Journal of Universal Computer Science, vol. 17, no. 10 (2011), 1511-1526
submitted: 30/10/10, accepted: 15/4/11, appeared: 1/6/11 © *J.UCS*

Task Models for Intention-Aware Systems

Benedikt Schmidt and Todor Stoitsev
(SAP Research CEC Darmstadt, Germany
benedikt.schmidt@sap.com, todor.stoitsev@sap.com)

Max Mühlhäuser
(TU Darmstadt, Germany
max@informatik.tu-darmstadt.de)

Abstract: Intention-aware systems integrate aspects of context-aware and attention-aware systems for the identification and support of intention. Focusing intention is justified by the impact of intention on awareness and interaction with the world. Therefore, proactive user support mechanisms can be improved by including a representation of intention. In this paper, the externalization of intention in task models is discussed: existing task models are reviewed and activity schemes are proposed as task model for intention-aware systems. A framework for intention-aware systems is presented and discussed in detail.

Key Words: intention-aware systems, context-aware systems, attention-aware systems, knowledge work
Category: H.4.1, H.1.2

1 Introduction

Contingent work execution processes complicate knowledge work support. Proactive user support - realized as recommender systems, adaptive user interfaces or assistants - presents itself as powerful mechanism to support knowledge work. One foundation for proactive user support is given by two complementary approaches: context-aware systems [Baldauf et al. 2007] and attention-aware systems [Roda and Thomas 2006]. The respective realizations mainly focus on the detection of user status (attention-aware) or environment status (context-aware). We see potential in utilizing the information of both system types, integrated by a task model. The task model needs to explicate the individual and implicit intentions and plans of users, to reason about attention and context information. This integration enables work execution support reflecting situational intentions and plans. We call such systems intention-aware.

In this paper we discuss intention-aware systems to connect context and attention data with user intention, using a task model. Initially, we conceptualize intention-aware systems by providing a human-environment interaction model (sec. 2). We focus on intention-aware systems in the domain of desktop computing and discuss the applicability of used task models to enable intention-awareness. We review different task models, e.g. used in context-aware and attention-aware applications (sec. 3). The review focuses on the task models already applied in such systems, the population of such models with instance information and the systems' purpose. All systems work on the same information base, the tracking of user-system interaction. We show a connection between the richness of the task model with respect to the human-environment interaction model. Finally, we propose activity schemes as task model for intention-aware systems (sec. 4) and use it in a framework for intention-aware systems (sec. 5).

2 Towards Intention-Aware Systems

Intention is "a composite concept specifying what the agent has chosen and how the agent is committed to that choice" [Cohen and Levesque 1990]. The statement highlights intention as something individual, only existing implicitly, as it is highly connected with an individuals' goal-directed perception of the environment. This environment is the locus of human-world interaction triggered by the commitment that results from intention. The structure of intention as organizing goals and their achievement by executing plans has been tackled by artificial intelligence research in a myriad of approaches [Cohen and Levesque 1990]. Still, it remains a difficult task to model, detect, and process the necessary information about users and environment to actually detect intention.

Recently, user and environment information have been tackled by context-aware and attention-aware systems. Both share common ground in the detection and externalization of status information and both make use of instrumented environments. Nevertheless, they stress different aspects. Context-aware systems focus on detection of situation-specific environmental features [Baldauf et al. 2007], whereas attention aware systems focus on situation-specific individual processes of perception and cognition [Roda and Thomas 2006].

An intention-aware system integrates these aspects. It identifies user intention based on the situation-specific user attention and the status of the environment and supports the user based on plans associated with the intention. In the following the influence of intention on individual context-awareness is examined, using a systemic-constructivist model.

2.1 Human-Environment Interaction Model

To model human-environment interaction we extend the K-system model which describes system-world interaction by means of a control circuit (see figure 1) [Stachowiak 1973] . Considering the K-system as a human being, the human is organized by perceptor, operator, and effector interrelated with the environment and cognitive processes tackling reasoned action. We extend the model in two directions. On the one hand, we specify the motivator as the connection of intention, attention, and planning. This realizes the modeling of intention as choice

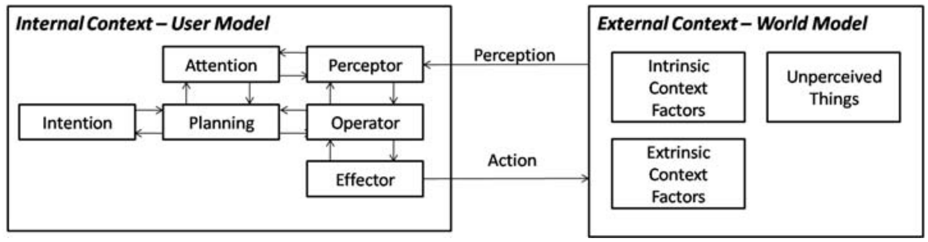

Figure 1: Human-environment interaction model

with commitment in terms of planning theory [Cohen and Levesque 1990]. On the other hand, the environment can be decomposed into three areas, following the work on context by [Öztürk 1998]. The environment consists of: i) those things which are directly related to human intention (intrinsic context), ii) those which are not related to intention (extrinsic context) and iii) those things which are not perceived.

The perceptor consumes and processes stimuli from the outside. Perception is capable of filtering and focusing stimuli. The motivator initiates actions which are intended to interact with the surrounding world. The operator organizes perceptor and effector with respect to higher level cognitive processes. These higher level cognitive processes are guided by individual intentions. Intentions trigger the creation of plans to realize the intentions. Plans trigger the operator, thus influencing perceptor and effector. Thus, even if the human is capable of perceiving context factors, they are still filtered based on plans with respective intentions.

Context-aware and attention-aware systems support this model by guiding user attention. They can provide support by a)supporting the interaction with the intrinsic context features b) showing deficits of the individual selection of intrinsic and extrinsic context features by proposing an extrinsic context feature as intrinsic context feature or vice-versa c) proposing unperceived things to the user to extent the intrinsic and extrinsic context. An important aspect is the general focus of context-aware and attention aware systems on few static intentions for which context features or awareness features are exploited. Therefore, they rely on implicit models of intention, e.g. based on the usage-scenario of an application. Once one assumes that human intention can vary, e.g. by extending the support scenario of context-aware or attention-aware systems beyond single transactions used in situations with static intention, it is necessary to explicitly model intention, too.

3 Task Models and Task Model Purpose

Task models provide information about tasks, generally referred to as atomic units of work [Godehardt et al. 2007]. By describing task objectives, task models provide a specific type of intention information. In some cases task models additionally integrate information about task execution processes which is a representation of plans to realize intentions. Thus, task models can be used to externalize aspects of individual intention and planning. In the following we review task models with respect to these attributes.

Task models have been applied for purposes like (1) model user system interaction, (2) structure people work, (3) propose actions and (4) propose artifacts (see figure 2). Existing task models apply different modeling methods. We have reviewed systems using (A) hierarchies of subgoals and actions, (B) grammars of actions, (C) sequences of actions and (D) collections of activities. Modeling method (A), hierarchies of subgoals and actions is close to the task planning described in psychological studies [Newell and Simon 1972]. Method (B) to (D) simplify the task model, having the effect of simpler generation of modeling instances with higher robustness to the weak structure in exchange to a less precise model. A respective organization of task models is visible in figure 3. A classification by modeling purpose shows a relation between the model purpose and the applied modeling methodology.

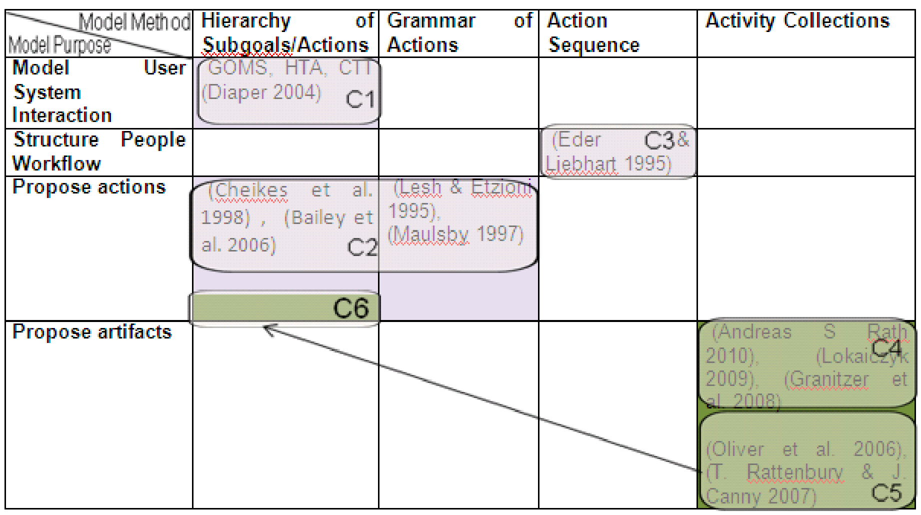

Figure 2: Association of Task Models to Purpose

Specific focus of the review are task models for context-aware and attention-aware systems. They are discussed in the following overview in more detail. Additionally, a separate overview of these systems and the applied task models is given in figure 3.

3.1 C1/C2 Manual model instance creation for high precision

The models which include a hierarchical task representation (A) are used for task analysis (C1) and context-aware systems to recommend actions (C2). Hierarchical task models decompose a task into smaller units of work which again can be decomposed until a preferred granularity is reached. The result is a soft decomposition of execution complexity. These models have the highest complexity of the reviewed task models and have a solid psychological foundation [Newell and Simon 1972]. In both cases task models of high complexity are created by experts. Whereas task analysis focuses on understanding and evaluating task execution (e.g. when interacting with a system) (C1), the context-aware systems (C2) execute the models to propose actions to the user [Bailey et al. 2006, Lesh and Etzioni 1995, Cheikes et al. 1998]. A comparable approach is the modeling of tasks based on grammars, in the sense that grammars can express hierarchical constructs. One example for grammar based task models is Activity Streams [Maulsby 1997].

Task analysis and the included recommender systems require a priori knowledge of existing tasks and manual task modeling. Task analysis demands manual modeling, as the model creation is a main aspect of the analysis process. In contrast task model creation for the proposal of user actions not necessarily demands the manual creation for knowledge gain. In fact, manual model creation for such systems is a tedious and error prone task. Nevertheless, automation of task model creation for proposals is difficult, due to the complexity of the models.

3.2 C3 Manual model instance creation for predefined workflows and execution support

Workflow systems (C3) use a simple task model for a predefined task execution process [Eder and Liebhart 1995]. The model is generated manually. Still, different approaches for the (semi-)automatic generation of model instances exist. The model is used to coordinate and control workflow execution. A priori knowledge of existing tasks is needed and the tasks are very strict execution sequences, not reflecting variance of the execution process.

Name	Task Model	Knowledge Base	Support	Referenz
LIP	Resources with competency requirements	4 Phase Context Ontology	Recommend learning resources	[Schmidt 2007]
SWISH	App. Name sequences	Machine Learning (ML), PLSI	Recommend next steps	[Oliver et al. 2006]
Task Tracer	Bag of resources	ML (e.g. Perceptron)	Resource recommendation	[Shen et al. 2007]
UICO	HTD	Action/Res. Ontology/ML	Resource recommendation	[Rath 2009]
Dyonipos	HTD	Action/Resource Ontology and ML (e.g. SVM)	Resource recommendation	[Granitzer et al. 2008]
WIMP for ETS	HTD	Goals/Actions Hierarchy	Recommend next steps	[Cheikes et al. 1998]
Suitor	Bag of keywords	Semantic Analysis	Display topics of interest	[Maglio et al. 2000]
ActivityStreams	Application and resource sequences	Grammar representation	Adapt User Interface	[Maulsby 1997]
CAM	Bag of resources	Automatic resource logging	Not described in reference	[Wolpers et al. 2007]
CAAD	Bag of resources	ML (Pattern mining)	Resource recommendation	[Rattenbury et al. 2007]
Goal recognizer	Action sequence / goal	Plans as modeled grammar	Recommend next steps	[Lesh et al. 1995]
PETDL	Action sequence / goal	Manually modeled patterns	Recommend next steps	[Bailey et al. 2006]
App. Monitor	Action sequence/wrds	Workflows and ML (e.g. SVM)	Propose resources to goal	[Godehardt et al. 2007]
UOH	Bag of resources	ML (Case Based Reasoning)	Resource recommendation	[Schwarz 2006]
Lumiere	Sequence of activities	ML (Bayes Models)	Recommend next steps	[Horvitz et al. 1998]
UMEA	Bag of resources	Program by demonstration	Resource recommendation	[Kaptelinin 2003]

Figure 3: Systems supporting users based in context and attention data in the domain of desktop computing

3.3 C4/C5 Automatic model instance creation for Activity Collections

Recommender systems to identify tasks and support the task execution process, apply mechanisms of supervised (C4) or unsupervised (C5) learning to detect task instances. The feature vectors used as input for the applied algorithms rely on the collection of action and artifact sequences.

Whereas the supervised approaches demand the initial detection of tasks and respective training data sets, the unsupervised approaches identify tasks solely based on actions with attributes like resources and time.

Examples for supervised machine learning are the Task Tracer system [Shen et al. 2009, Shen et al. 2007], CAM (ContextualizedAttentionMetadata), [Wolpers et al. 2007], Dyonipos [Granitzer et al. 2008], UICO [Rath et al. 2008], APOSDLE [Lokaiczyk 2009] or the UMEA system [Kaptelinin 2003]. The approach is similar for all systems: the system tracks resources or activity sequences used in a task context and uses this information to generate recommendations in upcoming executions based on a previously trained model. A wide range of algorithms for supervised machine learning have been used by these approaches, e.g. [Rath et al. 2008] applies Naive Bayes, Linear Support Vector Machines w. different cost parameters, J48 decision trees and k-nearest neighbor to classify bags of words of identified activities. [Lokaiczyk 2009] works with n-gram and spreading activation on graphs of used resources and applies naive bayes, one rule decision-tree, rule-learning and support vector machine on aggregated activity information.

The previously described approaches share one difficult assumption: tasks are not known a-priori. The CAAD system [Rattenbury and Canny 2007] and the SWISH [Oliver et al. 2006] system follow a different approach: tasks are previously unknown. The systems track the user system interaction and apply techniques like parameter estimation and clustering to identify tasks.

All described approaches (supervised and unsupervised) create instances for task models to support by resource recommendations on the fly. They solve problems of task switch detection, task classification and task support on resource level. They fail to provide detailed execution plans, as the respective information is only partially used to classify tasks.

3.4 Meeting requirements of Human-Environment Interaction Model

Following the human-environment interaction model, it is important to integrate an explicit representation of intention and plan within a task model, to fit user support to the intention guiding the interaction of human and environment. Within the spectrum of described task models this is only given for task models that include the task execution process as action sequence. The context-aware and attention-aware systems which were reviewed include the task execution process as a collection of resources, not reflecting the actual purpose of a resource within a task execution process.

A hierarchy, including a decomposition of a task into fine grained activities and decomposition of activities into operations is eligible. Still, it is important that such a hierarchy can be maintained based on automatically extracted data, to improve maintainability. In the following, a task model is proposed which follows these ideas.

3.5 Task-centric User Support Based on Activity Data

In the following we describe work based on the extraction of activity data from user system interactions, to generate and detect tasks. This includes the supervised and unsupervised task models mentioned in the previous section (C4/C5).

3.5.1 Supporting the User during Task Execution

Support during task execution uses sensors which generate events for user-system interactions. Most systems realize one generic process depicted in figure 4.

3.5.1.1 Task Switch Detection: Task switch detection based on the event stream (1)

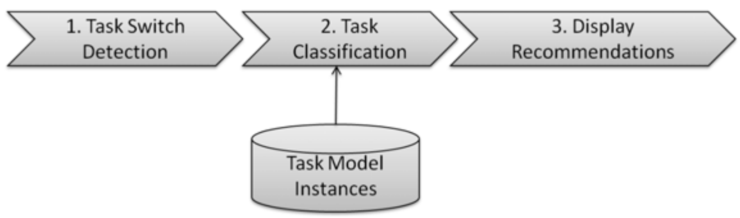

Figure 4: Task Detection for user support

Task switch detection has been modeled as change point detection [Shen et al. 2007] or likewise as sequence segmentation [Link et al. 2005]. Simple approaches using the markov assumption have been identified as prone to errors, as more than one latent variable exists, motivating task switches [Shen et al. 2006]. Algorithms, focusing on the tackled topic in a situation have shown good performance [Shen et al. 2006]. An extension of the included text and information of process steps in a task seem to be useful support areas.

3.5.1.2 Task Classification: Identified events are classified with respect to a task model (2)

Once the task switching problem is solved, the classification is simple. Based on the feature set for the extracted task a task model instance is selected.

3.5.1.3 Display Recommendations: Artifact or action recommendations are displayed to the user (3)

Existing approaches use recommender lists [Lokaiczyk 2009], net structures [Lokaiczyk 2009] or task bar elements [Shen et al. 2009]. Unobtrusive methods of recommendation are important, as a support and no supervision of the task execution process is intended.

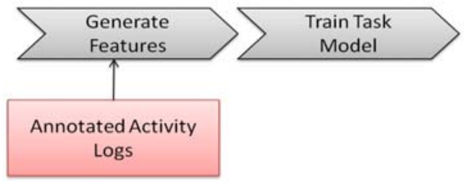

Figure 5: Task Model instance generation using supervised machine learning

3.5.2 Processing of User Activity Information for Task Identification and Classification

Two major approaches can be identified: supervised and unsupervised machine learning.

3.5.2.1 Supervised Machine Learning

The general process for supervised machine learning is depicted in figure 5: Based on an annotated activity log features are generated and a model for a selected algorithm (e.g. Support Vector Machine) is trained. Supervised machine learning is a well-observed field. The performances of various feature sets and algorithms has been evaluated [Granitzer et al. 2008], [Rath 2010], [Lokaiczyk 2009]. A main problem of these approaches is the need to know all existing tasks a priori and the tedious generation of useful training data for the tasks.

3.5.2.2 Unsupervised Machine Learning

Unsupervised approaches identify task classes without a demand for manually modeled training data. It is necessary to generate a convenient model of tasks and identify algorithms which extract task model instances based on user activity logs. The general process for unsupervised machine learning is depicted in figure

*Figure 6: Task Model instance generation using unsupervised machine learning, no standard process exists. *) can be element of the process*

6: activity logs (without annotations) are processed based on combinations of segmentation, clustering and classification algorithms to generate a task model.

Promising results were generated by using the window title information and sequence data with temporal information [Rattenbury and Canny 2008], [Oliver et al. 2006]. Generative probabilistic models used in the domain of information retrieval have proven useful, e.g. Probabilistic Latent Semantic Analysis [Hofmann 1999] and GaP [Canny 2004].

3.5.2.3 Promising direction: Use of Unsupervised Methods to Extract Tasks as Processes and Resources

A promising direction for intention-aware systems is the utilization of unsupervised methods to generate instances of task models which include a hierarchical representation of task execution. In the following, activity schemes are proposed for this purpose.

4 Task Model: Activity Schemes

To address the missing integration of plan representations and intentions in task models for context-aware and intention-aware systems, we present activity schemes. Activity schemes include probabilistic models of task execution plans, thus an externalization of the cognitive planning of the individual in the human-world interaction model. Activity schemes are based on observable facts: actual task execution on the computer desktop. Before describing activity schemes in detail, we describe knowledge actions as building blocks of task execution processes.

4.1 Knowledge Actions as Building Blocks of Tasks

In favor of identifying plans which make the execution process of a task transparent, we search for self-contained building blocks of tasks which reoccur in all kinds of tasks executed in a desktop environment. Task analysis provides a valuable toolbox to record domain specific tasks, realize solutions and validate them (c.f. [Diaper 2004]). Still, there is a lack of a generic classification of tasks and task elements. [Becks and Seeling 2001] who also identify this problem, describe some domain specific taxonomies which only apply for few domains, e.g. [Gaffar et al. 2004].

Still, there is evidence that such building blocks exist for desktop work. Computers are machines of sign transformation and, if used for office work, mainly realize interaction with information coded in signs and images. Though focusing on sign interaction in the learning process, the theory of media functions by [Hampel and Keil-Slawik 2001] shows that interaction can be organized by few operations. Primary media functions are: Create, Transfer, Arrange and Connect. A study by [Hädrich 2008] has comparable outcome: work can be described by a set of knowledge actions (expressing: authoring/co-authoring, translating: training, acquisition/monitoring: update/ feedback, networking: expert search/invitation).

It seems to be a useful approach to build upon the work of [Hädrich 2008] and [Hampel and Keil-Slawik 2001] towards reusable task building blocks. One can consider tasks as composed of finer grained actions - knowledge actions [Hädrich 2008], which reoccur in different settings within the execution processes of knowledge-intensive tasks. This means that a user will perform an information search or an authoring process in a similar manner, independently of the specific task. Consequently, we can train a system to identify such knowledge actions apart from the specific task. This simplifies the use of machine learning approaches, as the trained information on knowledge actions can be used independently from the task.

4.2 Activity Schemes as Layer Model

Activity schemes extend descriptive human task models by two additional levels of information: a semantic level and a process level (see figure 7). In the following we describe each of these levels.

Level 1. Task - Description The initial level captures generic information about the task. This includes definition of the task objective, of the role and the competency requirements of the executor. Dependent on the use of the task model this information can be automatically generated based on user-system interaction information or it can be modeled manually.

Level 2. Activity Scheme - Topic Assignment A task is composed of information access and information transformation processes using generic applications. Information created and consumed during task execution is an important task characteristic. We represent this information as collection of words and information objects which are utilized during execution. The similarity of tasks can be

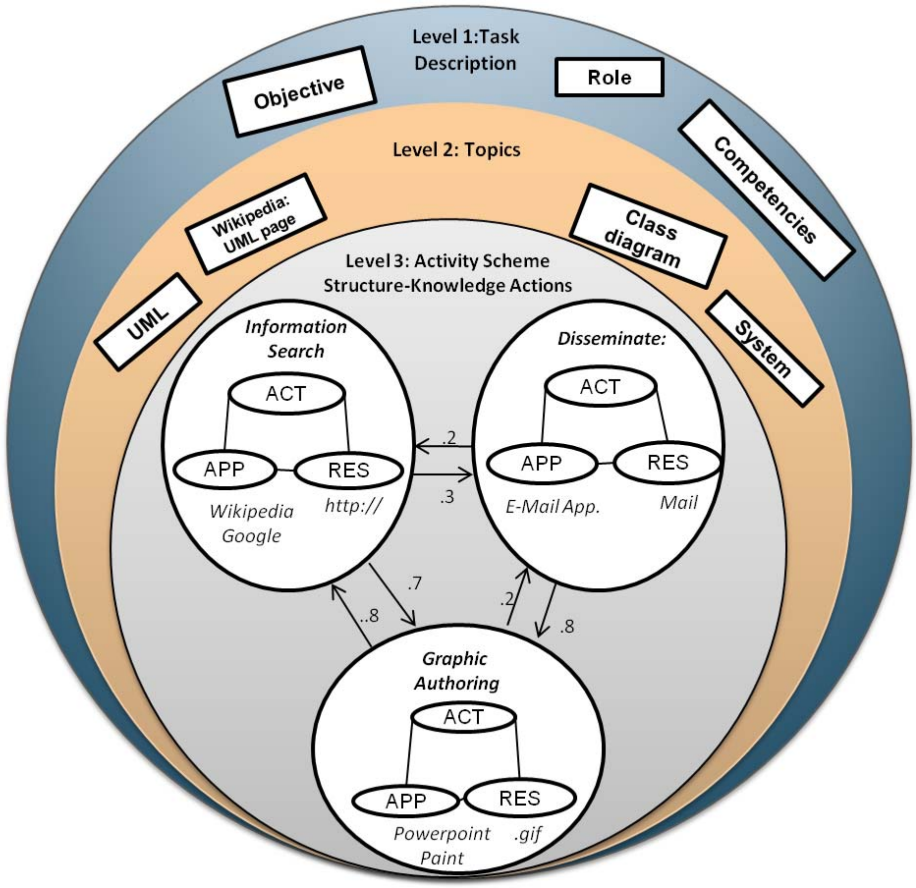

Figure 7: Task as Carrier for Activity Scheme

calculated using natural language processing approaches like vector space models (representation of task similarity as vector similarity) or generative topic models (presentation of task similarity as similarity of topic distributions).

Level 3. Activity Scheme - Knowledge Action Structure Execution plans are modeled as connected knowledge actions. Thus, actual execution processes are individual transitions between knowledge actions. Each knowledge action contains information about the applications used to perform it and the type of resource the work is executed on.

5 A Framework for Intention-Aware Systems

The human-environment interaction model describes the connection of intention, plans and actions. We have focused on task models to externalize intentions with respective plans for intention-aware systems. We reviewed existing context-aware and attention-aware systems and additionally task models in other domains. A lack of complexity with respect to plans has been identified in existing task models for context-aware and attention-aware systems and activity schemes have been proposed as task model to improve the plan and intention representation within task models. In the following we present a framework for an intention-aware system (see figure 8). The framework has three layers: "Context-Awareness Pipeline", "Intention Elicitation Pipeline" and "Situational Support". The framework includes activity schemes as task model which can be automatically generated based on user activity data.

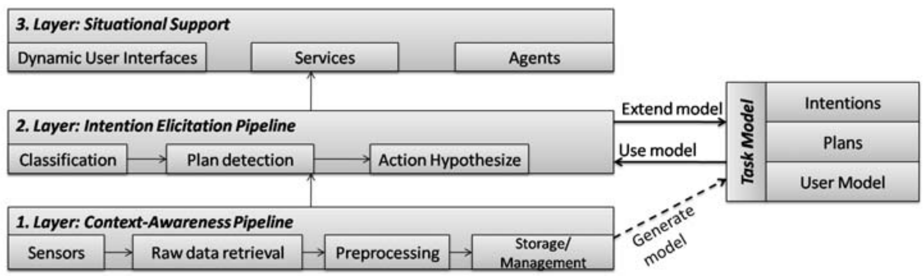

Figure 8: Framework for intention-aware systems

5.1 Context-Awareness Pipeline

The first layer, "Context-Awareness Pipeline" is a pipeline of sensors to detect observable facts about the interaction of a human with his environment, data aggregation and management of storage and data delivery. The layer itself is a context-aware application as proposed by [Baldauf et al. 2007, Hoh et al. 2006]. Aggregation of sensor data is used to generate knowledge actions with the specific information about used resources and applications. The "Context-Awareness Pipeline" provides input for two different processes: task model generation and user support during the working process. Task model generation means that after longer periods of work (e.g. daily after work completion) all retrieved sensor events are mined to create activity schemes as task model instances (dotted line labeled with "Generate model" in figure 8). User support during the work process refers to the classification of user actions with respect to activity schemes included in the system (layers two and three in figure 8).

5.2 Intention Elicitation Pipeline

The second layer, "Intention Elicitation Pipeline" processes the context and awareness information from the base layer to identify the current user intention. The output of the context-awareness pipeline is classified based on the activity schemes which were mined based on earlier task executions. The active knowledge action within the activity scheme is detected and the system generates hypotheses about subsequent actions. This information is forwarded to the third layer of the framework.

5.3 Situational Support

The third layer "Situational Support" utilizes the information about active activity scheme and hypothesis on user actions to create situation-specific support for the user. The system can select useful support mechanisms for the identified intention. The range of support mechanisms comprises dynamic user interfaces, service provision or agents [Eck and Soh 2009].

6 Conclusion: Intention-aware systems and their benefit

In this paper, we have presented intention-aware system, to integrate aspects of context-aware and attention-aware systems for the identification and support of intention. Based on a human-world environment model the importance of intention and respective planning has been highlighted. The externalization of intention and plans in task models has been discussed based on a literature review. Activity schemes have been proposed as task model for intention-aware systems specifically highlighting the process aspect of task execution. Based on activity schemes a framework for intention-aware systems has been proposed based on a framework for context-aware systems.

Our future work focuses the exemplary implementation of the described framework for intention-aware systems. This includes the application of context-event processing to aggregate sensor events to knowledge-actions and the mining of task model instances in streams of aggregated events.

References

[Bailey et al. 2006] Bailey, B., Adamczyk, P., Chang, T., and Chilson, N.: A framework for specifying and monitoring user tasks. Computers in Human Behavior, 22(4):709–732, 2006.

[Baldauf et al. 2007] Baldauf, M., Dustdar, S., and Rosenberg, F.: A survey on context-aware systems. International Journal of Ad Hoc and Ubiquitous Computing, 2(4):263–277, 2007.

[Becks and Seeling 2001] Becks, A. and Seeling, C.: A Task-Model for Text Corpus Analysis in Knowledge Management. In Proceedings of UM-2001 Workshop on User Modeling, Machine Learning and Information Retrieval, 8th International Conference on User Modeling, Sonthofen (Germany), July 2001.

[Canny 2004] Canny, J. (2004): GaP: a factor model for discrete data. Proceedings of the 27th annual international ACM SIGIR conference on Research and development in information retrieval.

[Cheikes et al. 1998] Cheikes, B., Geier, M., Hyland, R., Linton, F., and Rodi, L.: Embedded training for complex information systems, Tutoring Systems, pages 36–45, 1998.

[Cohen and Levesque 1990] Cohen, P. and Levesque, H.: Intention is choice with commitment. Artificial intelligence, 42(2-3):213–261, 1990.

[Diaper 2004] Diaper, D.: Understanding task analysis for human-computer interaction, chapter 1, pages 5–47, Laurence Erlbaum Associates, London, 2004.

[Eck and Soh 2009] Eck, A. and Soh, L.: Intelligent User Interfaces with Adaptive Knowledge Assistants. ponca.unl.edu, pages 1–19, 2009.

[Eder and Liebhart 1995] Eder, J. and Liebhart, W.: The Workflow Activity Model WAMO,. Information Systems, 1995.

[Gaffar et al. 2004] Gaffar, a., Sinnig, D., Seffah, a., and Forbrig, P.: Modeling patterns for task models, Proceedings of the 3rd annual conference on Task models and diagrams - TAMODIA '04, 2(1):99, 2004.

[Godehardt et al. 2007] Godehardt, E., Faatz, A., and Goertz, M.: Exploiting Context Information for Identification of Relevant Experts in LNCS, pages 217–231, 2007.

[Granitzer et al. 2008] Granitzer, M., Kroll, M., Seifert, C., Rath, A., Weber, N., Dietzel, O., and Lindstaedt, S.: Analysis of machine learning techniques for context extraction, ICDIM 2008. Third International Conference on Digital Information Management, pages 233–240, 2008.

[Hädrich 2008] Hädrich, T.: Situation-oriented Provision of Knowledge Services, Dissertation, Martin Luther Universität Halle-Wittenberg, 2008.

[Hampel and Keil-Slawik 2001] Hampel, T. and Keil-Slawik, R.: sTeam: structuring information in team-distributed knowledge management in cooperative learning environments, Journal of Educational Resources in Computing, 1(2es):3–es, 2001.

[Hofmann 1999] Hofmann, T. (1999): Probabilistic latent semantic analysis. In Proceedings of the 22nd annual international ACM SIGIR conference on Research and development in information retrieval, pages 50–57, 1999.

[Hoh et al. 2006] Hoh, S., Devaraju, A., and Wong, C.: A Context Aware Framework for User Centered Services, intelligentmodelling.org.uk, pages 1–8, 2006.

[Horvitz et al. 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K.: The Lumière project: Bayesian user modeling for inferring the goals and needs of software users, In Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence, pages 256–265,1998.

[Kaptelinin 2003] Kaptelinin, V. (2003): UMEA: translating interaction histories into project contexts, In Proceedings of the SIGCHI conference on Human factors in computing systems, number 5, pages 353–360. ACM, 2003.

[Lesh and Etzioni 1995] Lesh, N. and Etzioni, O.: A sound and fast goal recognizer, In International Joint Conference on Artificial Intelligence, volume 14, pages 1704–1710, 1995.

[Link et al. 2005] Link, H., Lane, T., and Magliano, J.: Models and Model Biases for Automatically Learning Task Switching Behavior, Foundations of augmented cognition, Lawrence Erlbaum Associates, 2005.

[Lokaiczyk 2009] Lokaiczyk, R.: Verfahren kontextbasierter Nutzerzielanalyse, Suedwestdeutscher Verlag fuer Hochschulschriften, 2010.

[Maglio et al. 2000] Maglio, P., Barrett, R., Campbell, C., and Selker, T.: SUITOR: An attentive information system, In Proceedings of the 5th international conference on Intelligent user interfaces, pages 169–176. ACM, 2000.

[Maulsby 1997] Maulsby, D.: Inductive task modeling for user interface customization, In Proceedings of the 2nd international conference on Intelligent user interfaces, pages 236–240. ACM, 1997.

[Newell and Simon 1972] Newell, A. and Simon, H.: Human problem solving, Prentice-Hall, Englewood Cliffs, NJ, 1972.

[Oliver et al. 2006] Oliver, N., Smith, G., Thakkar, C., and Surendran, A.: SWISH: semantic analysis of window titles and switching history, In Proceedings of the 11th international conference on Intelligent user interfaces, pages 201–209. ACM, 2006.

[Öztürk 1998] Öztürk, P.: A context model for knowledge-intensive case-based reasoning, International Journal of Human-Computer Studies, 48(3):331–355, 1998.

[Rath et al. 2008] Rath, A., Weber, N., Kröll, M., Granitzer, M., Dietzel, O., and Lindstaedt, S.: Context-aware knowledge services, Workshop on Personal Information Management: PIM, 2008.

[Rath 2009] Rath, A. S. (2009): UICO: An ontology-based user interaction context model for Automatic Task Detection on the Computer Desktop: CIAO '09: Proceedings of the 1st Workshop on Context, Information and Ontologies, pages 1–10, 2009.

[Rath 2010] Rath, A. S. (2010): User Interaction Context - Studying and Enhancing Automatic User Task Detection on the Computer Desktop via an Ontology-based User Interaction Context Model, Know-Center, 2010.

[Rattenbury and Canny 2007] Rattenbury, T. and Canny, J.: CAAD: an automatic task support system. In Proceedings of the SIGCHI conference on Human factors in computing systems, pages 696–706. ACM, 2007.

[Rattenbury and Canny 2008] Rattenbury, T. L. and Canny, J. F.: An Activity Based Approach to Context-Aware Computing, University of California at Berkeley Berkeley, 2008.

[Roda and Thomas 2006] Roda, C. and Thomas, J.: Attention aware systems: Theories, applications, and research agenda, Computers in Human Behavior, 22(4):557–587, 2006.

[Schmidt 2007] Schmidt, A.: Impact of context-awareness on the architecture of learning support systems, Architecture Solutions for E-learning Systems, Idea-Group Publishing, 2007.

[Schwarz 2006] Schwarz, S.: A context model for personal knowledge management, LNCS, 3946:18–33, 2006.

[Shen et al. 2009] Shen, J., Irvine, J., Bao, X., Goodman, M., Kolibaba, S., Tran, A., Carl, F., Kirschner, B., Stumpf, S., and Dietterich, T.: Detecting and correcting user activity switches: Algorithms and interfaces. In Proceedings of the 13th international conference on Intelligent user interfaces, pages 117–126, ACM, 2009.

[Shen et al. 2007] Shen, J., Li, L., and Dietterich, T.: Real-time detection of task switches of desktop users, In Proceedings of the International Joint Conferences on Artificial Intelligence, volume 7, pages 2868–2873, 2007.

[Shen et al. 2006] Shen, J., Li, L., Dietterich, T. G., and Herlocker, J. L. (2006): A hybrid learning system for recognizing user tasks from desktop activities and email messages, Proceedings of the 11th international conference on Intelligent user interfaces - IUI '06, pages 86–92, 2006.

[Stachowiak 1973] Stachowiak, H.: Allgemeine Modelltheorie, Springer, 1973.

[Wolpers et al. 2007] Wolpers, M., Najjar, J., Verbert, K., and Duval, E.: Tracking actual usage: the attention metadata approach. Subscription Prices and Ordering Information, 10(3):106, 2007.