

Machine Learning for User Learning

Janina Schuhmacher

Global Design Enablement, SAP SE

Abstract

To guide users through their business application's functionality requires an intelligent digital assistance system to adapt to the user's stage of expertise. Drawing on event segmentation theory and knowledge space theory, we propose to model the users' domain specific knowledge and their learning process dynamically in the interaction between system and user. In the support process, the system retrieves the support content that matches the user's knowledge state from a hierarchically organized case base. Using case-based reasoning as a psychologically inspired machine learning method facilitates incorporating the user's feedback in the interaction: the system continuously updates its user model to learn how to support the user most efficiently and effectively.

1 Introduction

With all the functionality that today's software provides, the individual user may easily lose track of all the options they have. Yet, no matter how far automation goes, in most cases it is the person who decides when and how to use their business software. Therefore, even the most complex system, no matter how much functionality it provides, can only be as good as its user allows it to be. So, how can we support the user in a way that helps them make optimal use of the computer's resources?

Psychological research indicates that the way users tackle tasks with their business software depends on their level of expertise (Falzon, 1990). Thus, to give efficient support, an assistance system needs to adapt to the user's state of knowledge rather than providing generic solutions. This paper offers a psychological approach into how intelligent digital assistance systems could support users in different stages of expertise. Drawing on event segmentation theory and knowledge space theory, the user's knowledge about their tasks can be dynamically modelled through machine learning and the system's support content is adapted accordingly. Moreover, we present a framework for how the system could learn from users in each interaction to optimize its user model. Thereby, the system learns how to propose the most effective and efficient type of support, using case-based reasoning as a psychologically inspired machine learning method.

2 Psychological Background: Knowledge and Learning

To personalize digital assistance, it is important to consider how human users learn, how they perceive their tasks and how this shapes their interaction with a computer system. The following section explains the psychological base for how to understand and model the user's knowledge and how to adapt the system's support to the individual's stage of expertise.

2.1 Procedural Learning and Skill Acquisition

Anderson (1982) characterizes the movement from novice to expert in skill acquisition as proceduralization. Novices use declarative knowledge to a large extent when they carry out new tasks, even for the smallest units of actions ('operators'). As learning proceeds, specific operators are combined to procedural chunks. At this stage, sequences of operators can be combined to achieve largescale goals: the user reaches an expert stage. Solving a task for an expert involves activating highly integrated procedural chunks that contain a specific goal and a sequence of operators that will achieve that goal.

This means, when guiding a user through a system, novice users may need very specific help, for example click-by-click instructions, to solve tasks effectively. For users who are familiar with the system but do not know its entire functionality on a procedural level yet, it may be enough to activate the procedural chunk corresponding to certain tasks. The users should receive detailed guidance only at the exact points where they struggle.

2.2 How Users Perceive Tasks: Event Segmentation Theory

In this paper, when we talk about a digital assistance system, we take as a starting point a system that guides the user through specific tasks in their working environment. Thus, the way users represent tasks is the base for the system's representation of these tasks. According to event segmentation theory (Zacks et al., 2007), people perceive and conceive activities in terms of discrete, typically goal-directed events with a beginning and an end.

In the given context, tasks can be understood as goal-directed, hierarchical events. The way users segment these tasks is determined both bottom-up, for example by characteristics of the task, and top-down, for example by prior work experience. Users can segment very fine-grained tasks. For example, when they cannot find a certain function in the system, hitting one single button could be segmented as one task. At the same time, they can perceive a whole sequence of clicks and mouse movements as a sequence of events that eventually leads to a goal, for instance the goal of creating a new product in the product catalogue. A user may need to complete several fine-grained sub-tasks in order to achieve one super-ordinate coarse-grained task. An appropriate proposal for guidance should match the level on which events are segmented by the user and how this changes throughout the learning process.

2.3 Modelling the User's Expertise: Knowledge Space Theory

As explained above, the knowledge of each user, that is, their level of expertise in a particular domain, shapes their ways of interacting with a computer system decisively. To provide adequate support, it is crucial that the computer has a model of the user's expertise available. Knowledge space theory provides a framework for modeling the user's state of knowledge. More precisely, the domain-specific knowledge of an individual can be operationalized as a mathematical family of knowledge states, that is, a particular subset of problems or tasks that the individual is capable of solving (Falmagne et al., 1990).

Skill acquisition theory (Anderson, 1982) and previous research (Wang, He & Andersen, 2017) indicate that procedural and semi-procedural domains such as knowledge about business applications are compositional. A task can be decomposed into a series of basic skills required to solve this task. This allows one to define a 'harder'-relation between tasks. One task is less hard than another one if it requires the user to apply only a subset of the basic skills necessary to solve the harder task. Thus, one can organize the task that compose a domain into a partial ordering graph where there is a directed edge between two problems if one problem is directly harder than the other. An individual's knowledge state is represented in terms of their knowledge fringe (Falmagne et al., 1990). The inner fringe contains the most advanced problems in the individual's state, that is, the hardest tasks that the user can still solve. The outer fringe is the set of problems that a user is ready to learn. The user's knowledge can so be represented in the graph as the path of tasks that the user can just solve. In the following, we propose a way of structuring the system's domain and task knowledge analogously to the system's model of the user's knowledge in a hierarchically organized case base. The challenge of providing guidance on the adequate level of abstraction in a given situation can then be solved by traversing the graph along the user's outer knowledge fringe.

3 Representing Knowledge in a Case Base

To realize a digital assistance system that supports the user with the business system's functionality in a comprehensible and personalized manner, knowledge about the system's functionality needs to be represented in a way that is understandable and transparent to the user and that enables learning. To this end, we propose a case-based reasoning approach.

3.1 Psychological Motivation of Case-Based Reasoning

Studies indicate that users appreciate comprehensive explanations of a machine learning systems' results and that this is linked to whether they understand the general idea of the algorithm that produces the results (Stumpf et al., 2009). In case-based reasoning, every case consists of a problem and a solution. New problems are solved by remembering a previous similar situation and by re-using information and knowledge of that situation. This information is adapted to the new situation. After applying a proposed solution, the usefulness and correctness of this solution when it comes to solving the initial problem is revised. This is how

the system learns to find the appropriate solution in a given situation. Updating the retrieval mechanism when problem has been solved enables sustained and incremental learning, analogous to how people learn from experience. This makes case-based reasoning as a methodology intuitively understandable to any user who may not be familiar with computer sciences. This promotes acceptability of the support system and increases users' motivation to improve the system by providing feedback.

3.2 Organizing the Case Base Hierarchically

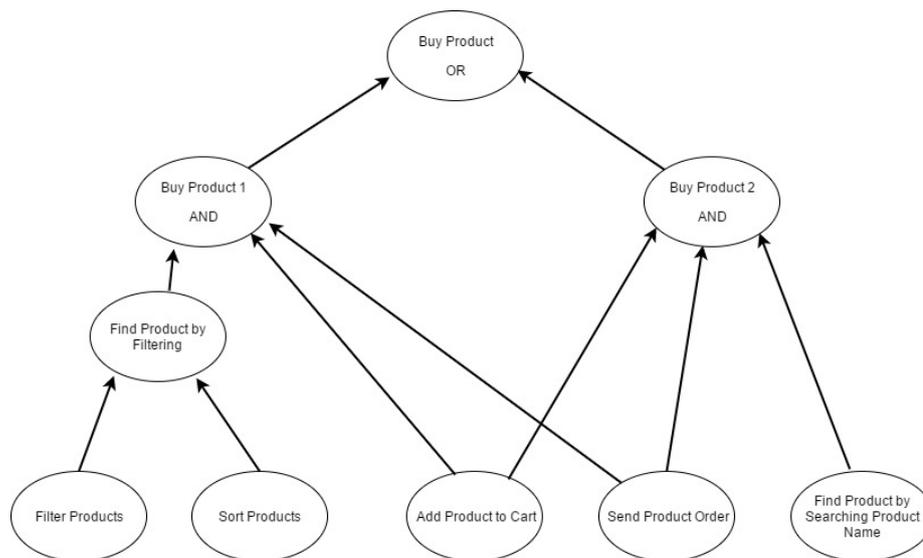


Figure 1: The case of buying a product in the catalog is represented in the case base on different levels of abstraction.

The system's functional knowledge is stored in a case base. As explained before, the knowledge in a given domain can be described as a set of tasks. To achieve their goals, users need to complete tasks by applying sets of operators. Tasks are hierarchical events, meaning that they can be decomposed into sub-tasks and that they are represented on different levels of abstraction by users of different levels of expertise. Consequently, the case base is structured as a hierarchical graph, going from a highly abstract description of a given problem to explanations of sub-tasks. The bottom layer of this case hierarchy captures operators, that is, the smallest meaningful units of action in the interaction between system and user, for example "Upload a File". More complex tasks require the application of a sequence of operators, and on an even more complex level, the solution of a case includes a sequence of sub-tasks. Information about one event, such as creating a new product in the product catalogue, can be represented on different levels in the case base (Figure 1). Supporting the user efficiently requires the assistance system to retrieve the case that will most probably help the user to achieve their tasks on the level of abstraction that matches the user's knowledge state. The following chapter explains how the system can learn efficiency through machine learning.

4 User Modelling to Provide Adequate Support

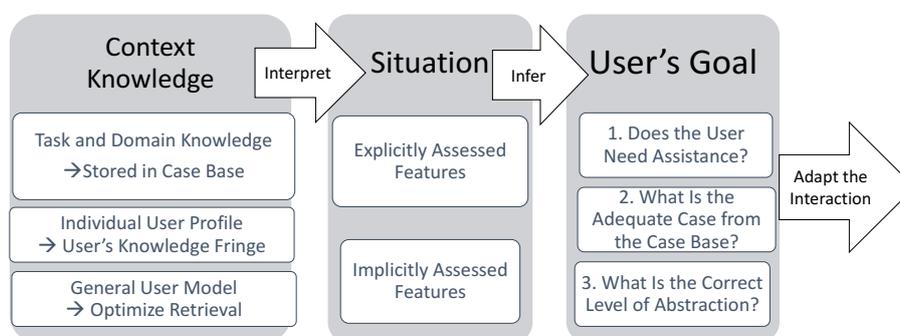


Figure 2: Scope of the user model

The user model captures and structures the information based on which the system adapts its interaction behavior (Figure 2). On this case, the system infers goals and domain-specific knowledge of the user. To begin with, the digital assistance system requires knowledge about the context. This includes knowledge about the domain and application and task knowledge, describing the processes in domain and application. This is stored in a hierarchically organized case base as described in the previous chapter.

Secondly, the user model entails an individual user profile in terms of the user's assumed knowledge fringe. Based on the individual user profile, the system adapts on which level of abstraction it provides support. As a starting point of the interaction, the system generates a default user model. In the interaction, the user can now explicitly request guidance on the levels of abstraction that match their needs. This triggers the system to traverse the case base on a lower or higher (that is, more explicit or more abstract) level of the case hierarchy. Thus, in each session, the system models the user's knowledge state. It infers where the user is in the case hierarchy, that is, which problems they can already solve, and which knowledge state they need to reach in order to achieve their goal. The system proposes guidance by retrieving the sequence of cases that the user should perform to reach their goal on the level of abstraction which they request. Imagine the example of a user entering the product catalog and telling the system: "I am looking for a well-priced laptop." The system uses its context knowledge to interpret the current situation. It draws on its general user model (which contains information on how to understand the user's utterance) to retrieve the case that matches the user's assumed goal (the case "Buy Product 1" from figure 1). In earlier sessions, the user has successfully bought products by searching their name. Thus, based on the individual user profile, the system assumes the sub-tasks of adding the product to the cart and submitting the order to lie in the user's inner knowledge fringe. To be efficient, it only guides the user through filtering the products in the catalog (only laptops) and sorting the results (by price). After each user session, the user profile is updated in terms of the individual's domain specific knowledge. The computer captures what type of assistance the user needed in the specific context of the interaction and which tasks they achieved successfully. Then it adapts the user's knowledge

fringe in the individual user model. In the next session, the system will propose guidance more efficiently along the user's knowledge fringe. In our example, the next situation may be that the user needs to buy equipment for the team. This will require them to perform more complex combinations of the sub-tasks they already know and further, potentially unknown sub-tasks.

Thirdly, the system derives a general user model from its interaction with all. This is how the system infers the user's goal in a specific situation and retrieves the most similar case from the case base. In our example, the system would recommend the user to find the laptop using the filter and sort functions. By following this recommendation, the user confirms their goal in this specific situation. The characteristics of the situation (here, the phrase "I am looking for a well-priced laptop") are then mapped to the corresponding case and integrated into the general user model as one possible value of a feature vector. Thus, the system learns inductively how to interpret the user's behavior and how to abductively create recommendations, for example by using Naïve Bayes classifiers to retrieve the most probable meaning for a user's utterance.

Based on this general user model, the system interprets each new session of interaction extracting certain features. These features can be assessed explicitly (for example via natural language processing) or implicitly (for example by tracking the user's navigational behavior). The features are used to determine when the user needs help, to infer the user's current goal and to find the case from the case base that would most probably help the user to reach their goal. This is done on the level of abstraction that matches the user's assumed knowledge state.

Based on psychological theory, the technical support system presented in this paper learns from the user how to propose effective, adequate solutions. At the same time, the user learns from the system's recommendations. As they interact, the system's efficiency increases as it adapts to the user's needs, that is, their current knowledge state. Consequently, introducing such a system offers considerable economic potential by increasing users' performance and satisfaction.

5 Literaturverzeichnis

- Anderson, J.R. (1982). Acquisition of Cognitive Skill. *Psychological Review*, 89 (4), 369-406.
- Zacks, J. M., Speer, N. K., Swallow, K. M., Braver, T.S., & Reynolds, J.R. (2007). Event perception: a mind-brain perspective. *Psychological Bulletin*, 133 (2), 273-293.
- Falmagne, J.-C., Doignon, J.-P., Koppen, M., Villano, M., & Johannesen, L. (1990). Introduction to knowledge spaces: how to build, test and search them. *Psychological Review*, 97 (2), 201-224.
- Falzon, P. (1990). *Cognitive Ergonomics Understanding, Learning and Designing Human-Computer Interaction*. London, GB: Academic Press.
- Stumpf, S., Rajaram, V., Li, L., Wong, W.-K., Burnett, M. et al. (2009). Interacting meaningfully with machine learning systems: Three experiments. *International Journal of Human Computer Studies*, 67 (8), 639-662.
- Wang, S., He, F., & Andersen, E. (2017). A unified framework for knowledge assessment and progression analysis and design. *CHI 2017*