic perspective on motivation, section 3 refers to a user-centered and practice-oriented perspective, mainly based on psychological, organizational and socio-cultural approaches.

In the following we present methods that support user participation and can be employed for designing tools and implementing them in organizational contexts. There are different research paradigms that aim at enabling user participation in design process. The one paradigm is Participatory Design (PD), which focuses on user participation in design time. The second paradigm is End-Users Development (EUD), which aims at designing adaptable systems in order to enable users to participate in use time. Finally, the Integrated Organization and Technology Development (OTD) is concerned with organizational mechanisms (e.g., policies, incentives) or activities such as community development in order to promote user participation. Each of the paradigms has its associated methods and techniques. The methodology we suggest for INSEMTIVES integrates the ideas and methods from these paradigms. In the following we briefly describe each paradigm.

3.1 Methods for Participatory Design

If one is striving for technical tools for the support of user-driven semantic content creation, these tools should enable and encourage users to contribute to a semantic content authoring community. To ensure the necessary affordances of such tools, such applications should follow a user-centered approach and should integrate incentive models as reinforcement for participation. Besides game-theoretically derived external motivation/reinforcement mechanisms, we additionally focus on internal motivation of (potential) end users by involving them into our development processes. To develop software for the participation of end-users, we propose a **participatory way** of designing these software tools, integrating potential users by participatory design methods. According to Horst Rittel:

"The experience of having participated in a problem makes a difference to those who are affected by the solution. People are more likely to like a solution if they have been involved in its generation, even though it might not make sense otherwise." (cited from [23], pp. 42)

The Participatory Design (PD) approach was developed for the improvement of the participation of workers in software development processes and the cooperation between software developers and end-users ([4], [5], [42]). To support the dialogue and collaboration between designers/system developers and end-users, PD researchers developed methods which allow users to participate in IT development projects as experts of their own work processes. PD approaches combine design-by-doing methods, scenarios and different forms of prototyping (such as mockups, rapid prototypes), work organization games and ethnographic methods (e.g. [30], [15]). PD can lead to perceived legitimation of design decisions and a higher acceptance of tools by users.

Participatory design workshops are essential for allowing developers, business representatives and users work together to design a solution. For example, Muller [52] suggested the PICTIVE concept. Following this concept, designers created mock-ups of the envisaged system on the basis of their knowledge about the application domain and the existing work practices. These mock-ups are discussed in the workshop to evaluate the questions of interest. Users that are expected to contribute to semantic content creation and ontology development (section 2.1) are invited to take part in direct evaluation of prototypes and discussion with software developers. Design workshops create an 'in-between' region which contains attributes of the environments of both participating groups, i.e. software designers and participants of the application domain. The workshops are documented on several media, video and audio recordings, pictures, and notes, as well as the 'workspace' created. The major difference between rapid prototyping and PICTIVE is the fact that PICTIVE does not use computer technology that can be confusing to the non-technical participant. The computer layperson would be disempowered using the rapid prototyping method because of the technology involved in the design process. The result of this would be the user being forced to express their ideas through the developer or designer [53]. See also additional participatory techniques such as the CARD technique [73], which refers to Collaborative Analysis of Requirements and Design.

In addition, cultural probes can be used to allow users to articulate their experiences in their working contexts [27], [7]. Cultural probes are sets of simple artefacts (such as maps, postcards, cameras, or diaries) that are given to users for them to record specific events, feelings or interactions in their usual environment, in order to get to know them and their culture better. Cultural probes are used to uncover aspects of culture and human interaction like emotions, values, connections, and trust. Originally the cultural probes are playful artefacts that support an engagement of participants early in the design process. Over the last years many different probes variations were introduced (for an overview see [7]), including those called Technology Probe [40] or Infrastructure Probes [14], which enable users to document and reflect on their IT infrastructures. These probes would allow for a wider involvement of end-users than just using direct ethnographic observation or experiments as stand alone analytical tools.

3.2 Methods for End-User Development

Early PD approaches only deal with user participation in *design time*. Since not all potential users can participate in the design process, the effects of these PD methods are limited on those who actually participated. Some studies have shown that evolutionary and participative software development has to be supplemented by activities performed by end-users or local experts of the application environment ([35], [45]). The approaches of tailorable systems and End-User Development (EUD) aim to overcome this problem by designing highly flexible systems that enable users to participate during the use of the system by adapting and modifying the tools according to their needs/preferences ([45]). The main goal of EUD is to empower end-users to develop and adapt systems themselves, by design them to be easy to understand, to learn, to use, and to teach as well.

Fischer and Giaccarcdi [24] suggest a meta-design framework to create socio-technical environments that empower domain experts to engage actively in the continuous development of systems. Meta-design defines and creates not only technical infrastructures for the software system but also social infrastructures in which users can participate actively as designers to shape and reshape the socio-technical systems through collaboration. Fischer et al. [23] present guidelines for using the metadesign framework to design socio-technical systems. They are called metadesign guidelines and include recommendations such as: Support human-Problem interaction; Under design for emergent behavior; enable legitimate peripheral participation; share control; promote mutual learning and support; reward and recognize contributions; foster reflective communities

To provide users with EUD-enabled IT systems, Lieberman et al. [45] discuss three main strategies: flexibility of technology, intuitive interfaces and appropriation support. With respect to social networks and online communities (e.g., open source software community), during the last years the participatory design discourse has paid more attention to collaborative, community-oriented design activities and to autonomous, self-organized initiatives by end-users. Kahler [41] and Pipek [60] suggest that EUD activities should be supported by the building of communities in which end users can effectively share their EUD-related knowledge and artefacts with their peers. In the INSEMTIVES project, ideas of EUD shall inform the development of tools and applications, especially the user interface design to allow for easy adaptation to users' needs and the collaborative usage of semantic annotation applications.

3.3 Methods for Integrated Organization and Technology Development (OTD)

While PD mainly focuses on the design of artefacts, OTD focuses on the introduction, appropriation etc. of artefacts/tools into social practice and integrates organization development methods and community development. As described in sections 2.1 and 2.2 human contribution to semantic annotation tasks depends on collaborative efforts (e.g., processes of negotiation and consensus-building for ontology development in semantic content communities) and on the motivation of end-users to participate in these tasks.

According to the insight that participation is a social and cooperative activity not for individuals but for groups of users, it has been argued that tailoring activities have to be embedded in an enabling and supporting "tailoring culture" ([38]). Historically the early approach of "Socio-technical Systems" (STS) tried to integrate social and technological development processes. The Tavistock approach of socio-technical systems (STS) focuses complex organizational work design that recognizes the interaction between people and technology in workplaces [17]. Emery and Trist [17] looked upon organizations as an integration of a technical system and a human (social) system. According to their assumptions, it organizational success cannot be achieved by optimizing either the technical or the social system but to aim at a "joint optimization" of both of them, taking into consideration the social and psychological impact of technology on workers [18]. Since the 1960's, the STS approach influenced very much the discussions on quality of working life and the Scandinavian school of user participation in system development ([58], [3]). Furthermore, STS was a very important source of inspiration for the discipline of Participatory Design (e.g., [6]).

Despite the original focus on working environments, the STS approach is promising for a wide range of collaborative and technologically supported activities. Since semantic content creation and ontology development refers to collaborative efforts, a shared understanding of tasks at hand, and consensus-building/division of labour in user communities, STS offers an integrated perspective for the support of social practices in semantic content creation (see sections 2.1 and 2.2). With its emphasis of the integrative focus on technological and social systems, STS can be seen as theoretical model preceding later approaches like the Scandinavian participatory design method MUST [43], [42] or the OTD framework presented in the following.

The integrated approach of Organization and Technology Development (OTD) combines cultural issues, organizational development processes and participatory design concepts [67], [76], [65]. In accordance with approaches of organization development, the OTD framework provides orientation for analysis, planning, intervention and evaluation in software design and introduction projects in organizational settings. For the elaboration of the OTD approach in the early nineties, the PD-influenced, cyclic process model for software-engineering STEPS has been adopted. End user-oriented activities of tailoring and the enabling frame by an established socio-organizational "tailoring culture" were considered to be important elements of a framework for participatory technology development within the broader context of integrated organization and technology development.

The original OTD approach was based on theories on:

- human activity and work psychological task analysis,
- theories on organization development,
- and evolutionary and participative models of software engineering and participatory design.

Methodologically, OTD follows the paradigm of the action research approach, which was introduced by the social psychologist Lewin [44]. The approach focuses on intervention in real world systems to improve practice and it is typically conducted cooperatively by a combined team of practitioners and researchers. Although Lewin [44] suggested a three-step spiral process of planning, taking actions and collecting data about the results of the action, Figure 2 illustrates the cyclic character of the re-iterations in the action research approach by a four-step model. According to this model, action research is performed in a cyclic process of collection of data about the organization and its problems, presenting and discussing these data within the organization, planning of interventions to overcome the problem and performing the intervention within the organization. Afterwards the same steps are re-iterated: data about remaining problems are collected and so on [25].

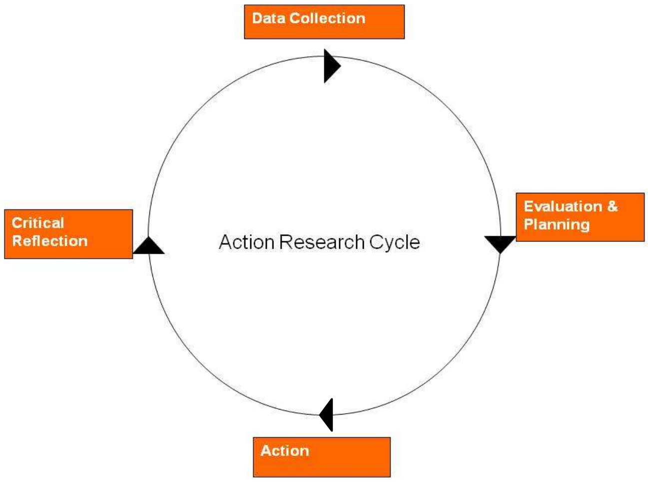

Figure 2: Action Research Cycle

Overall, the action research paradigm can be described as a family of research methodologies, which focus on action (or organizational change) and research (or understanding) at the same time. It is based on an emergent process, which takes shape as understanding increases. Action research emphasizes direct professional-client collaboration and focuses on group dynamics as the appropriate basis for practical problem solving. Therefore, it usually combines participative and qualitative methods of analysis, planning, intervention and evaluation. Since action research is focused on understanding of practice, mainly ethnographic and descriptive methods are used. Concerning collaborative human contribution to semantic content creation, annotation tools should be tested/evaluated in the social practice of end-users, prototypical implementations would be introduced into usage practice and re-designed according to a critical evaluation.

Integrated Organization and technology development (OTD) is defined as "the process of change of an organization in which organization and technology are designed and developed task- and needs-oriented by the members affected with integrated means: the organization members affected consider the existing problems, search and evaluate the problems' causes, and negotiate a process of problem solving. OTD only takes place if and as long as the members affected are willing and able to define contents and course of the OTD-process and are taking part in immediately (...)" ([35], 311, translation by the authors). The OTD-process is characterized by the concentration on parallel development of workplace, organizational and technical systems, on the management of (existing) conflicts by discursive and negotiative means, on immediate participation of the organization members affected, and on the concept of organizational margin ([36]).

OTD is focused on participation of organization members and users of technology and therefore uses participatory, ethnographic and qualitative methods for analysis and evaluation, e.g., interviews, participant observation, workshops. With regard to technology development, methods for participators design such as different methods of prototyping are applied. To test technical requirements, methods for usability testing (usability lab, video observation etc.) and semi-experimental questionnaire studies, based on scenarios are conducted (e.g., [77]).

Based on these considerations and approaches, the OTD framework offers a process model for the design and introduction of groupware in a specific work system within an organization (Figure 3).

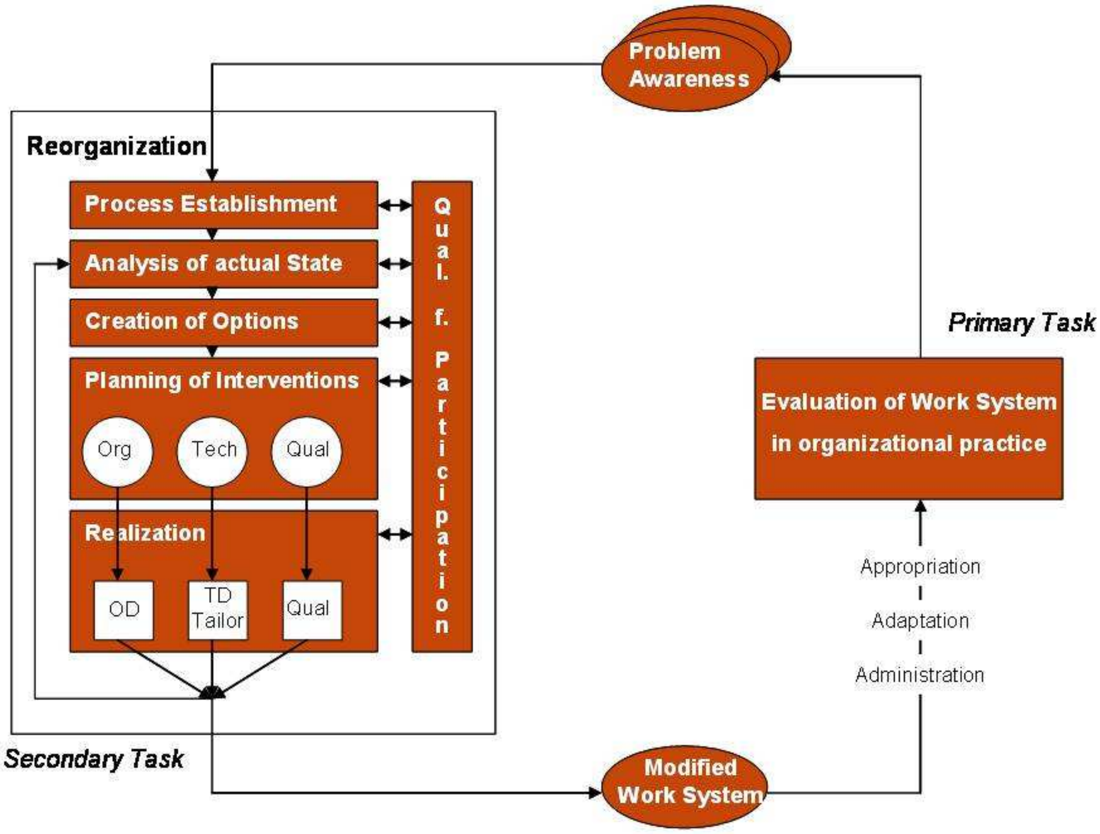

Figure 3: Process Model "Integrated Organization and Technology Development (OTD)"

The OTD framework was applied to software development projects for virtual or online communities ([59], [64], [66]). When software systems are designed and introduced for non-traditional, more informal organizational structures (such as virtual social networks and communities), informal socio-dynamic processes gain more relevance. Especially if the design of work places is not the primary focus but processes of networking and community-building, theoretical concepts of social identification, trust-building and collaborative learning have to be analyzed with regard to their possible impact on community-ware design and introduction.

Based on these concepts, the INSEMTIVES project will apply methods from PD, EUD and OTD for the development and design of user-centered tools for the participation of end-users in a semantic content authoring community.

4 Evaluation

The goal of evaluation is to provide feedback in system development, thus supporting an iterative development process. Preece et al. [61] list four main reasons for conducting evaluation: (1) Understanding the real world. How do users employ the technology? (2) Comparing designs: which is the best? (3) Engineering toward a target: Is it good enough? (4) Checking conformation to standard. Does this product conform to the standard? Evaluation is an ongoing process. During the development of a product, it provides feedback whether the development is moving toward the desired requirements. After the product is finished, it helps to understand whether the final product satisfied certain standards or requirements [72].

In the context of INSEMTIVES, the evaluation is driven by questions about how well the design or particular aspects of it satisfy user's needs. Some of these questions can be high-levels goals to guide the evaluation. Others may be much more specific. The design goals formulated earlier such as designing for usability, designing for sociability, etc. can be regarded as high-level goals. Each of them can be evaluated with respect to how well the design satisfies these general requirements.

Since the project focuses on incentive mechanisms for users of semantic technologies such as annotation tools, evaluation in INSEMTIVES takes users' needs as highly relevant criteria for requirements specification and evaluation of these applications. In particular, first, software engineers are designing applications and tools in context of the INSEMTIVES platform, and second, in three use cases existing platforms (like seekda! Web services search engine, TID knowledge management intranet portal and the PGP gaming applications) are further developed and re-designed according to the INSEMTIVES framework's requirements. Therefore, evaluation studies conducted to test the usability of existing systems/applications and prototypes. Furthermore, in requirements analysis user needs are evaluated with respect to design requirements for prototyping and software development. All technical INSEMTIVES tools and applications will be developed in two design/re-design cycles. For both cycles evaluation of prototypes and revisions will be evaluated with regard to functionality, usability, sociability and motivational issues.

There is a set of evaluation methods for user-centered testing/evaluation/validation of systems, which can be used through the entire life cycle of a product. In this section we present the most relevant methods, which can be applied in the context of INSEMTIVES to evaluate multiple concerns.

We should note that in the literature the methods are clustered in different ways. For example, Preece et al. [61] summarized evaluation methods into four core evaluation paradigms (i.e., "quick and dirty" evaluations; usability testing; field studies and predictive evaluation). In contrast, Te'eni et al. [72] grouped the most commonly used methods into the following categories: analytical and empirical evaluations. The main difference is that analytical evaluations normally do not need evidence from users but rely on evaluators using structured approaches for inspections, while empirical evaluations draw conclusions based on empirical data, which can be qualitative or quantitative in nature. In this section, we group methods into these two categories and follow mainly to Te'eni et al. [72].

4.1 Analytical Methods

Analytical methods are normally conducted by experts or designers to inspect potential design problems. Most of them do not need to involve users. There are several methods suggested in the literature (see [61], [72], [69]). In order to complement the experimental methods described in section 2.2 we will deploy a set of methods focusing specifically on software design, and in particular on the evaluation of usability/sociability of annotation tools. To derive new design requirements for software products according to users' needs, we rely on methods that are explorative and hopefully lead to a richer understanding of the use practice of technical applications.

We describe briefly the following commonly used methods:

1. Heuristic Evaluation
2. Guideline Review
3. Cognitive Walk-Through
4. Pluralistic Walk-Through

4.1.1 Heuristic Evaluation

Heuristic Evaluation is an informal usability inspection method [56]. It is the mostly widely adopted method in practice due to its ease of application [72]. This evaluation is guided by a set of higher-level usability principles known as usability heuristics such as visibility of system status or user control and freedom. A group of evaluators (experts, developers, or even novices with some training) can evaluate whether user-interface elements (e.g. menus, navigation structure) conform to the principles. The application of a heuristic requires first an adoption of a limited set of heuristics that are easy to understand and relevant to the product. Examples of heuristics and the description of the process of heuristic evaluation can be found in [56], [69].

4.1.2 Guideline Review

Guideline Review method is guided by a set of guidelines which are derived from practical experience or empirical studies. Expert or designers check the conformance of the interface with the guidelines. Guideline review can consider guidelines for specific task or activities (for example, guidelines for navigation or for getting the user's attention). This type of evaluation can be done in early and late stages of the development as well as for finished products (consult [69] for more).

4.1.3 Cognitive Walk-Through

Cognitive Walk-Through is a usability inspection method used to identify usability issues in a system or web site [56]. It focuses on how easy it is for new users to accomplish tasks with the system. A cognitive walkthrough starts with a task analysis that specifies the sequence of steps or actions required by a user to accomplish a task, and the system responses to those actions. The evaluators of the software then walk through the steps. As they do this, they try to answer the following questions [61]: (1) Will the correct action be sufficiently evident to the user? (i.e., will the user know what to do to achieve the task?) (2) Will the user notice that the correct action is available? (Can users see the button and menu items that they should use for the next action?) (3) Will the user associate and interpret the response from the action correctly? (Will users know from the feedback that they have made a correct or incorrect choice of action?). Data is gathered during the walkthrough, and afterwards a report of potential issues is compiled. The method can be applied early in the design phases to detect the problems early on so that they may be removed or fixed.

4.1.4 Pluralistic Walk-Through

Pluralistic Walk-Through is another type of walkthrough in which users, developers and usability experts work together to step through a task scenario, discussing usability issues associated with dialog elements involved in the scenario steps [56]. Each group of experts is asked to assume the role of typical users. The walkthroughs are done by following a set of steps (see [61]), which include the development of scenarios, the presentation of the scenario to the panel of evaluators, writing down the sequence of actions that the evaluators would take to move from one screen to another, discussing the documented actions. The benefits of pluralistic walk-through include a strong focus on user's tasks. The approach also lends itself well to participatory design practices by involving a multidisciplinary team in which users play a key role. The pluralistic walk-through is limited to only a number of scenarios that can be evaluated.

Finally, we should note that there are further analytical evaluation methods that we have not described here. For example, Te'eni et al. [72] refer to framework-based inspection such as using the Task-Semantic-Syntactic-Lexical (TSSL) model and user-model-based analysis such as using the GOMS model to predict user's behavior and performance during interaction with a system. Nevertheless, regarding the evaluation of prototypic semantic annotation technologies in INSEMTIVES we consider the above mentioned methods in context of usability test studies as being appropriate means.

4.2 Empirical Methods for Investigation into Practice

Experimental and field-experimental methods, as described in chapter 2.2, are generally used to test hypotheses, e.g., derived from game theoretical assumptions. With respect to actual usage practices in new social environments, we need methods capable of helping practitioners to generate context-specific hypotheses and assumptions. To derive this kind of hypotheses on user behavior, the appropriation and adaptation of annotation tools and generate assumptions that are grounded in users' practice, we will apply mainly explorative empirical methods.

These socio-empirical and ethnographic methods are usually conducted in prototyping processes by involving users and collecting facts about users interacting with computers [72]. Data being collected can be either quantitative or qualitative in nature. Commonly used methods include:

1. Survey or Questionnaire
2. Interview studies including focus groups
3. Lab experiment
4. Field studies

4.2.1 Survey/ Questionnaire

Survey/Questionnaire is commonly used to collect quantitative data from a large poll of respondents. A survey may focus on opinion or factual data depending on its purpose. But all surveys involve administering questions to individual respondents [72]. A survey can be conducted in ways such as telephone, e-mail, or mail. The advantage of survey methods include being inexpensive and flexible to conduct, involving a large number of respondents, thus offering validity, reliability, and statistical significance; allowing anonymity of respondents, thus encouraging more candid and honest responses. There are also several disadvantages of the surveys: surveys cannot reveal the complex and or dynamic nature of certain phenomena. Respondents may sometimes unable to answers questions appropriately. Especially with regard to technical design requirements and prospective questions, large-scale survey studies are limited with regard to their internal and external validity. While internal validity is increased by hypothesis-based methods like experiments, the external validity can be increased by interview or field studies that go deeper into the details of the investigated subject of matter.

4.2.2 Interviews and focus groups

Interview is viewed as a “conversation with a purpose” between one or more interviewers and one or more interviewees. A focus group is a form of group interview where the interviewees a group of people that are selected to represent typical users [72]. Interviews can be open-ended (unstructured), semi-structured, or structured. They indicate the level of control the interviewers impose on the conversation by following a predetermined set of questions. Interviews can be productive if specific issues are addressed and discussed directly with the users. Interviews can also be costly and time-consuming, and they require skilled interviewers. Regardless of the structure of the interviews, all questions should be carefully thought out and presented. There are many guidelines for how to develop questions.

In the INSEMTIVES project we will apply semi-structured in-depth interviews following collaboratively constructed interview guidelines. The interviews are recorded (mainly audio-recordings), transcribed by trained researchers and analyzed according to ex-post categories. In cases of usability studies of existing prototypes in INSEMTIVES the interview technique will integrate the observation of concrete user interactions with technological systems. These observation/interview sessions will be video-recorded and analyzed according to observation schemes by trained researchers.

Individual interviews will be complemented by group interview and discussion settings which focus on specific design alternatives and solutions. These group interviews focusing on concrete design choices are going to be conducted as video-recorded focus groups.

4.2.3 Lab Experiments

Lab experiments allow evaluators to address specific aspects of design by manipulating a number of factors associated with the design and assessing the related user performance [72]. That is, some other factors can be controlled or excluded so that a better understanding of the issues at hand can be achieved. Lab experiments can be conducted on a system in different development stages, such as mock-up, and advanced prototype, or a finished product, especially if one aims at testing (verification vs. falsification) of particular hypotheses. Naturally, this kind of test can be fruitfully employed once the practice to be evaluated is in place and the main causal linkages between variables have been established. This is why, during the early stages of the project, we will mainly apply ethnographic methods in the field that will allow us to improve the analytical understanding of the situations encountered in the use cases and to better design the experimental stage of the study.

4.2.4 Field Studies

Field studies are conducted to observe and monitor the real users actually using the system in a real setting, i.e. in a normal working environment or the real field. Field studies can be valuable during the development as well as in use and impact evaluations [72]. Longitudinal data collection is possible by observing users over an extended period of time. Ethnographic observation attends to actual user experience in real-world situations with minimum interruption or intervention from the evaluators. Data may be collected by evaluators taking notes, recording with video or audio devices, or using computer logs. Several actual use episodes can be observed if necessary. Data analysis can be tedious and time-consuming. Depending on the data collected and evaluation goals, both qualitative (notes, video or audio data) and quantitative (logs) data analysis may take place. See [21] and [54] for more in depth description of field studies.

The INSETIVES use cases offer appropriate research fields for the application of methods aiming at gaining a rich and deep understanding of use practice and the derivation of user-centered design requirements for the technological support of human contribution to semantic content creation (section 2.1). Therefore, field study methods will be preferred to lab experiments in the early stages on the INSEMTIVES project.

4.3 General Guidelines for Conducting Evaluations

With regard to our evaluation studies in INSEMTIVES we mainly follow the guidelines of the DECIDE approach. Preece et al. [61] suggest a framework (called DECIDE) to guide evaluation. Well-planned evaluations are driven by clear goals and appropriate questions. The DECIDE framework is used to guide evaluations according to the following checklist:

1. Determine the goals
2. Explore the questions
3. Choose the evaluation paradigm and techniques
4. Identify the practical issues
5. Decide how to deal with the ethical issues
6. Evaluate, interpret, and present the data

Preece et al. [61] also recommend to do pilot studies which refer to a small trial run of the main study. The aim of a pilot study is to make sure that the plan is viable and that the procedure, the interview scripts, questionnaires, experiments, etc. are valid. In the INSEMTIVES project, usability testing of existing application with students and test users are designed as pilot studies for the later field studies in the use cases.

5 Conclusions

In this deliverable we addressed the methodological issue of how to design and implement semantic authoring tools that motivate users to participate in semantic content creation. For this purpose, we described the basic activities of a comprehensive methodological approach, which include the analysis, design and evaluation of the tools. The activities should be applied iteratively and in an integrated fashion. For example, design and evaluation activities can take place within the analysis activity to support the understanding of user requirements.

For the analysis activity we have presented several methods that can be employed to understand the requirements related to the users, the tasks and contexts. The design and implementation activities included methods for designing of tools as well as for implementing them in (organizational) contexts. Here, the Participatory Design paradigm provides different methods to include users in design time whereas the End-Users Development paradigm provides methods for how to design systems that are adaptable and enabling users to participate in use time. In addition, the Integrated Organization and Technology Development provides methods for integrating technology design and the design of organizational mechanism (e.g., policies, incentives) or activities such as community development in order to promote user participation. Finally, for the evaluation activity several methods have been described to test a product through its entire life cycle, i.e., during its design (formative evaluation) as well as after the product has been finished (summative evaluation).

The methodology we have suggested for INSEMTIVES integrates the ideas and methods from these paradigms. Yet, we do not claim that the list of methods for the activities is complete. We are aware of that there are a variety of other methods and also methods that may emerge during the research period of the INSEMTIVES project. Therefore, it is intended to include further methods and improve the integration of the methods at a later stage of the INSEMTIVES project.

References

[1] Barnard, C. (1938): “The functions of the executive”. Cambridge, MA: Harvard University Press.

[2] Benson, P.G. (2000): “The Hawthorne effect”. In W.E. Graighead and C.B. Nemeroff (eds.), “The Corsini Encyplodepia of Psychology and Behavioral science”, New York: Wiley, third edition, Volume 3.

[3] Bjerknes, G., Bratteteig, T. (1988): “The memoires of two survivors or Evaluation of a computer system for cooperative work”. In: CSCW 88, Proceedings of the Conference on Computer-Supported Cooperative Work. ACM SIGCHI & SIGOIS, Protland, Oregon, 167-177.

[4] Bjerknes, G., Ehn, P., Kyng, M. (Eds.) (1987): “Computers and Democracy - a Scandinavian Challenge”. Aldershot, England, Avebury.

[5] Bødker, K.; Kensing, F.; Simonsen, J. (2005): Participatory IT Design. Designing for Business and Workplace Realities. Boston: MIT Press.

[6] Bødker, S., Grønboek, K. (1991): “Design in Action: From Prototyping by Demonstration to Cooperative Prototyping”. In: J. Greenbaum & M. Kyng (Eds.): “Design At Work - Cooperative Design of Computer Systems”, Hillsdale, N.J.: Lawrence Erlbaum Associates, 197-218.

[7] Boehner, K., Vertesi, J., Sengers, P., and Dourish, P. (2007): “How HCI interprets the probes”. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems (San Jose, California, USA, April 28 - May 03, 2007). CHI '07. ACM, New York, NY, 1077-1086.

[8] Brank, J., Grobelnik, M., and Dunja, M. (2005): “A survey of ontology evaluation techniques”. In: Conference on Data Mining and Data Warehouses (SiKDD).

[9] Camerer, C. and Hogarth, R. (1999): “The effect of financial incentives in experiments: a review and a capital-labor production framework”. In: Journal of risk uncertainty 19, 7–42.

[10] Colby, A., Sippola, L., Phelps, E. (2001): “Social responsibility and paid work in contemporary American life”. In: A. S. Rossi (Ed.): “Caring and doing for others: Social responsibility in the domains of family, work, and community”, Chicago: University of Chicago Press, 463–501.

[11] Cristani, M. and Cuel, R. (2005): "Ontology Methodologies: a Survey". Invited paper on International Journal on Semantic Web and Information Systems.

[12] Deci, E. L., Ryan, R. M. (2000): “The “what” and “why” of goal pursuits: Human needs and the self-determination of behavior”. Psychological Inquiry, 11, 227-268.

[13] Dörner, C. (2009): “Tailoring Software Infrastructures: Integration of End-User Development and Service-Oriented Architectures”. (Dissertation), University of Siegen, Germany.

[14] Dörner, C., Hess, J., Pipek, V. (2008): “Fostering user-developer collaboration with infrastructure probes”. In: Proceedings of the 2008 international workshop on Cooperative and human aspects of software engineering.

[15] Ehn, P. (1993): “Scandinavian design: On participation and skill”. In: Douglas Schuler and Aki Namioka (eds): “Participatory Design: Principles and Practices”, Hillsdale, NJ: Lawrence Erlbaum Associates, p. 41-77.

[16] Ehrig, M. and Sure, Y. (2004): “Ontology mapping - an integrated approach”. Technical report, AIFB Karlsruhe.

[17] Emery, F.E., Trist, E.L. (1960): “Socio-Technical Systems”. In: Management Sciences, Models and Techniques, vol. 2. London.

[18] Emery, F.E., Trist, E.L. (1969): “Socio-technical systems”. In: F.E. Emery (eds.): “Systems thinking”, Harmondsworth: Penguin, 281-296.

[19] Euzenat, J. and Shvaiko, P. (2007): “Ontology Matching”. Berlin Heidelberg Springer.

[20] Euzenat, J., Mocan, A., and Scharffe, F. (2007): “Ontology Alignments”, Semantic Web and Beyond, 6, Springer.

[21] Falconer, S. M. and Storey, M.A. (2007). “A cognitive support framework for ontology mapping”. In Asian Semantic Web Conference (ASWC 2007).

[22] Fetterman, D. M. (1998): “Ethnography: Step by step”. Second Edition, Sage, Thousand Oaks, CA.

[23] Fischer, G., Nakakoji, K., Ye, Y. (2009): “Metadesign: Guidelines for supporting Experts in software development”. IEEE Software, September/October 2009, 37-44.

[24] Fischer,G., Giaccardi, E., Ye, Y., Sutcliffe, A.G., Mehandjiev, N. (2004): “Meta-design: A Manifesto for End-User Development”. In: Communications of the ACM, 47(9) 2004, 33-37.

[25] French, W.L., Bell, C. H. (1973): “Organization Development”. Englewood Cliffs

[26] Friedberg, E. (1993) : “Le pouvoir et le regle. Dynamique de l’action organisée”. Paris : Seuil.

[27] Gaver, B., Dunne, T., and Pacenti, E. (1999): “Design: Cultural probes”. In: “Interactions” 6, 1, 21-29.

[28] Gomez-Perez, A. (2003): “A survey of ontology learning methods and techniques”. Technical report, OntoWeb Deliverable 1.5.

[29] Gomez-Perez, A., Fernandez-Lopez, M. and Corcho, O. (2004): “Ontological Engineering”. In: “Advanced Information and Knowledge Processing”, Springer.

[30] Greenbaum, J. and Kyng, M. (1991): “Design at Work”. Hillsdale, NJ: Lawrence Erlbaum Associates.

[31] Griffin, R.W. (1982): “Task Design: An Integrative Approach”. Scott, Foresman and Company, Glenview, IL.

[32] Hackman, J. R., and Oldham, G. R. (1980): “Work redesign”. Reading, MA: Addison-Wesley.

[33] Hackman, J. R., Oldham, G. R. (1976): “Motivation through the design of work: Test of a theory”. In: “Organizational Behavior and Human Performance”, 16(2), 250-279.

[34] Harrison, G.W. and List, J.A. (2004): “Field experiments”. In: Journal of Economic Literature 42, 1009–55.

[35] Hartmann, A. (1994): „Integrierte Organisations- und Technikentwicklung – ein Ansatz zur sach- und bedürfnisgerechten Gestaltung der Arbeitswelt“. In: Hartmann, Anja; Herrmann, Thomas; Rohde, Markus; Wulf, Volker (Hrsg) 1994: “Menschengerechte Groupware - Software-ergonomische Gestaltung und partizipative Umsetzung“, Stuttgart: Teubner, 303-328.

[36] Hartmann, A., Rohde, M. (1993): „Organisations- und Dispositionsspielräume und ihre Bedeutung für die Gestaltung vernetzter Systeme2. In: H. Reichel (Hrsg.): “Informatik, Wirtschaft, Gesellschaft”, Berlin, Springer, 175-180.

[37] Hartmann, J., Sure, Y., Giboin, A., Maynard, D., Suarez-Figueroa, M. C. and Cuel, R. (2005): “Methods for ontology evaluation”. Technical report, Knowledge Web Deliverable D1.2.3.

[38] Henderson, A. and Kyng, M. (1991): “There's No Place Like Home: Continuing Design in Use”. In: Greenbaum, J.; Kyng, M.: “Design at Work - Cooperative Design of Computer Artefacts”, Hillsdale, 219 - 240.

[39] Hertwig, R. and Ortmann, A. (2001): “Experimental practices in economics: a methodological challenge for psychologists?”. In: “Behavioral and Brain Sciences”, 24, 433–451

[40] Hutchinson, H., Mackay, W., Westerlund, B., Bederson, B., Druin, A., Plaisant, C., Beaudoun-Lafon, M., Conversy, S., Evans, H., Hansen, H., Rousell, N., Eiderbäck, B., Lindquist, S., Sundblad, Y. (2003): “Technology Probes: Inspiring design for and with families”. In: Proceedings of the SIGCHI conference on Human factors in computing systems (2003), 17-24.

[41] Kahler, H. (2001): “Supporting Collaborative Tailoring”. Roskilde University, Datalogiske Skrifter.

[42] Kensing, F., Blomberg, J. (1998): “Participatory Design: Issues and Concerns”. In: “Computer Supported Cooperative Work”, 7, Dordrecht, The Netherlands, Kluwer Academic Publishers, 167-184.

[43] Kensing, F., Simonsen, J., and Bødker, K. (1998): “MUST - a Method for Participatory Design”. In: “Human-Computer Interaction”, vol. 13, no 2.

[44] Lewin, K. (1946): “Action research and minority problems”. In: Journal of Social Issues, 2, 34-46.

[45] Lieberman, H., Paternó, F., Wulf, V. (eds.) (2006): “End User Development”. Springer, Dordrecht.

[46] Locke, E. A., and Latham, G. P. (1990): “A theory of goal setting and task performance”. Englewood Cliffs, NJ: Prentice Hall.

[47] March, J.G., Simon, H.A. (1958): “Organizations” New York: Wiley.

[48] Mas-Colell, A., Whinston, M., and Green., J. (1995): “Microeconomic Theory”. New York, NY: Oxford University Press.

[49] Masino, G. and Zamarian, M. (2003) “ Information technology artefacts as structuring devices in organizations: design, appropriation and use issues”. In: Interacting with Computers, 15, 693-707.

[50] Mørch, A.I., Mehandjiev, N. D. (2000): “Tailoring as Collaboration: The Mediating Role of Multiple Representations and Application Units”. In: “Computer Supported Cooperative Work”, 9(1), 75-100.

[51] Morgeson, F. P., and Humphrey, S. E. (2006): “The Work Design Questionnaire (WDQ): Developing and validating a comprehensive measure for assessing job design and the nature of work”. In: Journal of Applied Psychology, 91, 1321-1339.

[52] Muller, M. (1991): “PICTIVE- Exploration in participatory design”. In: Proceedings of CHI’91. 221-23.

[53] Muller, M., Widlman, D., and White, E. (1993): “Equal opportunity: PD Using PICTIVE”. In: Communications of the ACM, 36(4), 64-66.

[54] Nardi, B. (1997): “The use of ethnographic methods in design and evaluation”. In: M. Helander, T. K. Landauer, & P. Prabhu (eds.): “Handbook of Human-Computer Interaction” (Second Edition). Amsterdam: Elsevier, 361-366.

[55] Nardi, B. A. (1993): “A Small Matter of Programming - Perspectives on end user computing”. MIT-Press, Cambridge.

[56] Nielsen J., Mack R. (1994): “Usability Inspection Methods”. John Wiley and Sons.

[57] Noy, N. and Musen, M. (2001): “Anchor-prompt: Using non-logical context for semantic matching”. In: IJCAI Workshop on Ontologies and Information Sharing, Seattle (WA US), 63-70.

[58] Nygaard, K. (1986): "Program Development as a Social Activity". In: Information Processing 86, H.-J. Kugler (eds.), Elsevier Science Publishers B.V. (North Holland), IFIP, (Proceedings from the IFIP 10th World Computer Congress, Dublin, Ireland, September 1-5, 1986), 189-198.

[59] Pape, B., Reinecke, L., Rohde, M., Strauss, M. (2003): “E-Community-Building in WiInf-Central”. In: Proceedings of the 2003 International ACM SIGGROUP Conference on Supporting Group Work (GROUP 2003), ACM-Press, New York, 11- 20.

[60] Pipek, V. (2005): “From Tailoring to Appropriation Support: Negotiating Groupware Usage”. PhD Thesis, Faculty of Science, Department of Information Processing Science, University of Oulu, Oulu, Finland.

[61] Preece, J., Rogers, Y., Sharp, H. (2002): “Interaction Design: Beyond Human-Computer Interaction”. New York: Wiley.

[62] Reeve, L. and Han, H. (2005): “Survey of Semantic Annotation Platforms”. ACM Press, 1634-1638.

[63] Reynaud, J.D. (1989) :“Le regles du jeu. L’action collective e la regulation sociale”.Paris:Colin,.

[64] Rohde, Markus (2004): Find what binds. Building social capital in an Iranian NGO community system. In: Huysman, M., Wulf, V. (eds.) 2004: Social Capital and Information Technology, Cambridge: MIT Press, 75-112.

[65] Rohde, Markus (2007): Integrated Organization and Technology Development (OTD) and the Impact of Socio-Cultural Concepts - A CSCW Perspective. Datalogiske skrifter, University of Roskilde.

[66] Rohde, Markus; Reinecke, Leonard; Pape, Bernd; Janneck, Monique (2004): Community-building with Web-Based Systems – Investigating a Hybrid Community of Students, In: International Journal on Computer Supported Cooperative Work (JCSCW), 13, 471-499.

[67] Rohde, Markus; Wulf, Volker 1995: Introducing a Telecooperative CAD-System - The Concept of Integrated Organization and Technology Development. Workshop Computer Supported Cooperation in Product Design, In: Preceedings of the HCI International '95, 6th International Conference on Human-Computer Interaction in Yokohama, Elsevier Science Publishers, Amsterdam, 787-792.

[68] Rohde, Markus; Wulf, Volker 1996: An early evaluation of technical mechanisms supporting negotiations in groupware. In: Proceedings of the Second International Conference on the Design of Cooperative Systems (COOP 96), Juan les Pin, INRIA, 281-297.

[69] Shneiderman, B., Plaisant, C., Cohen, M., Jacobs, S. (2009): “Designing the User Interface - Strategies for Effective Human-Computer Interaction”. Addision-Wesley, 2009.

[70] Simon, H.A., (1947): “Administrative Behavior”. New York, NY: Macmillan,.

[71] Siorpaes, K. and Simperl, E. (2009): “Human intelligence in semantic content creation”. World Wide Web Journal (WWW), Springer, December 2009.

[72] Te'eni D., Carey J., Zhang P. (2007): “Human Computer Interaction, Developing Effective Organizational Information Systems”. Wiley & Sons 2007, USA

[73] Tudor, L., Muller, M., and Dayton, T. (1993): “A C.A.R.D. game for participatory task analysis and redesign: macroscopic complement to PICTIVE”. In: Interact '93 and Chi '93 conference companion on human factors in computing systems, 51-52.

[74] Turban, D. B., Greening, D. W. (1997): “Corporate Social Performance and Organizational Attractiveness to Prospective Employees”. Academy of Management Journal 40, 658–672.

[75] Uren, V., Cimiano, P., Iria, J., Handschuh, S., Vargas-Vera, M., Motta, E., and Ciravegna, F. (2006): “Semantic annotation for knowledge management: Requirements and a survey of the state of the art”. In: Web Semantics: Science, Services and Agents on the World Wide Web, 4(1), 14-28.

[76] Wulf, Volker; Rohde, Markus 1995: Integrated Organization and Technology Development - an Approach to Manage Change. In: Brandt, D.: Proceedings of the 5th IFAC-Symposium on Automated Systems based on Human Skills, 135-140.

[77] Wulf, Volker; Rohde, Markus 1996: Reducing Conflicts in Groupware: Metafunctions and their Empirical Evaluation. International Journal on Behaviour and Information Technology, Vol. 15, No. 6, 339-351.