



Automatic detection of accommodation steps as an indicator of knowledge maturing

Johannes Moskaliuk, Andreas S. Rath, Didier Devaurs, Nicolas Weber, Stefanie N. Lindstaedt, Joachim Kimmerle, Ulrike Cress

► To cite this version:

Johannes Moskaliuk, Andreas S. Rath, Didier Devaurs, Nicolas Weber, Stefanie N. Lindstaedt, et al.. Automatic detection of accommodation steps as an indicator of knowledge maturing. *Interacting with Computers*, 2011, 23 (3), pp. 247-255. 10.1016/j.intcom.2011.03.006 . hal-00861580

HAL Id: hal-00861580

<https://hal.science/hal-00861580>

Submitted on 13 Sep 2013

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.



Open Archive Toulouse Archive Ouverte (OATAO)

OATAO is an open access repository that collects the work of Toulouse researchers and makes it freely available over the web where possible.

This is an author-deposited version published in: <http://oatao.univ-toulouse.fr/>
Eprints ID: 8005

To link to this article:

DOI:10.1016/j.intcom.2011.03.006

Official URL: <http://dx.doi.org/10.1016/j.intcom.2011.03.006>

To cite this version:

Moskaliuk, Johannes and Rath, Andreas S. and Devaurs, Didier and Weber, Nicolas and Lindstaedt, Stefanie N. and Kimmerle, Joachim and Cress, Ulrike Automatic detection of accommodation steps as an indicator of knowledge maturing. (2011) Interacting with Computers, vol. 23 (n° 3). pp. 247-255. ISSN 0953-5438

Any correspondence concerning this service should be sent to the repository administrator:
staff-oatao@inp-toulouse.fr

Automatic detection of accommodation steps as an indicator of knowledge maturing

Johannes Moskaliuk^{a,*}, Andreas Rath^b, Didier Devaurs^c, Nicolas Weber^b, Stefanie Lindstaedt^b, Joachim Kimmerle^a, Ulrike Cress^d

^a University of Tuebingen, Germany, Konrad-Adenauer-Str. 40, 72072 Tuebingen, Germany

^b KnowCenter, Austria, Inffeldgasse 21a, A-8010 Graz, Austria

^c CNRS/LAAS and Université de Toulouse/UPS, INSA, INP, ISAE/LAAS; 7 avenue du colonel Roche, F-31077 Toulouse, France

^d Knowledge Media Research Center, Germany, Konrad-Adenauer-Str. 40, 72072 Tuebingen, Germany

ABSTRACT

Jointly working on shared digital artifacts – such as wikis – is a well-tried method of developing knowledge collectively within a group or organization. Our assumption is that such knowledge maturing is an accommodation process that can be measured by taking the writing process itself into account. This paper describes the development of a tool that detects accommodation automatically with the help of machine learning algorithms. We applied a software framework for task detection to the automatic identification of accommodation processes within a wiki. To set up the learning algorithms and test its performance, we conducted an empirical study, in which participants had to contribute to a wiki and, at the same time, identify their own tasks. Two domain experts evaluated the participants' micro-tasks with regard to accommodation. We then applied an ontology-based task detection approach that identified accommodation with a rate of 79.12%. The potential use of our tool for measuring knowledge maturing online is discussed.

Keywords:

Co-evolution

Knowledge development

Knowledge maturing

Task detection

Learning

Context-awareness

1. Introduction

Social software tools for knowledge work make new forms of collaboration and learning possible (Raitman et al., 2005; Reinhold, 2006). Wikis are first-hand examples of such tools. Through a wiki (a website that allows users to change its content online), individuals can create texts collaboratively. People can easily revise all parts of the text, add, change, or delete anything at their discretion. Wikis are used in various contexts where people build a shared representation of their knowledge, for example the community of Wikipedia users who are contributors to the online encyclopedia, or a working team who is documenting their working results. What is possible through wikis is not only an accumulation of knowledge (whereby the knowledge of many individuals is brought together and made available to others (cf. Kimmerle et al., 2007, 2008), but also knowledge emergence, that is, the creation of new knowledge (Johnson, 2001; Moskaliuk and Kimmerle, 2009). It has been repeatedly shown that jointly writing a wiki text is not only a method of sharing information, but also of developing new knowledge (e.g., Cress and Kimmerle, 2008). During their writing activity,

users may link their individual ideas to those of others, generate new and innovative ideas, discuss their own arguments with others, and reach agreement – otherwise their own text will probably be deleted by others. The creation of a shared, homogenous text compels people to construct shared meaning and build mutual understanding (Erkens et al., 2005). This enables an evolution of knowledge. Emergent knowledge – that was not part of the individual knowledge of any single user before – may arise as a result of this activity (Kimmerle et al., 2011).

Collaboratively written wiki text is influenced by the social and cultural background of its authors and represents the knowledge of the corresponding community (cf. Lave and Wenger, 1991; Vygotsky et al., 1978). Contributing to the wiki supports the reflection of one's own knowledge and stimulates individual learning processes as well as learning processes of the whole community or organization. This may lead to a continuous development of knowledge over time: knowledge matures from expressing individual ideas to a state of formalized knowledge at an organizational level (Schmidt, 2005). As each individual can manipulate the wiki text (through writing, deleting, changing content), it will develop, and a co-evolution between the individuals' knowledge and the wiki text may occur (Cress and Kimmerle, 2008). Individual knowledge and the shared information in a wiki text influence each other and trigger a mutual development of both: The wiki text matures over time and, at the same time, the individuals' knowledge develops further. This maturing process is condensed in the

* Corresponding author. Tel.: +49 (0)7071 979 204; fax: +49 (0)7071 979 322.

E-mail addresses: j.moskaliuk@iwm-kmrc.de (J. Moskaliuk), arath@know-center.at (A. Rath), devaurs@laas.fr (D. Devaurs), nweber@know-center.at (N. Weber), slind@know-center.at (S. Lindstaedt), j.kimmerle@iwm-kmrc.de (J. Kimmerle), u.cress@iwm-kmrc.de (U. Cress).

wiki text. By looking at the writing process closely, the knowledge development process itself can be analyzed, and new insights may be gained into how people construct new knowledge.

The central research question of this paper is to find indicators for knowledge maturing, and to develop a tool that automatically identifies maturing processes when users work on a wiki. For this purpose, we combine social-cognitive theories about the development of knowledge (cf. Cress and Kimmerle, 2008; Schmidt, 2005) with innovative methods of task detection (Dragunov et al., 2005; Oliver et al., 2006; Rath et al., 2009a; Shen et al., 2009). This approach takes the writing process itself into account to find an indicator for knowledge maturing.

We present an empirical study in which participants had to work on a wiki, improve the text, and add further arguments. The participants obtained additional information that contradicted the content of the wiki. This was supposed to lead to some prototypical editing steps and elicit the development of knowledge. We used the data to train a task detection tool and evaluate its performance. The goal was to identify those features of the writing process that will make it possible to measure knowledge development automatically.

In the following paragraphs, we will explain our task detection approach and our assumptions about knowledge maturing. We will introduce a framework model that explains the development of knowledge over time in terms of a co-evolution of cognitive and social systems. We assume that accommodation is one main indicator of knowledge maturing. We will then discuss what the task detection approach provides to detect accommodation steps automatically. In Section 3, we will describe our research setting and how we used it to develop and validate the task detection approach. The results will be presented in Section 4. In conclusion, we will discuss our results against the background of the question on how it can be ensured that knowledge workers contribute to the maturing of knowledge.

2. Theories on the development of knowledge

In this section, we present the co-evolution model of cognitive and social systems, which distinguishes between accommodation and assimilation processes, and introduce cognitive conflicts as the key incitement factor for knowledge maturing. We will then present the task detection approach and describe how it was used to analyze accommodation as an indicator of knowledge maturing.

2.1. Co-evolution model

The co-evolution model (Cress and Kimmerle, 2008; Kimmerle et al., 2010a) is a framework that describes the development of individual and collective knowledge with wikis and other social software tools. The model considers two relevant systems (cf. Luhmann, 1995, 2006): the wiki, as a representation of a social system, and individual knowledge, represented in the cognitive system of each user. Each system has its own specific mode of operation. The social system operates in the mode of communication (which becomes manifest as written text through a wiki). The cognitive system operates through individual psychological processes like perception, reasoning, or learning. When people contribute collaboratively to a wiki text, both the social and the cognitive system develop further and become more and more complex (cf. Kimmerle et al., 2010c). Individuals externalize their knowledge and contribute it to the wiki, and, at the same time, they internalize information from the wiki. So information contained in a wiki (that represents the knowledge of a community) evolves in the course of time. The wiki becomes more complex, which can be described as *knowledge maturing* (Schmidt, 2005).

At the same time, knowledge in the individuals' cognitive systems is increased, which can be described as individual learning. This mutual development of social and cognitive systems leads to a co-evolution of both systems. The model states that it is cognitive conflicts (Piaget, 1977a; Kimmerle et al., 2010b) that trigger this co-evolution. A cognitive conflict occurs if a person perceives information that does not match his or her own individual knowledge. Such conflicts motivate individuals to contribute to the wiki or to change their own knowledge structure, in order to establish some equilibrium between their own knowledge and information in the wiki. The model proposed by the Cress and Kimmerle (2008) specifies the co-evolution process and describes two different processes on the basis of the ideas of Piaget's ideas (1977a,b): assimilation and accommodation.

Assimilation means interpreting and explaining current experiences and new information by understanding it with existing schemas. Accommodation, in contrast, means changing one's own cognitive schemas, in order to fit them to new information. In Piaget's understanding, assimilation and accommodation are internal processes in the cognitive system; the co-evolution model, however, expands Piaget's concept by describing accommodation and assimilation not only from the perspective of an individual's cognitive system, but also as external processes of a social system: in the case of external assimilation, users make contributions that will not change the basic message and structure of the wiki, but only add supplementary aspects. External accommodation happens if users contribute their knowledge in such a way that the message is changed and, sometimes, new structures are being created. External accommodation has taken place if the text has been re-organized or new aspects have been integrated (Piaget, 1977a). So accommodation tends to result in some qualitative modification of the wiki text, whereas assimilation has primarily to do with quantity, introducing additional arguments or new examples, but no fundamental innovation (cf. Moskaliuk et al., 2009).

The co-evolution model postulates that a development of knowledge takes places in the course of contributing to the wiki: Evolution of individual cognitive structures occurs through internalization of new information from the wiki, integration with existing knowledge, and externalization of emergent knowledge back to the wiki. This leads, in turn, to a development of the wiki text. While assimilation only leads to some quantitative development, accommodation leads to a qualitative development in the sense of a more sophisticated knowledge structure, more balanced argumentation, and innovation. So if we are interested in knowledge maturing, we have to focus on accommodation steps during the writing process. These accommodation steps will lead to a higher complexity of the wiki and, accordingly, to the development of knowledge in other people's cognitive systems.

We need to differentiate this idea of accommodation steps as an indicator of knowledge maturing from related work on measuring the quality of text. Different methods exist to measure the quality of text with computational methods, like reading scores (cf. Flesch, 1948; Klare, 1974; Stvilja et al., 2005) text cohesion (Graesser et al., 2004), latent semantic analysis (Landauer and Dumais, 1997), or presentation format and numbers of links (Braun and Schmidt, 2007). There are additional methods that consider the specific characteristics of collaboratively written wiki text and use, apart from text features, also the available information about authors as one of their criteria for assessing text quality (cf. Wöhner and Peters, 2009; Hu et al., 2007; Kramer et al., 2008). These methods focus on the written text and consider semantic or linguistic features of the text, or they focus on the contributors and their edits. Our approach differs fundamentally from these methods in that we take the writing process itself into account. Here we can make use of the interaction of a person with the wiki text. In order to measure knowledge maturing, we adopt the method of task detection

to analyze the writing process and to detect accommodation steps. The task detection approach is described in the following subsection.

2.2. Task detection approach

Context-aware (or sentient) systems are systems that adapt their operations or behavior to their current context of use, without explicit user intervention. They can detect the current task, which a user is performing, making it possible to provide users with personalized and relevant support (Coutaz et al., 2005; Dey et al., 2001). Task detection methods have been used to recognize Web-based tasks (Gutschmidt et al., 2008), tasks within e-mails (Shen et al., 2009) or tasks from the complete desktop of a computer user (Rath et al., 2009a; Shen et al., 2009). Context-aware systems rely on sophisticated mechanisms for the acquisition and analysis of contextual information.

Context information may be gathered in a variety of ways, such as applying (physical or virtual) sensors, recording network information and device status, or browsing user profiles and organizational databases. A context model is needed for storing the recorded user context data in a machine-processable form. We use a populated ontology to model (Baldauf et al., 2007; Strang and Linnhoff-Popien, 2004) the context and construct training/testing instances to classify the current task users are performing. Automatic task detection is classically modeled as a machine learning problem, and – more precisely – a classification problem. It has already been possible to demonstrate that such an ontology-based task detection approach is applicable to routine and knowledge-intensive computer desktop tasks (Rath et al., 2010). For the current study, we consider accommodation steps as micro-tasks, which we aim to detect.

3. Study

Accommodation processes are considered to be triggered by cognitive conflicts (Cress and Kimmerle, 2008; Kimmerle et al., in press). In this study, cognitive conflicts were elicited by presenting

to people information in a wiki that was fairly different from their own knowledge. This enabled us to analyze the writing process in detail and to gain deeper insights into the external part of co-evolution and the according processes of knowledge maturing.

3.1. Writing task

We designed a writing task in which participants had to contribute to a wiki text. In order to trigger accommodation processes, we provided the participants with additional information, which was not part of the wiki text. The information presented in the wiki and the additional information contradicted each other. This was supposed to provoke accommodation steps and is therefore considered as an ideal condition for obtaining the required data set. The text was about “pros and cons” of violent computer games, and the participants’ task was to complete a given text of a wiki (initial state), with the demand that a scientifically balanced article about violent computer games and their risks to users and society should be written. Two windows were available to participants on their monitors during the study: a wiki page and a page with the additional information. Those two windows were accessible through two tabulators at the top of the page. The wiki page (wiki-tab) presented a text that, initially, was biased towards a position of opposing violent computer games. Here, only the risks and dangers of violent computer games were presented. The page with additional information (info-tab) contained ten different arguments that were biased in the other direction, invalidating arguments in the wiki text or explaining why these were one-sided or incorrect. Participants could copy, paste, delete, and edit the wiki freely or type in new text. Fig. 1 shows a screenshot of the writing software.

Participants had a total of 50 min to edit the wiki text. They were instructed to save their changes after each set of alterations that “belonged together” in their opinion. This procedure led to a series of single micro-tasks (e.g., adding sentences or rewriting an argument). Two independent domain experts rated each micro-task on the basis of the rating scheme used by the Moskaliuk et al. (2009). Therefore, the experts obtained a separate document for each micro-task. This document compared the two versions of

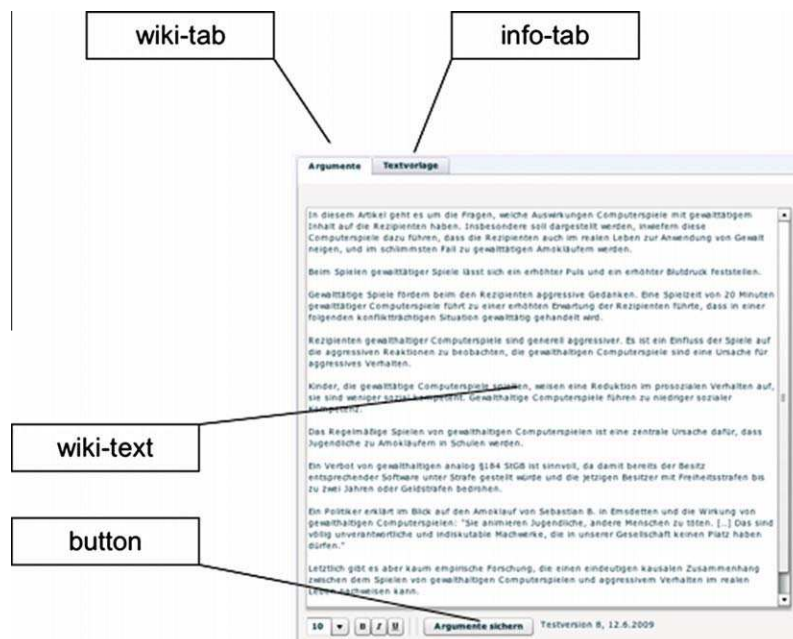


Fig. 1. Screenshot of the used writing software.

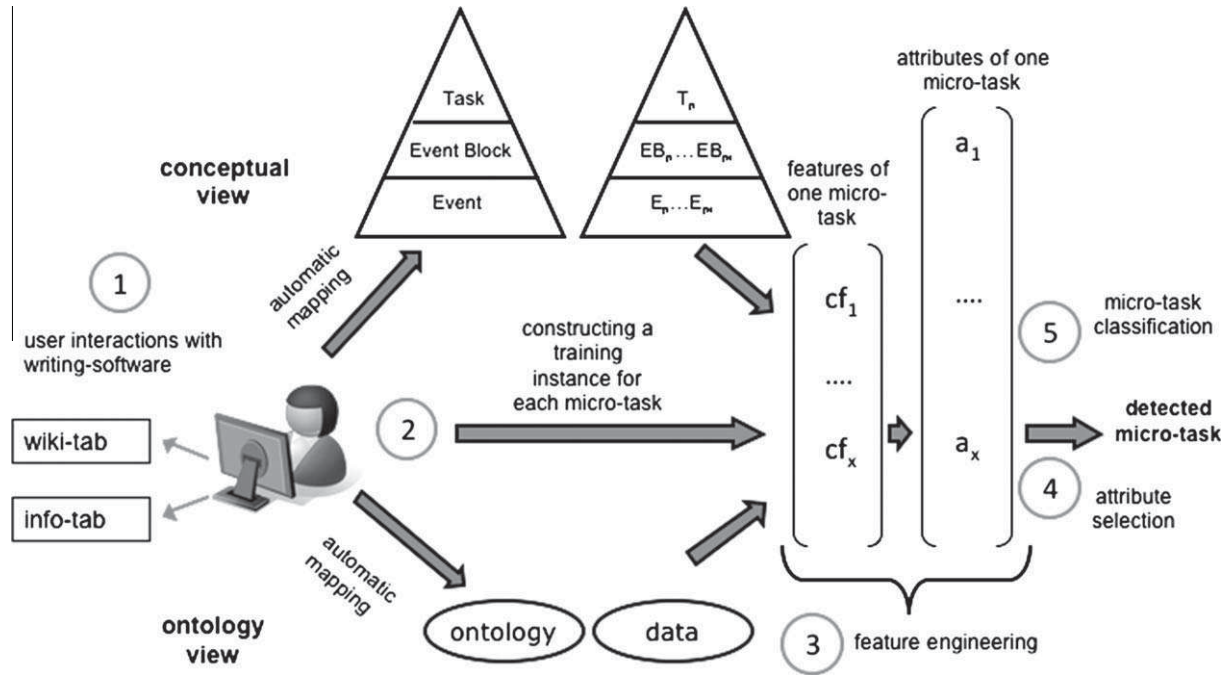


Fig. 2. This figure visualizes the complete *user interaction context ontology task detection pipeline* (UICO pipeline) starting from (1) the automatic unobtrusive user interaction observation mechanisms to (4) detecting the user's task. The automatic population of the user interaction context model (2) is displayed in two ways (i) the instantiation of entities in the conceptual model (*conceptual view*) and (ii) in the ontology model (*ontology view*). In (3) the feature engineering process for transforming a task instance into a training/testing instance is shown. The training/testing instance is further fed to attribute selection (4) and learning algorithms for detecting the task (5).

the wiki-text before and after the participants' edits and highlighted the changes. The experts were familiar with the arguments of the texts ("pros and cons" of violent computer games), as they had been involved in the development of the wiki-text and the additional information. This enabled them to decide which changes in the wiki-text would be necessary to obtain a scientifically balanced article about violent computer games and their risks to users and society. The expert ratings were used to categorize each micro-task as weak, medium, or strong accommodation and obtain three classes with a similar number of tasks. Weak accommodation occurred, for example, if participants only copied an argument from the text on the info-tab and pasted it into the wiki on the wiki-tab without modifying it or integrating it into the rest of the text. Strong accommodation occurred if they added an argument from the text on the info-tab, but integrated it into the rest of the text on the wiki-tab by adding sentences like "This argument is refuted by results from a study that examines ...", or if they drew a conclusion like "To sum up the contradicting results ...", or if they rearranged arguments to guide the reader through the text.

This research setting yielded data from different participants, who performed similar micro-tasks during the writing process. We used these data to select interaction-based features in order to detect accommodation steps, that is, to find attributes of a micro-task that classified it as accommodation. The task detection system observed the users' writing process and their interaction with the wiki-tab and the info-tab. The experts' classification of the micro-task was the basis of training the task-detection system to detect accommodation steps automatically. In the following subsection, we will describe our applied ontology-based task detection approach to solve this micro-task classification problem.

We also calculated reading scores in order to differentiate knowledge maturing from other common text measurements. We applied the *Flesch Reading Ease test* (Si and Callan, 2001), the *Gunning fog index* (Gunning, 2004), and the *Flesch-Kincaid readability test* (Flesch, 1948) to each micro-task. These scores are all based on quantitative metrics like sentence length, number of syllables,

or number of words. We computed the reading scores on the state of the document before and after the micro-task. For each reading score, we subtracted the respective resulting values.

3.2. Method

We solved the micro-task classification problem on the basis of five steps, which are illustrated by Fig. 2 and explained in the following: (1) User interactions with the writing software were captured by system and application sensors. (2) Parts of these data were chosen as features to build classification training/testing instances at the micro-task level. (3) To obtain valid inputs for the machine learning algorithms, these features were first transformed into attributes. (4) Attribute selection was performed to select the best discriminative attributes. (5) Finally, classification/learning algorithms were trained and tested.

3.2.1. Capturing users' interactions with the writing software

The first step of the task detection process consisted of capturing the *user interaction context*, which described all interactions between users and the wiki (e.g., click on wiki-tab/info-tab, typing word, deleting phrases, etc.). For that purpose we employed context sensors that have already been applied in previous user interaction context observations (Rath et al., 2008) and in task detection studies (Rath et al., 2009a,b). We developed new sensors for Macromedia Flash, which is the base technology of the writing software. Additional fine-granular user interaction context information included (1) switch to info-tab, (2) switch to wiki-tab, (3) text formatting, (4) text editing, (5) text selection, and (6) selection of a specific argument in the info-tab. The content of the wiki and the text around the cursor were also recorded for each single user interaction.

3.2.2. Storing users' interactions via an ontology-based context model

In order to store the captured users' interaction data in a machine processable form, we utilized the user interaction context

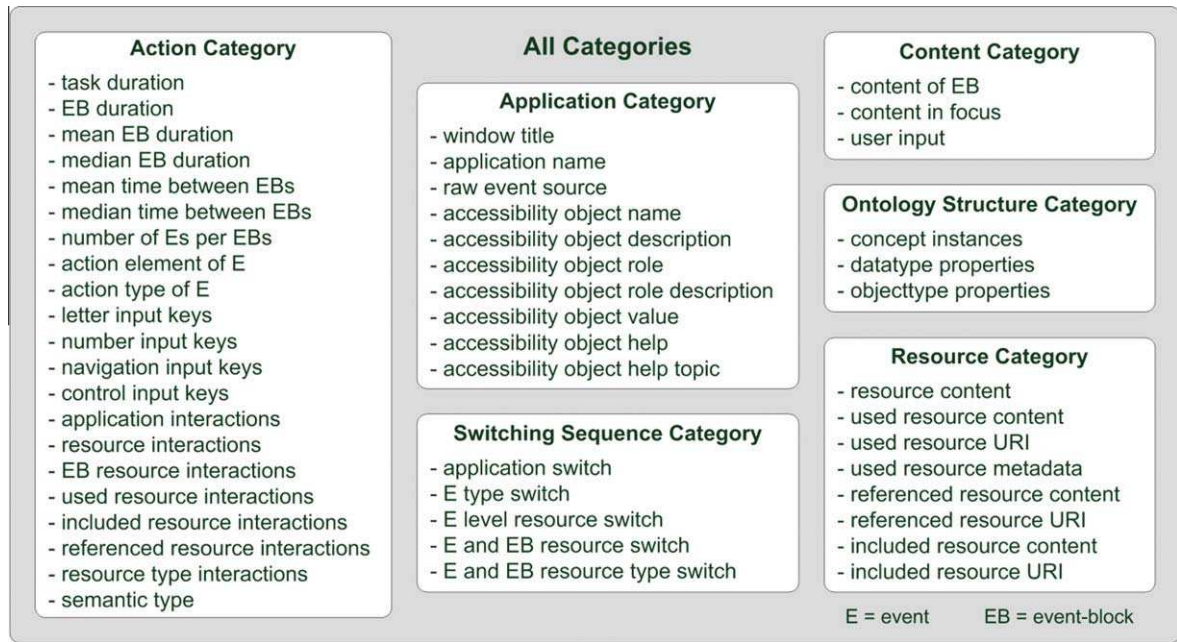


Fig. 3. This figure shows all the 50 features classified into six feature categories. All the features were extracted based on the user interaction context ontology (UICO).

ontology (UICO) developed by the Rath et al. (2009a), which contains 88 concepts and 272 properties, and is modeled in OWL-DL (the Ontology Web Language, a W3C standard). From a high-level perspective, the concepts of this ontology can be grouped into five different dimensions: action dimension, resource dimension, user dimension, information need dimension, and application dimension. This study specifically focused on the action dimension (user interactions with the writing software), and the resource dimension (the text components) of the ontology. The wiki context sensors provided information on which elements of the wiki-tab or info-tab a user was working on. This information was used to construct resources (in this case the wiki-tab, the info-tab, or one of the ten arguments in the info-tab) in the ontology with a unique URI.

The low-level data that constituted the stream of events resulting from the user's interactions with the writing software was progressively aggregated, and used as a basis for populating the UICO, that is, instantiating its concepts and creating relations between the concept instances. The *Event* concept was directly instantiated, based on the captured user interaction context data. In order to instantiate the *EventBlock* concept, *Events* were first aggregated into *EventBlocks*, using generic static rules and heuristics. Each *EventBlock* connected a user's actions, which were associated with one specific resource (a tab or argument). As the individual participants of the study had to save their changes after each set of alterations that "belonged together" in their opinion, we could use this information to aggregate the *EventBlocks* into micro-tasks. To create the relations between concept instances, we employed regular expressions, information extraction, as well as application and resource-specific algorithms.

3.2.3. Engineering features and attributes from the user's interactions

To detect accommodation steps, we had to identify features that could classify a micro-task as a weak, medium, or strong accommodation. For this purpose, we engineered 50 features based on the concepts and relations of the UICO, which we tested for their discriminative power. These features were grouped into six categories (see Fig. 3): action, application, switching sequences, content, ontology structure, and resource. The *action category* represented

user interactions and contained features about interactions with applications (the writing software and the web browser in which it was embedded), resource types, resources (tabs and arguments), key input types (navigational keys, letters, numbers), the number of events and event blocks, duration of the event blocks, and time intervals between event blocks. The *application category* contained the classical window title feature (Oliver et al., 2006; Rath et al., 2009a), the application name feature (Granitzer et al., 2009), and graphical user interface elements (*accessibility objects*¹) features. The *switching sequences category* comprised features about switches between applications, resources, as well as event and resource types. The *content category* consisted of the content of task-related resources, the content in focus (text around the cursor), and the user's text input. The *ontology structure category* contained features representing the number of instances of concepts and the number of data type and object type relations used per task. The *resource category* included the complete contents and URIs (URLs) (Shen et al., 2009) of the used, referenced and included resources, as well as a feature that combines all the metadata about the used resources in a 'bag of words'. All categories exploited the relationships between events, event-blocks, and resources that were computed and stored in the UICO.

Using the machine learning toolkit Weka (Witten and Frank, 2005), features were then transformed into attributes that feed the classification algorithms. The following steps were performed to preprocess the content of text-based features (in this sequence): (1) remove end of line characters, (2) remove markups, for example, `
` and `!<CDATA`, (3) remove all characters except letters, (4) remove German and English stop words, (5) remove words shorter than three characters. We transformed text-based features into vectors of words with the *StringToWordVector* function of Weka. For numeric features, we applied the Weka *PKIDiscretize* filter to replace discrete values by intervals.

3.2.4. Classifying micro-tasks from user interactions

The last step of our method consisted of classifying the micro-tasks, based on the features engineered from the populated UICO.

¹ Microsoft Active Accessibility: <http://msdn.microsoft.com/en-us/accessibility/>.

For the purposes of this study, a new training/testing instance was built for each set of alterations that belong together (i.e., each micro-task), as defined by the participants. The training/testing instances were given to machine learning algorithms to train/build a classification model and then decide to which class another training/testing instance belongs. The classes were “weak”, “medium”, and “strong” accommodation.

4. Results

The participants were ten graduate students from a German university, their mean age was 25.30 ($SD = 0.51$). Four of these were women, six were men. During the wiki study, we recorded a dataset of 158 micro-tasks. Each participant saved on average 15.8 ($SD = 7.29$, $min = 7$, $max = 26$) micro-tasks.

4.1. Expert ratings and reading scores

Two experts rated each micro-task with regard to the occurrence of accommodation processes. Their ratings correlated significantly, $r(156) = .80$, $p < .01$, which showed the reliability of the expert ratings. We then calculated the correlation between the reading-scores of each transformation and the average of the accommodation ratings of the two experts. Since the used variables showed no normal distributions, we used rank order correlations. The results revealed no significant correlations. This supported our assumption that reading scores were not appropriate to measure accommodation steps within a wiki text.

4.2. Accommodation classes and training/testing instances

For each micro-task, a new training/testing instance for the machine-learning algorithm was built, based on the recorded usage data. During the study we recorded a dataset of 158 micro-tasks. For 19 of the 158 micro-tasks the log files showed no difference between the two text versions, so they were excluded from further analysis. Based on the average accommodation ratings of the content experts, we divided the remaining 139 micro-tasks into tertiles. This was the basis for constructing three classes; we computed the boundaries of the classes with $weak \leq 1.5$; $medium \leq 2.5$; $strong > 2.5$ (based on the average accommodation ratings of the two experts on a five-point Likert scale). This led to three classes representing different levels of accommodation: weak accommodation (30 instances), medium accommodation (55 instances), and strong accommodation (54 instances). In order to evaluate factors that influenced this classification issue, we varied the following parameters: (1) the learning algorithm, (2) the set of used features, and (3) the number of attributes generated from the features. Furthermore, the set of features used was varied by including (1) each feature individually, (2) each feature category individually, (3) all feature categories, and (iv) the *top k* best performing single features, with $k \in \{2, 3, 4, 5, 6, 7, 8, 9, 10, 15, 20\}$.

We studied the following well-known text classifiers: Naive Bayes (NB), Linear Support Vector Machine (SVM) with cost parameter $c \in \{2^{-5}, 2^{-3}, 2^{-1}, 2^0, 2^1, 2^3, 2^5, 2^8, 2^{10}\}$,² J48 decision tree (J48), and *k*-Nearest Neighbor (KNN-*k*) with $k \in \{1, 5, 10, 35\}$. To build a task instance, we selected the *g* attributes having the highest Information Gain value, in which *g* was varied among 50 different values distributed between 3 and 10,000. We measured the accuracy of the algorithms used, the micro precision, and the micro recall (Witten and Frank, 2005).

4.3. Results of the classification

Table 1 shows the best results obtained while detecting micro-tasks as weak, medium, and strong accommodation steps by stratified 10-fold cross-validation for each feature category, for each single feature, and for the *k* top performing single features. The evaluation results showed that a combination of four UICO features achieved an accuracy of 79.12% with the Naive Bayes algorithm for detecting weak, medium, and strong accommodation tasks. In comparison, the probability of random guesses of whether a micro-task belongs to the weak, medium, or strong accommodation class was 39.57% on our dataset. This means that applying the ontology-based task detection approach improved accuracy considerably. A detailed discussion of the features will be provided in the following paragraphs.

4.3.1. Feature categories

The best performing feature category was the *content category* with an accuracy of 77.03% and the NB algorithm ($g = 175$, $p = .86$, $r = .77$). The combination of all 50 features was closely behind with an accuracy of 74.12% with the same algorithm, but required 1500 attributes ($p = .84$, $r = .72$). The *action category* was third in the category ranking with an accuracy of 69.07% with KNN-10 algorithm and with 25 attributes ($p = .80$, $r = .66$). The difference between the best and the worst accuracy values was 23.68.

4.3.2. Single features

The best performing feature was the *content in focus feature*, with an accuracy of 74.07% with the NB algorithm and with 75 attributes ($p = .84$, $r = .72$). This feature was constructed based on the text the user had interacted with, and it represented a term vector from which all stop words were removed. This feature belonged to the content category, which was the best performing category with 77.03%. The second best performing feature with an accuracy of 70.44% was the *control input keys feature* with only one attribute and the SVM algorithm ($p = .81$, $r = .66$). This attribute represents the number of times a control key, for example, SHIFT, RETURN or INSERT, had been pressed by the user during the editing tasks. The distribution of values for the attribute showed that control keys were less used for weak accommodation steps than for strong accommodation steps. The values for medium accommodation steps were distributed in a balanced way. The *user input feature* ranked at third place among the best performing single features with an accuracy of 69.89% with the KNN-5 algorithm and with 10 attributes ($p = .80$, $r = .66$). This feature contained the content the user produced by keyboard input or copy and paste combinations.

4.3.3. Top *k* features

The best combination of the best performing single features was the top $k = 4$ feature combination with an accuracy of 79.12% with the NB algorithm and with 100 attributes ($p = .88$, $r = .79$). The top 4 features were the *content in focus feature*, the *control input keys feature*, the *user input feature*, the *resource interaction feature*. The resource interaction feature represented for each resource (a tab or an argument) the number of interactions of the user with that resource. The third best combination was the top $k = 3$ combination that only utilized the best three performing single features, but also achieved a high accuracy value of 78.53% ($l = NB$, $g = 100$, $p = .86$, $r = .76$). The top $k = 4$ and the top $k = 3$ performed almost equally in terms of accuracy with only 0.59% difference. All top *k* feature combinations, except the worst one, outperformed each single feature, each feature category, and the combination of all 50 features. This shows that not all features were helpful in the classification decision, and that only a small set of features was required to achieve a high accuracy in detecting a single micro-task

² Values were chosen according to the libSVM guide: <http://www.csie.ntu.edu.tw/~cjlin/libsvm/>.

Table 1

Overview of the best accuracies (a) obtained while detecting tasks with weak, medium, and strong accommodation by stratified 10-fold cross-validation for each feature category, for each single feature as well as for the k top performing single features (f). Also provided are the learning algorithm (l), the number of attributes (g), the micro precision (p), the micro recall (r), and the ranking in the corresponding section (R_s) and across sections (R_G).

Set	R_s	f	l	g	Q	P	r	R_G
Feature categories	1	Content Cat.	NB	175	77.03	0.86	0.77	8
	2	All Categories	NB	1500	74.12	0.84	0.72	11
	3	Action Cat.	KNN-10	25	69.07	0.80	0.66	18
	4	Resource Cat.	SVM-C = 2^1	3000	68.41	0.77	0.60	21
	5	Ontology Str. Cat.	SVM-C = 2^5	5	64.12	0.74	0.57	38
	6	Switching Seq. Cat.	KNN-5	10	64.07	0.77	0.60	40
	7	Application Cat.	NB	10	53.35	0.65	0.50	48
Single features	1	Content in focus	NB	75	74.07	0.84	0.72	13
	2	Control input keys	SVM-C = 2^8	1	70.44	0.81	0.66	15
	3	User input	KNN-5	111	69.89	0.80	0.66	16
	4	Res. interact.	J48	175	69.78	0.80	0.67	17
	5	Used res. interact.	J48	5	69.01	0.78	0.63	19
	6	Resource content	SVM-C = 2^{-2}	75	69.01	0.78	0.61	20
	7	UICO concept	J48	104	68.35	0.79	0.64	22
	8	Referenced res. interact.	SVM-C = 2^1	298	67.69	0.78	0.62	23
	9	E type switch seq. 2	J48	10	67.69	0.78	0.61	24
	10	Res. types interact.	J48	14	67.58	0.78	0.62	25
	11	Used res. metadata	SVM-C = 2^{10}	1000	66.98	0.76	0.59	26
	12	EB duration	J48	9	66.21	0.77	0.60	27
	13	Applications interact.	SVM-C = 2^{-1}	3	66.15	0.75	0.57	28
	14	Included res. interact.	SVM-C = 2^{-1}	75	66.04	0.75	0.58	29
	15	UICO object type rel.	J48	57	65.60	0.77	0.63	30
	16	No. of E/EB	J48	10	65.49	0.76	0.61	31
	17	Navigation input keys	NB	1	64.89	0.74	0.58	32
	18	Included res. content	SVM-C = 2^5	50	64.78	0.77	0.63	33
	19	EB res. interact.	SVM-C = 2^{-1}	125	64.78	0.74	0.55	34
	20	Referenced res. content	SVM-C = 2^{-5}	50	64.73	0.77	0.63	35
Top k features	1	Top $k = 4$	NB	100	79.12	0.88	0.79	1
	2	Top $k = 10$	NB	200	78.52	0.87	0.79	2
	3	Top $k = 3$	NB	100	78.52	0.86	0.76	3
	4	Top $k = 8$	NB	175	78.30	0.87	0.78	4
	5	Top $k = 6$	NB	150	77.69	0.86	0.77	5
	6	Top $k = 7$	NB	150	77.64	0.86	0.77	6
	7	Top $k = 9$	NB	175	77.09	0.86	0.77	7
	8	Top $k = 5$	NB	125	76.21	0.84	0.75	9
	9	Top $k = 2$	NB	125	75.55	0.85	0.73	10
	10	Top $k = 20$	NB	750	74.07	0.84	0.73	12
	11	Top $k = 15$	NB	1000	73.46	0.84	0.73	14

as accommodation step. For detecting accommodation classes, the top 4 features achieving the highest accuracy highlight that the “content” viewed or contributed by the user during a transformation plays a key role.

4.4. Comparison with other task detection approaches

The evaluation results showed that we achieved an accuracy of 79.12% and a precision of .86 for classifying micro-tasks into accommodation classes. This is a remarkably positive result regarding the fine granularity of a micro-task. We aimed at detecting accommodation steps during a well-defined writing process and classify them as weak, medium or strong. In the current study, the duration of a micro-task was short, and the user’s interactions involved only two applications (the web browser and the Flash writing software embedded in it). Tasks involved in other task detection studies had a coarser granularity. Examples are “buying a book” (Oliver et al., 2006), or scientific research tasks (Granitzer et al., 2009). Existing task detection approaches focus on higher level tasks, but report similar performances: an accuracy of 76% with a precision of .49 (Oliver et al., 2006) an accuracy of 74.51% with a precision of .91 (five classes) and an accuracy of 76.42% with a precision of .90 (four classes) (Granitzer et al., 2009), and a precision of .80 (96 and 81 classes) (Shen et al., 2009).

The most popular features which have been identified as having a high discriminative power among other tasks are the *window title feature* (Oliver et al., 2006; Granitzer et al., 2009), the *file path/web*

page URL (Shen et al., 2009), and the *content in focus feature* (Granitzer et al., 2009). The task detection results on our dataset showed that in our data the *window title* and the *file path/web page URL* features did not perform well. This is attributable to the fine-granularity of the tasks that we were aiming to detect, which involved only two applications. In terms of attributes used for training the machine learning algorithms, an interval of 200–300 attributes is suggested to be sufficient (Shen et al., 2009; Granitzer et al., 2009). Our results support the assumption that only a small ratio of attributes is required to detect a specific micro-task successfully as accommodation step. Contrary to the study by the Granitzer et al. (2009), with our datasets the SVM and the KNN classifiers were outperformed by the Naive Bayes algorithm.

To sum up our results, complementing previous studies, we have been able to demonstrate that task detection approaches may also be useful for classifying very fine-granular application-specific tasks, such as accommodation during a writing process. We were able to detect accommodation processes by utilizing the high discriminative features for accommodation, which we found during our evaluation, in order to train a classifier that can classify single micro-tasks with a high degree of accuracy.

5. Conclusion

In the research presented here, we adopted the task detection approach to analyze the writing process within a wiki text. Based

on the ratings of two experts, we were able to distinguish between three classes of accommodation. The ontology-based task detection approach finally yielded an identification rate of 79.12%. Our study was the first step in the development of methods for the automatic detection of accommodation steps as indicators for knowledge maturing. The results of our study showed that it was indeed possible to detect accommodation during the writing process, using only a limited amount of contextual information. This allowed an efficient detection of knowledge maturing on the basis of user behavior.

The co-evolution model (and its system-theoretical background) states that cognitive conflicts within a community will trigger innovation and knowledge maturing. So additional steps for future work will have to take into account discursive processes between two or more users, using the wiki for their collaboration. The same method might be used for detecting accommodation for collaborative writing processes. This could be done by an experiment in which participants with different prior knowledge about a given topic have to work in turns on the same text. The diversity of the participants' prior knowledge is then supposed to cause cognitive conflicts, which would enable discursive processes between them (via the collaborative writing task), and support knowledge maturing.

For further applications, our findings could lead to tools that might guide the users' writing process. Research on writing processes shows that feedback on the quality and readability of text may help writers to improve their own texts (Hayes and Flower, 1986). Using the automatically detected accommodation steps as feedback about one's own writing process may improve the text which a user has written, and enhance knowledge maturing. This would expand what has been found in some related work on knowledge-maturing support (Schoefegger et al., 2009), using reading-scores (Braun and Schmidt, 2007) as feedback to users about their own wiki texts. Our method takes into account the writing process itself, and analyzes the process of accommodation. This is an important aspect in the larger context of social media, collaborative learning environments (Trentin, 2009; Forte and Bruckman, 2010) and knowledge management, because the study provided some insights into those processes that lead to accommodation. The collaboratively written text, as an epistemic artifact (Scardamalia and Bereiter, 2006), is supposed to support discursive processes, and we think it is necessary to focus on collaborative writing as a tool for knowledge maturing.

Our automatically performed detection of accommodation steps might be relevant to further studies that may identify possible support mechanisms to improve and/or encourage accommodation (towards supporting knowledge maturing). It is possible to implement the task detection tool as a MediaWiki extension and to apply the task-detection method to real-world settings with real communities working on wikis. It is conceivable to observe writing processes automatically in real wikis, as we did in the wiki editor used in the present study.

In addition to the online detection of accommodation steps, a post hoc analysis of revision histories of collaboratively written text would be of great interest. This would allow to analyze existing text corpora on the Internet, for example, of Wikipedia, and to use the task-detection tool in experiments under laboratory conditions. On the whole, we hope that our previous findings and our considerations concerning potential future studies will stimulate other researchers to initiate corresponding research of their own, and that this will shed some light on understanding processes of knowledge maturing.

References

Baldauf, M., Dustdar, S., Rosenberg, T., 2007. A survey on context-aware systems. *Int. J. Ad Hoc Ubiqu. Co.* 2, 263–277.

- Braun, S., Schmidt, A., 2007. Wikis as a technology fostering knowledge maturing: what we can learn from Wikipedia. In: *Proceedings of the 7th International Conference on Knowledge Management (IKNOW '07), Special Track on Integrating Working and Learning in Business (IWL)*, pp. 321–329.
- Coutaz, J., Crowley, J.L., Dobson, S., Garlan, D., 2005. Context is key. *Commun. ACM* 48, 49–53.
- Cress, U., Kimmmerle, J., 2008. A systemic and cognitive view on collaborative knowledge building. *Int. J. Comput.-Supp. Collab. Lear.* 3, 105–122.
- Dey, A.K., Abowd, G.D., Salber, D., 2001. A conceptual framework and a toolkit for supporting the rapid prototyping of context-aware applications. *Hum.-Comput. Interact.* 16, 97–166.
- Dragunov, A.N., Dietterich, T.G., Johnsrude, K., McLaughlin, M., Li, L., Herlocker, J.L., 2005. TaskTracer: a desktop environment to support multi-tasking knowledge workers. In: *Proceedings of the 10th International Conference on Intelligent User Interfaces (IUI '05)*. ACM, New York, pp. 75–82.
- Erkens, G., Jaspers, J., Prangma, M., Kanselaar, G., 2005. Coordination processes in computer supported collaborative writing. *Comput. Hum. Behav.* 21, 463–486.
- Flesch, R., 1948. A new readability yardstick. *J. Appl. Psychol.* 32, 221–233.
- Forte, A., Bruckman, A., 2010. Citing, writing and participatory media: wikis as learning environments in the high school classroom. *Int. J. Learn. Media* 1, 23–44.
- Graesser, A., McNamara, D., Louwerse, M., Cai, Z., 2004. Coh-Metrix: analysis of text on cohesion and language. *Behav. Res. Methods* 36, 193–202.
- Granitzer, M., Rath, A.S., Kroell, M., Seifert, C., Ipsmiller, D., Devaurs, D., Weber, N., Lindstaedt, S., 2009. Machine learning based work task classification. *J. Digit. Inf. Manag.* 7, 306–314.
- Gunning, R., 2004. Plain language at work newsletter. <<http://www.impact-information.com/impactinfo/newsletter/plwork08.htm>>.
- Gutschmidt, A., Cap, C.H., Nerdinger, F.W., 2008. Paving the path to automatic user task identification. In: *Workshop on Common Sense Knowledge and Goal-Oriented Interfaces, (IUI '08)*. <<http://ftp.informatik.rwth-aachen.de/Publications/CEUR-WS/Vol-323/paper03.pdf>>.
- Hayes, J.R., Flower, L.S., 1986. Writing research and the writer. *Am. Psychol.* 41, 1106–1113.
- Hu, M., Lim, E.P., Sun, A., Lauw, H.W., Vuong, B.Q., 2007. Measuring article quality in Wikipedia: models and evaluation. In: *Proceedings of the 16th ACM Conference on Information and Knowledge Management*. ACM, New York, NY, pp. 243–252.
- Johnson, S., 2001. *Emergence: The Connected Lives of Ants, Brains, Cities, and Software*. Scribner, New York.
- Kimmmerle, J., Cress, U., Held, C., 2010a. The interplay between individual and collective knowledge: technologies for organisational learning and knowledge building. *Knowl. Manag. Res. Pract.* 8, 33–44.
- Kimmmerle, J., Cress, U., Held, C., Moskaliuk, J., 2010b. Social software and knowledge building: Supporting co-evolution of individual and collective knowledge. In: Gomez, K., Lyons, L., Radinsky, J. (Eds.), *Learning in the Disciplines: Proceedings of the 9th International Conference of the Learning Sciences (ICLS 2010)*, vol. 1, Full Papers. International Society of the Learning Sciences, Chicago, IL, pp. 9–16.
- Kimmmerle, J., Cress, U., Hesse, F.W., 2007. An interactional perspective on group awareness: alleviating the information exchange dilemma (for everybody?). *Int. J. Hum.-Comput. Struc.* 65, 899–910.
- Kimmmerle, J., Moskaliuk, J., Cress, U., in press. Using wikis for learning and knowledge building: Results of an experimental study. *Educ. Techn. Soc.*
- Kimmmerle, J., Moskaliuk, J., Cress, U., Thiel, A., 2011. A systems theoretical approach to online knowledge building. *AI & Soc.* 26, 49–60.
- Kimmmerle, J., Moskaliuk, J., Harrer, A., Cress, U., 2010c. Visualizing co-evolution of individual and collective knowledge. *Inf. Comm. Soc.* 13, 1099–1121.
- Kimmmerle, J., Wodzicki, K., Cress, U., 2008. The social psychology of knowledge management. *Team Perform. Manag.* 14, 381–401.
- Klare, G.R., 1974. Assessing readability. *Read. Res. Quart.* 10, 62–102.
- Kramer, M., Gregorowicz, A., Iyer, B., 2008. Wiki trust metrics based on phrasal analysis. In: *Proceedings of the 2008 International Symposium on Wikis (ACM WikiSym '08)*, Porto, Portugal. <<http://wikisym.org/ws2008/proceedings/research%20papers/18500017.pdf>>.
- Landauer, T.K., Dumais, S.T., 1997. A solution to Plato's problem: the latent semantic analysis theory of the acquisition, induction, and representation of knowledge. *Psychol. Rev.* 104, 211–240.
- Lave, J., Wenger, E., 1991. *Situated Learning: Legitimate Peripheral Participation*. Cambridge University Press, New York.
- Luhmann, N., 1995. *Social Systems*. Stanford University Press, Stanford.
- Luhmann, N., 2006. System as difference. *Organization* 13, 37–57.
- Moskaliuk, J., Kimmmerle, J., 2009. Using wikis for organizational learning: functional and psycho-social principles. *Dev. Learn. Organ.* 23, 21–24.
- Moskaliuk, J., Kimmmerle, J., Cress, U., 2009. Wiki-supported learning and knowledge building: effects of incongruity between knowledge and information. *J. Comput. Assist. Lear.* 25, 549–561.
- Oliver, N., Smith, G., Thakkar, C., Surendran, A.C., 2006. SWISH: semantic analysis of window titles and switching history. In: *Proceedings of the 11th International Conference on Intelligent User Interfaces (IUI '06)*. New York, ACM, pp. 194–201.
- Piaget, J., 1977a. *The Development of Thought: Equilibration of Cognitive Structures*. Viking, New York.
- Piaget, J., 1977b. Problems of equilibration. In: Appel, M.H., Goldberg, L.S. (Eds.), *Topics in Cognitive Development*, vol. 1. Plenum, New York, pp. 3–14.
- Raitman, R., Augar, N., Zhou, W., 2005. Employing wikis for online collaboration in the e-learning environment: case study. In: *Proceedings of the 3rd International Conference on Information Technology and Applications (ICITA '05)*, Sydney, Australia. ACM, New York, pp. 142–146.

- Rath, A.S., Devaurs, D., Lindstaedt, S.N., 2009a. UICO: An ontology-based user interaction context model for automatic task detection on the computer desktop. In: Proceedings of the 1st Workshop on Context, Information and Ontologies (CIAO '09), Workshop as part of the ESWC '09, Heraklion, Greece. ACM, New York, pp. 1-10.
- Rath, A.S., Devaurs, D., Lindstaedt, S.N., 2009b. Detecting real user tasks by training on laboratory. In: Proceedings of Exploitation User Attention Metadata (EUAM '09), Lecture Notes in Informatics. <http://know-center.at/download_external/papers/arath_et_al_euam2009.pdf>.
- Rath, A.S., Devaurs, D., Lindstaedt, S.N., 2010. Studying the factors influencing automatic user task detection on the computer desktop. In: Wolpers, M., Kirschner, P.A., Scheffel, M., Lindstaedt, S., Dimitrova, V. (Eds.), Sustaining TEL: From Innovation to Learning and Practice, Proceedings of EC-TEL 2010, pp. 292-307.
- Rath, A.S., Weber, N., Kroell, M., Granitzer, M., Dietzel, O., Lindstaedt, S.N., 2008. Context-aware knowledge services. Workshop on Personal Information Management (PIM2008) at the 26th Computer Human Interaction Conference (CHI '08), Florence, Italy. <http://www.know-center.tugraz.at/download_external/papers/pim2008-rath-et-al.pdf>.
- Reinhold, S., 2006. Wikitrails: augmenting wiki structure for collaborative, interdisciplinary learning. In: Proceedings of the International Symposium on Wikis 2006 (WikiSym '06). ACM, New York, pp. 47-58.
- Scardamalia, M., Bereiter, C., 2006. Knowledge building: theory, pedagogy, and technology. In: Sawyer, K. (Ed.), The Cambridge Handbook of the Learning Sciences. Cambridge University Press, New York, pp. 97-115.
- Schmidt, A., 2005. Knowledge maturing and the continuity of context as a unifying concept for knowledge management and e-learning. In: Proceedings of the 5th International Conference on Knowledge Management (I-KNOW '05), Graz, Austria, pp. 424-431.
- Schoefegger, K., Weber, N., Lindstaedt, S., Ley, T., 2009. Knowledge maturing services: supporting knowledge maturing in organisational environments. In: Karagiannis, D., Jinpeng, Z. (Eds.), Knowledge Science, Engineering and Management, 3rd International Conference, KSEM 2009. Springer, Berlin, pp. 370-381.
- Shen, J., Irvine, J., Bao, X., Goodman, M., Kolibaba, S., Tran, A., Carl, F., Kirschner, B., Stumpf, S., Dietterich, T.G., 2009. Detecting and correcting user activity switches: algorithms and interfaces. In: Proceedings of the 13th International Conference on Intelligent User Interfaces (IUI '09). ACM, New York, pp. 117-126.
- Si, L., Callan, A.J., 2001. Statistical model for scientific readability. In: Proceedings of CIKM '01, Atlanta, Georgia, USA. ACM, New York, pp. 574-576.
- Strang, T., Linnhoff-Popien, C., 2004. A context modeling survey. In: Workshop on Advanced Context Modeling, Reasoning and Management, UbiComp, pp. 1-8.
- Stvilia, B., Twidale, M.B., Gasser, L., Smith, L.C. (2005). Information quality in a community-based encyclopedia. In: Proceedings of the International Conference on Knowledge Management (ICKM '05), pp. 101-113.
- Trentin, G., 2009. Using a wiki to evaluate individual contribution to a collaborative learning project. J. Comput. Assist. Lear. 25, 43-55.
- Vygotsky, L.S., Cole, M., John-Steiner, V., Scibner, S., Souberman, E., 1978. Mind in Society: The Development of Higher Psychological Processes. Harvard University Press, Cambridge.
- Witten, I.H., Frank, E., 2005. Data Mining: Practical Machine Learning Tools and Techniques. Morgan Kaufmann, San Francisco.
- Wöhner, T., Peters, R., 2009. Assessing the quality of Wikipedia articles with lifecycle based metrics. In: Proceedings of the 5th International Symposium on Wikis and Open Collaboration (ACM WikiSym '09), Orlando, Florida. <<http://www.wikisym.org/ws2009/procfiles/p116-woehner.pdf>>.