

Application-independent evaluation of context-awareness

Arthur van Bunningen, Maarten Fokkinga

August 20, 2007

Introduction In this position paper we will focus on the evaluation of a framework for context-aware applications. Our framework focuses on applications that make use of a data management system. Through context-aware adaptation of the answers for certain queries to the underlying database system, we can achieve context-awareness in a way that is partly application-independent.

The problem As pointed out by Abowd and Mynatt ubiquitous systems need to be context-aware (e.g., adapt to the context of the user) [1]. At the same time Abowd and Mynatt point out that these systems should be evaluated in a context as close as possible to the context of authentic use.

Brown *et al.* learn us that people claim that “to become credible in the marketplace, context-awareness needs a killer application” [2]. Typical for killer applications is that it is difficult to predict *if* a certain application becomes a killer (e.g., SMS) and *when* a technique will result in a killer application (e.g., hypertext). Combining the observations of Brown *et al.* and Abowd and Mynatt, the challenge that we are facing is that it is difficult to evaluate context-awareness in its authentic use, without having a killer application.

As has been proposed by several people, for the purpose of evaluation, adaptive systems can be decomposed in several layers. As an exemplary example we will use the structure as proposed by Weibelzahl and Weber [4], given in Fig. 1. The adaptive system first observes the user (1), then infers user properties (2), decides how to adapt (3), and finally presents the adapted interface to the user (4). Each layer builds upon the previous one and all layers need to be evaluated to “guarantee success”. Testing with users traditionally focus on the first and fourth layers for the obvious reason that the user is directly involved in these layers. However, these layers are also the most dependent on specific (killer) applications.

Position But, even if there is no killer application for a specific technology, it could still be beneficial to investigate aspects of the technology that are intrinsic to the technology; even when there was no killer application for hypertext, one

could already research link structures and even importance of documents based on their incoming links. Furthermore, there are indicators that context-awareness as a technology is usefull; for example for changing the language of a website (based on the language settings of the visting computer) and for automatically opening doors (when a person is approaching). Combining these observations we claim that it is usefull to define, research, and evaluate aspects of context-awareness that are application independent.

As indicated in our previous work [3], we identified explainability as one of the major requirements for context-aware systems; meaning that a user should be able to get an explanation why and how the results were adapted to the context. In Fig. 1 this means that the properties inferred in step (2) should be explainable to the user. We claim that the explainability of these properties is an example of such an aspect that can be evaluated independent of the application.

Our case To store the user properties our framework uses a set of rules consisting of context properties and document properties together with a weight. Taken together, these rules determine the belief that the system has about the preference of a user. These user properties allow us to both evaluate the representation language of the context and document properties as well as the combination of the weights into a preference of the user.

Since the weigths of the rules are less dependent on the application than the representation language, we decided to evaluate only the weights of the rules. Also, in order to be able to evaluate the system in limited time we focus on rules that are explicitly entered by us (taken from MySpace user profiles, so they present opinions of real people). This also gives us the possibility to to have rules that are applicable during the evaluation. Together, these design choices allow us to present the rules in natural language where the task of the user is to give a weight to the rules in such a way that she thinks that it represents a model of her preferences.

Finally, we decided to use predefined queries to focus on the result adaptation part (because if the query is posed in the

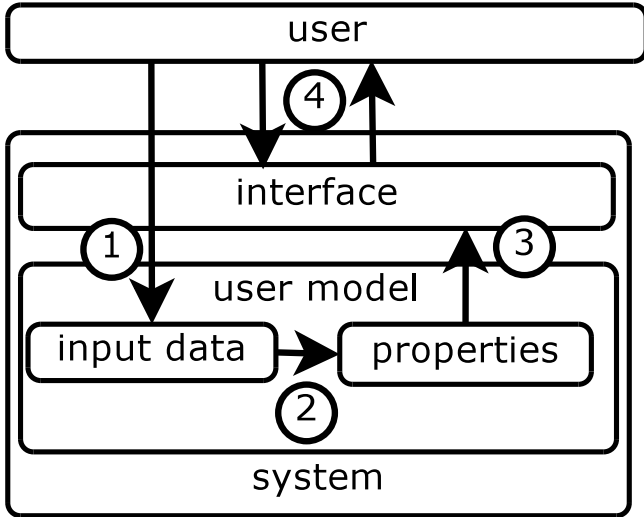


Figure 1: Layers for evaluation of adaptive systems from Weibelzahl and Weber.

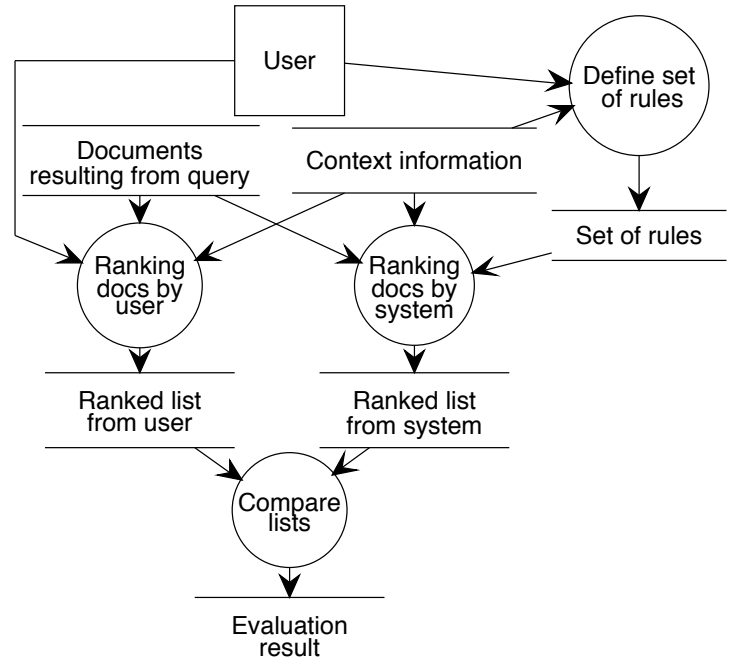


Figure 2: Evaluation process.

same expressive language as the rules, we more or less test to which extend a user has forgotten the rules he entered, and if it is not, we just measure the difference in expressiveness between the rule language and query language). These queries were in the area of TV programs, tourist destinations, and music. This results in the evaluation process in Fig. 2.

Thirty-two people took part in our evaluation. We have asked them, for all three queries, to give weights to individual rules and after that rank the set of query results. The ranks of the users were compared to different methods of weight combination were the baseline was formed by taking the mean rank of all other users (a kind of collaborative filtering). The outcome was that taking the weights of the rules contains more (useful) information than only taking the relative rank of the rules, collaborative filtering performs better than only using the weights, but a combination of weighting rules and collaborative filtering outperforms all other methods in providing the best first system result. Additional survey questions resulted in the outcome that many users saw benefit for adaptation and many users indicated that they would be prepared to fill in a profile and update it if it would lead to better query answers.

Conclusion In short, our message is that usefull results from evaluating context-aware systems with users are diffi-

cult without being application specific. However, there are some partial exceptions of which one is the evaluating the explainability of the user model. We showed the feasibility by providing a specific example of evaluating the scoring of preference rules. This particular evaluation indicated that users are able to provide scores for their preferences that improve query results upon (good) baselines. Combined with the willingness of users to provide profile information and edit their model, it seems to reinforce our indication that explainable adaptive systems are possible and usefull.

References

- [1] G. D. Abowd and E. Mynatt. Charting past, present, and future research in ubiquitous computing. *ACM Trans. Comput.-Hum. Interact.*, 7(1):29–58, 2000.
- [2] P. J. Brown, W. Burleson, M. Lamming, O.-W. Rahlff, G. Romano, J. Scholtz, and D. Snowdon. Context-awareness: Some compelling applications. In *CHI Workshop on The What, Who, Where, When, Why and How of Context-Awareness*, 2000.
- [3] A. H. van Bunningen, L. Feng, and P. M. G. Apers. Context for ubiquitous data management. In *International Workshop on Ubiquitous Data Management (UDM2005)*, April 2005.
- [4] S. Weibelzahl and G. Weber. Advantages, opportunities and limits of empirical evaluations: Evaluating adaptive systems. *KI*, 16(3):17–20, 2002.