Generating and Presenting User-Tailored Plans

Detlef Küpper
Dept. of Computer Science, Univ. of Applied Science Aalen
D-73433 Aalen, Germany
Detlef.Kuepper@fh-aalen.de

Alfred Kobsa
Dept. of Information and Computer Sc., Univ. of California
Irvine, CA 92697-3425, U.S.A.
kobsa@ics.uci.edu

Abstract

The paper describes methods for generating user-tailored advice and for suitably presenting it to users, taking their capabilities and knowledge into account.

Keywords

User-tailored advice, plan presentation, user modeling

INTRODUCTION AND OVERVIEW

The effectiveness of advice-giving systems essentially hinges on users' ability to capitalize on the received advice to reach their goals. Assume, for instance, that a user must be advised how to print a file that is located on a remote host. The advice will be very different, depending on the topology of the network, and the operating systems and programs that are to be used. Also the combinatorial complexity is high. All this speaks for the need of plan generation for advice giving. In addition, however, users' knowledge and capabilities vary widely, which calls for *user-tailored* advice generation.

The problem of whether a user can understand an advice already received considerable attention in the literature (see, e.g. [7] for a survey). An additional prerequisite for the success of advice is that the user must be able to execute it, i.e. have the capabilities that are necessary to perform each step of the advice. Capability in this sense means the user's personal abilities and authorization to perform the actions contained in the advice. In our approach, an initial plan is therefore generated based on each user's individual abilities, which would enable her in principle to achieve her goals. In order to carry out this plan, the user may however still need additional information. In the next step, therefore, the knowledge must be determined that the user still needs beyond a listing of the individual plan steps. Thereby, (typical) user misconceptions about the executability of individual actions must be considered if they can lead to erroneous plan execution. If several ways exist in which users can achieve their aims, it also makes sense to select an option that requires few additional explanations only. Since a generate-and-test approach (which generates all possible user plans and then filters out those that are too cumbersome for the user) is too resource-consuming, certain aspects of plan presentation should already be taken into account during plan generation.

USER CAPABILITIES

The way in which plan knowledge is represented in the field of user modeling (specifically in plan recognition, see e.g. [9]) is not sufficient for plan presentation purposes. One reason is that plan recognition algorithms do not have to take the relationship between individual plan steps into account, and therefore mostly do not represent the preconditions and effects of plan steps. These are however the crucial components for plan generation. Plan recognition moreover does not have to distinguish between users' capabilities and knowledge since users who are being observed carrying out plans are seemingly able to do so and have knowledge of them. For the generation of user-specific plans, the difference between the principled ability to carry out plans (including the permissions to do this) and the knowledge how to do this is crucial. The lack of the latter can be remedied during plan presentation, while the lack of the former cannot be resolved by mere advice.

Consequently, the user model of our system also reflects this distinction and allows for the representation of whether or not a user knows an action, and independently, whether or not he is able to carry it out. Fig. 1 shows (in a very simplified manner) the user model knowledge base that represents system assumptions about user beliefs (in SBUB), user capabilities (SBUC), stereotypes, and system domain knowledge (SB). The user model knowledge base is represented in BGP-MS [4], taking advantage of its knowledge partitioning mechanism. Within each partition, a terminological knowledge representation system is employed [3]. BGP-MS offers convenient mechanisms for the construction, maintenance and revision of user model contents based on observations about the user. Concepts also form the basis for the communication between the system and the user.

In [5][6], we show in detail how planning knowledge can be represented in this formalism as well. The central idea is to describe the abilities of the user through plan operators, and leave the generation of plans to a classical AI planner (UCPOP [8] in our case). This is done by making only those plan operators available to the planner which are contained in the capabilities part of the user model. The generated plan then only contains steps that the respective user is able and authorized to execute. The planner thereby does not have to be modified, and the planning process is independent of a specific planner.

For printing our example file that is located on a remote host, the user must first transfer it into his local file system and then print it through his word processor. Depending on the permissions of the groups to which the user belongs, this can be achieved by mounting a directory on a remote host, downloading the file through ftp, or copying the file
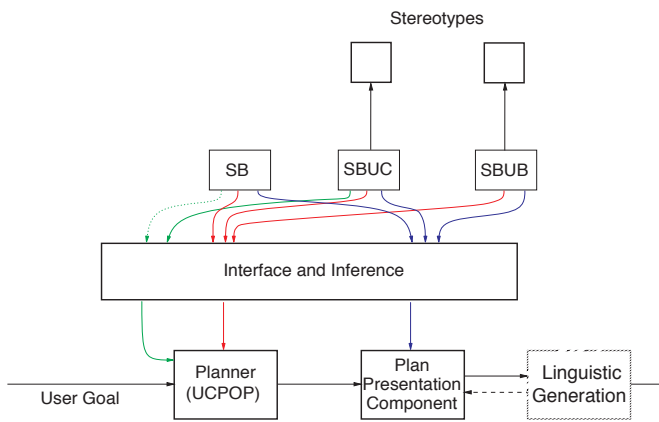
Fig. 1: System components for user-tailored advice generation

to a diskette and transferring it physically. Different plans can be generated depending on the user's permissions.

PLAN PRESENTATION

Plan presentation should not only to inform the user about the individual steps of a suggested plan, but also supply him with knowledge that she possibly lacks. Moreover, users normally would like to know the purpose of each step in the plan (see [2]). In our model, knowledge about causal links between steps is created during the plan generation process, and communicated to the user.

A third way in which user-tailored generated plans still have to be personalized during plan presentation is through user-specific decomposition of plan steps into sub-steps. We assume that the operators that are used in the planning process are described in such detail that all dependencies and interactions between plan steps are accounted for. However, it does not make sense to use extremely low-leveled plan operators (e.g., in our networking example, at the level of finger movements). This would on the one hand tremendously increase the planning efforts (which grow exponentially with the plan length). On the other hand, such low-level plans also would hardly be acceptable to users, *unless they are unfamiliar with these low-level substeps*. The appropriate decomposition of plan steps is therefore user-dependent.

We allow one or more decompositions to be associated with plan operators. They have themselves the structure of plans and describe how the superordinate plan operator can be carried out[1]. They constitute a sort of *recipe* for the execution of the plan. Such recipes can be recommended to the user if the system assumes that he has no reliable knowledge how to execute a plan step. Our definition of reliability is recursive since sub-steps can themselves be decomposed into even lower sub-steps:[2] a user has reliable knowledge of a plan step if it is contained in his knowledge model and marked as *atomic* (i.e., not further analyzable by the system); or if he only knows correct decompositions of the plan step, at least one of which he is

able to perform and has reliable knowledge about.

If the user does not have reliable knowledge about a plan step or decomposition, a recipe is given to him based on his knowledge and his capabilities. If necessary, recipes are moreover given for individual steps of this decomposition. We use a recursive algorithm for plan decomposition which terminates when it finds that the user has sufficient knowledge about each step of the current plan presentation. If more than one recipes exist for a plan step which the user can carry out, the algorithm selects the recipe with the best rating. We currently assess the length of the recipe (including all necessary decompositions). Other criteria would be user preferences for certain plain steps or users' previous reliability carrying out certain steps.

An enhancement of our model considers the ratings of plan steps already in the plan generation process. The current plan candidates are thereby rated in each planning cycle, and only the best candidate is further processed in the subsequent cycle. As long as the overall rating monotonously decreases with increased plan length, no possible solution becomes ruled out and the first solution will be the one with the best rating. Several planners, including UCPOP, support the inclusion of such strategies

References

[1] Erol, K.; Hendler, J. and Nau, D.: Semantics for hierarchical task-network planning. Tech. Report CS-TR-3239, Computer Sc. Dept., Univ. of Maryland, 1994.

[2] Gilbert, G.N. and Jirotka, M.: Planning procedural advice. *Interacting with Computers 2*, 1990, 313-329.

[3] Kobsa, A.: Utilizing knowledge: The components of the SB-ONE knowledge representation workbench. In *Principles of Semantic Networks*, J. Sowa (ed.), Morgan Kaufmann, San Mateo, CA, 1991, 457-486.

[4] Kobsa, A. and Pohl, W.: The user modeling shell system BGP-MS. *User Modeling and User-Adapted Interaction 4*, 2, 1995, 59-106. http://www.ics.uci.edu/~kobsa/papers/1995-UMUAI-kobsa.pdf

[5] Küpper, D. and Kobsa, A.: User-tailored plan generation. In *Proc. 7th International Conference, on User Modeling* (Banff, Canada), 1999, 45-54. http://www.ics.uci.edu/~kobsa/papers/1999-UM99-kobsa.pdf

[6] Küpper, D.: User modeling for user-specific plan generation and presentation (in German). Dept. of Math. and Computer Sc., Univ. of Essen, Germany, 2002.

[7] van Mulken, S.: *User Modeling for Multimedia Interfaces: Studies in Text and Graphics Understanding*. Dt. Univ.-Verlag, Wiesbaden, Germany, 1999.

[8] Penberthy, J. and Weld, D.: UCPOP: A sound, complete, partial order planner for ADL. In *Principles of Knowledge Representation and Reasoning: Proceedings of the 3rd Conference (KR'92)*, 1992, 103-114.

[9] Weida, R. and Litman, D.: Terminological reasoning with constraint networks and an application to plan recognition. *Proceedings KR'92*, 1992, 282-29.

---

[1] This is comparable with HTN planning (cf. e.g. [1]), except that we assume that plan sub-steps do not interact.

[2] Normally though plan steps are only decomposed into one, at most two, lower level plans.