Towards 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 are introduced as a system class which enables user support based on intention detection. Thereby, intention-aware systems build on the user-centric support approach of attention-aware systems and the environment-centric support approach of context-aware systems. A framework for intention-aware systems is proposed, highlighting the importance of a task model. We review 16 context-aware and attention-aware systems as foundation for the work on a task model for intention-aware systems.

Key Words: intention-aware systems, context-aware systems, attention-aware systems, knowledge work
Category: H.4.1, H1.2

1 Introduction

Knowledge work often is difficult to support, due to its weak structure and unpredictable information requirements. Supporting this kind of work requires situation specific adaptation of information delivery in consent to the user intention. Context-aware systems [Baldauf et al., 2007] and attention-aware systems [Roda and Thomas, 2006] are two approaches for the design of such systems. The respective realizations mainly focus on the detection of user status (attention-aware) or environment status (context-aware) and process this information based on a task model. We see potential in utilizing the information of both system types with respect to a task model. The task model needs to comprise information on the user and their interaction with the environment, by explicating the individual and implicit intentions and plans, to reason about attention and context information. We call such systems intention-aware. Intention-aware systems do not just add another class to software taxonomies. They enable work execution support reflecting the individual working process in a given situation.

In this paper we discuss intention-aware systems to connect context and attention data with user intention. Initially, we conceptualize intention-aware systems (sec. 2) by providing a human-environment interaction model. This model is the foundation of a framework for intention-aware systems. Thereafter, we focus on intention-aware systems in the domain of desktop computing. We review sixteen systems from the domain of 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 and the support functionalities of the system. The outlook section finally summarizes the main aspects of our argumentation and hints towards upcoming work.

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 on 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 on 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 detects user intention based on the situation-specific user attention and the status of the environment. In the following we present a human-environment interaction model. Based on the model, we subsequently describe a framework for an intention-aware system.

2.1 Human-Environment Interaction Model

To model the interaction of human and environment we have extended the K-system model [Stachowiak, 1973] which describes system-world interaction by means of a control circuit (see fig. 1). Considering the K-system as a human being, the human is organized by perceptor, operator, and motivator in connection to the environment. This has been extended in two directions. On the one hand, we have specified the motivator as the connection of intention, attention, and planning. This realizes the modeling of intention as choice with commitment

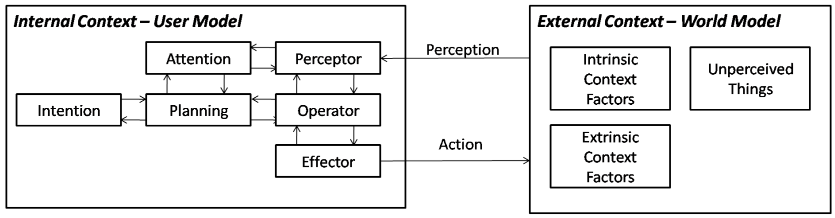

Figure 1: Human-environment interaction model

in terms of planning theory [Cohen and Levesque, 1990]. On the other hand, the environment has been 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.

Perception and action depend on the intention and the related planning. Focusing on awareness as top-down process, it is an instrument to guide perception based on intention. Thus, the context factors only have value once they are associated with intention and resulting plans.

Context-aware and attention-aware systems are valid in this model. An important aspect is, that they generally focus 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 scenario beyond a single application, it is necessary to explicitly model intention, too.

2.2 A Framework for Intention-Aware Systems

The human-environment interaction model has described the connection of intention, awareness and perceived context. To realize a system which is able to reason about these aspects, it is necessary to identify methods to decide on human intention based on observable facts. In the following we propose a three layer architecture for intention-aware systems (see fig. 2).

The lowest layer, "Context-Awareness Pipeline" describes a pipeline, using sensors to detect observable facts about the interaction of the user with their environment, clusters this data and manages its storage or delivery. The layer itself is a context-aware application as proposed by [Baldauf et al., 2007, Hoh et al., 2006].

The second layer, "Intention Elicitation Pipeline" processes the context and awareness information from the base layer to identify the current user intention.

Based on the detected intention and the respective plans, the system generates hypotheses about subsequent actions. Unlike context-aware or attention-aware system, this processing is based on a task model which integrates information about user intentions, related plans, and a user model. This connects the attention information with the context information and links them to intention. As manual modeling of instances for such a task model is a tedious and error-prone task, semi-automatic approaches seem to be useful. Possible approaches are programming by demonstration [Cypher, A. And Halbert, 1993] or learning sets to train classifiers based on information from the base layer. Based on the information processed in layer 2 additional data can be obtained through extension of existing models.

The third layer "Situational Support" provides support to the user based on the intention and on the knowledge about the user (general demand and preference of support). The system can select useful support mechanisms for the identified intention. The range of support mechanisms comprises dynamic user interfaces, service provision or agents.

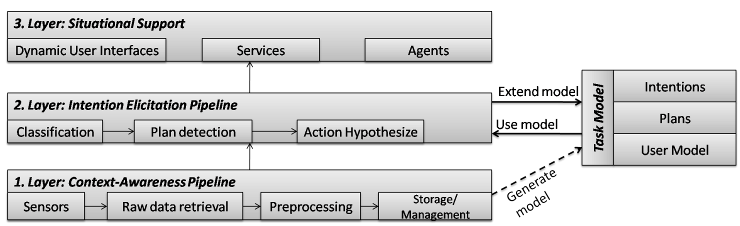

Figure 2: Framework for intention-aware systems

3 Intention-Aware Desktop Computing

In the following we focus on intention-aware desktop computing. The computer desktop is an environment which can be easily instrumented with sensors to identify context features and user awareness. We have highlighted that a task model, integrating user intentions, connected plans and a user model is the most important aspect of an intention-aware system. The work on such a task model shall reflect existing efforts on task modeling.

3.1 System Task Model Review

We review task models of sixteen applications. All reviewed systems use an instrumented desktop environment to detect user-system interaction. As such they all provide user support based on a similar information source: a sequence of classified system events. In the following a task is generally referred to as an atomic unit of work [Godehardt et al., 2007]. The reviewed systems classify tasks based on facets of the task execution process, thus the model includes additional information. Each system has been reviewed with respect to the task model, the knowledge base which is used to create task instances for the model and the type of support given by the system. The review is based on the respective publications. If no task model has been made explicit, the classification is inferred from the description of data models on data processing. We included systems which use collected data to generate user support. Ex-post analysis of such data, as described in [Fern et al., 2007] and [Ellis et al., 2006] was not considered. The overview of the review is given in table 1.

3.1.1 Tasks as Bag of Words

The Suitor system [Maglio et al., 2000] considers a task as a set of keywords. These keywords are extracted based on resources the user interacted with and are used to identify information of interest.

3.1.2 Tasks as Bags of Resources

Many systems identify tasks as bags of resources which are used for the actual task execution!!!!!!. Such systems are the Task Tracer system [Shen et al., 2009, Shen et al., 2007], the user observation hub [Schwarz, 2006], CAM (Contextualized Attention Metadata) [Wolpers et al., 2007] or the UMEA system [Kaptelinin, 2003]. The approach is similar: the system tracks resources used in a task context and uses this information to generate recommendations in upcoming executions. The sequence of resource use in a task is unimportant. An extension is the LIP system, which proposes learning resources to the user: it calculates a competency gap based on the resources the user works with and on a user competency model [Schmidt and Engineering, 2007]. The CAAD (Context-Aware Activity Display) [Rattenbury and Canny, 2007] follows an interestingly different approach: the system uses pattern mining to identify clusters of resources, eg. based on temporal co-occurrence. These cluster then implicitly represent a task.

3.1.3 Tasks as Sequences of Actions

The following systems consider tasks as sequence of actions on resources and application: Dyonipos [Granitzer et al., 2008], UICO [Rath, 2009], Aposdle Monitor [Godehardt et al., 2007] and SWISH [Oliver et al., 2006] use machine learning algorithms (e.g. Support Vector Machines, Graph Kernel) to identify a task based on the user interaction sequence. Machine learning through a Bayes Model for the use of a single application has been realized by the Lumière system [Horvitz et al., 1998]. Another approach is the modeling of tasks based on automatically detected grammars. This is done by Activity Streams [Maulsby, 1997]. Unlike the previous approaches, the goal recognizer [Lesh and Etzioni, 1995] and the PETDL [Bailey et al., 2006] demand the manual creation of a grammar for a task.

3.1.4 Tasks as Hierarchical Work Decompositions

Hierarchical Task Analysis (HTA) is a popular approach in the domain of task analysis. A task is decomposed in smaller units of work which again can be decomposed until a preferred granularity is reached. The result is a soft decomposition of execution complexity. The WIMP system for embedded training systems [Cheikes et al., 1998] uses manually modeled task decompositions.

3.2 Intention-aware systems and their benefit

TODO: using user attention to identify intention and focus on those aspects of the context which are useful for the resp. intention. +short example

4 Conclusion

Intention-aware systems combine the strengths of attention-aware systems and context-aware systems to enable situation-specific work execution-support. The task model, bridging user information and context data with respect to execution activities is the central challenge for realizing intention-aware systems.

The reviewed task models show that task models of high complexity often demand manual user effort for their creation (e.g. the HTA models). Automation has been realized, focusing on the sequence of actions by applying machine learning algorithms. Only few task models integrate user modeling or goal modeling (which can hint towards intention modeling). The effective integration of a user model and an intention model into a task model remains an open topic. Currently, we work on the extension of a task model to enable intention-aware user support. Thereby, we especially focus on the aspect of user and intention modeling.

References

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

[Baldauf et al., 2007] Baldauf, M., Dustdar, S., and Rosenberg, F. (2007). A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263277.

[Cheikes et al., 1998] Cheikes, B., Geier, M., Hyland, R., Linton, F., Rodi, L., and Schaefer, H. (1998). Embedded training for complex information systems. *Lecture Notes in Computer Science*, page 3645.

[Cohen and Levesque, 1990] Cohen, P. and Levesque, H. (1990). Intention is choice with commitment. *Artificial intelligence*, 42(2-3):213261.

[Cypher, A. And Halbert, 1993] Cypher, A. And Halbert, D. (1993). *Watch what I do: programming by demonstration.* The MIT press.

[Ellis et al., 2006] Ellis, C., Rembert, A., Kim, K.-h., and Wainer, J. (2006). Beyond workflow mining. *Lecture Notes in Computer Science*, 4102:49.

[Fern et al., 2007] Fern, X. Z., Komireddy, C., and Burnett, M. (2007). Mining Interpretable Human Strategies: A Case Study. *Seventh IEEE International Conference on Data Mining (ICDM 2007)*, pages 475–480.

[Godehardt et al., 2007] Godehardt, E., Faatz, A., and Goertz, M. (2007). -Exploiting Context Information for Identification of Relevant Experts in. *Lecture Notes in*, pages 217–231.

[Granitzer et al., 2008] Granitzer, M., Kroll, M., Seifert, C., Rath, A., Weber, N., Dietzel, O., and Lindstaedt, S. (2008). Analysis of machine learning techniques for context extraction. In *Digital Information Management, 2008. ICDIM 2008. Third International Conference on*, page 233240.

[Hoh et al., 2006] Hoh, S., Devaraju, A., and WONG, C. (2006). A CONTEXT AWARE FRAMEWORK FOR USER CENTERED SERVICES. *intelligentmodelling.org.uk*, pages 1–8.

[Horvitz et al., 1998] Horvitz, E., Breese, J., Heckerman, D., Hovel, D., and Rommelse, K. (1998). The Lumiere project: Bayesian user modeling for inferring the goals and needs of software users. In *Proceedings of the fourteenth Conference on Uncertainty in Artificial Intelligence*, page 256265. Citeseer.

[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, page 353360. ACM New York, NY, USA.

[Lesh and Etzioni, 1995] Lesh, N. and Etzioni, O. (1995). A sound and fast goal recognizer. In *International Joint Conference on Artificial Intelligence*, volume 14, page 17041710. Citeseer.

[Maglio et al., 2000] Maglio, P., Barrett, R., Campbell, C., and Selker, T. (2000). SUITOR: An attentive information system. In *Proceedings of the 5th international conference on Intelligent user interfaces*, page 169176. ACM New York, NY, USA.

[Maulsby, 1997] Maulsby, D. (1997). Inductive task modeling for user interface customization. In *Proceedings of the 2nd international conference on Intelligent user interfaces*, page 236. ACM.

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

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

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

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

[Roda and Thomas, 2006] Roda, C. and Thomas, J. (2006). Attention aware systems: Theories, applications, and research agenda. *Computers in Human Behavior*, 22(4):557587.

[Schmidt and Engineering, 2007] Schmidt, A. and Engineering, I. P. (2007). Impact of context-awareness on the architecture of learning support systems. *E-learning*, 2007(c).

[Schwarz, 2006] Schwarz, S. (2006). A Context Model for Personal Knowledge Management. *German Research*.

[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. (2009). Detecting and correcting user activity switches: Algorithms and interfaces. In *Proceedings of the 13th international conference on Intelligent user interfaces*, page 117126. ACM.

[Shen et al., 2007] Shen, J., Li, L., and Dietterich, T. (2007). Real-time detection of task switches of desktop users. In *Proc. of IJCAI*, volume 7, page 28682873.

[Stachowiak, 1973] Stachowiak, H. (1973). *Allgemeine Modelltheorie*. Springer.

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

Table 1: Context-aware and attention-aware systems on the user desktop

Name	Task Model	Knowledge Base	Support	Referenz
LIP	Resources with competency requirements	manually modeled ontology	Recommend learning resources	[Schmidt and Engineering, 2007]
SWISH	Application sequences with similar semantics	machine learning feature vector	Recommend next steps	[Oliver et al., 2006]
Task Tracer	Bag of resources	machine learning feature vector	Resource recommendation	[Shen et al., 2007]
UICO	Hierarchical task decomposition	Manually modeled ontology and machine learning (among others GraphKernels)	Resource recommendation	[Rath, 2009]
Dyonipos	Hierarchical task decomposition	Manually modeled ontology and machine learning (among others SVM)	Resource recommendation	[Granitzer et al., 2008]
WIMP for ETS	Hierarchical task decomposition	Manually modeled hierarchy	Recommend next steps	[Cheikes et al., 1998]
Suitor	Bag of interest keywords	Semantic Analysis	Display topics of interest	[Maglio et al., 2000]
ActivityStreams	Application and resource sequences	Grammar representation	Restructured User Interface	[Maulsby, 1997]
CAM	Bag of resources	Automatic resource logging		[Wolpers et al., 2007]
CAAD	Bag of resources	Machine learning (Pattern mining)	Resource recommendation, task visualization	[Rattenbury and Canny, 2007]
Goal recognizer	Action sequence with goal	Plans as manual modeled grammar	Recommend next steps	[Lesh and Etzioni, 1995]
PETDL	Action sequences with goals	Manually modeled patterns	Recommend next steps	[Bailey et al., 2006]
Aposdle Monitor	Action sequence	Manually modeled workflows and machine learned resources (among others SVM)	Navigational, informational and transactional goal support	[Godehardt et al., 2007]
User Observation Hub	Bag of resources	Machine Learning (Case Based Reasoning)	Resource recommendation	[Schwarz, 2006]
Lumiere	Sequence of activities	Machine Learning (Bayes Models)	Next step recommendation	[Horvitz et al., 1998]
UMEA	Bag of resources	Program by demonstration	Resource recommendation	[Kaptelinin, 2003]