User-Tailored Plan Presentation

Detlef Küpper[1] and Alfred Kobsa[2]

[1] GMD FIT, German Nat'l Research Center for Information Technology,
D-53754 St. Augustin, Germany, kuepper@gmd.de
[2] Dept. of Information and Computer Science, University of California,
Irvine, CA 92697-3425, U.S.A., kobsa@ics.uci.edu

Abstract. This paper discusses plan presentation, the second phase in user-tailored advice giving. Its main task is to determine what knowledge must be provided to ascertain that the user comprehends the plan and is able to perform it, even if he detects unexpected obstacles. Plan presentation is guided by a model of the user's knowledge and of his capabilities to perform actions in the domain. Finally we describe how to bias the plan generation process to prefer plans that contain as little information unfamiliar to the user as possible.

1 Introduction

In [2] we introduced a plan generation algorithm for advice-giving systems that exploits a model of the user's capabilities (namely his physical abilities and his authorization to perform actions) to produce plans that are *in principle* executable by the user. In order to perform the plan, he may however still need additional information. This possibly missing knowledge – or in general the gap between a user's capabilities to perform plans and his knowledge how to do this – determines the scope of user-tailored advice. The main task of *plan presentation* is to identify and supply the knowledge that the user still needs to perform the plan.

We focus on two types of knowledge. *Structural information* of the plan is needed to comprehend the functionality of plan. Additionally it supports the user's replanning in case unexpected obstacles arise during plan execution. Furthermore, the user needs to know *how to perform* the steps of the plan. Such knowledge is however only useful if the user is able and authorized to carry them out. Thus the presentation process must consider both the user's knowledge and his capabilities.

2 Determining the Contents for Plan Presentation

A plan is an abstract object that usually needs further elaboration by the user who will execute it (cf. [4]). Often the user must decompose plan steps into substeps that he can execute directly, or has to modify the plan to remedy unexpected obstacles during plan execution. For both tasks, the user needs to comprehend the plan's functionality, i.e. he must know the role of each step in the plan. Our planning process represents this knowledge by *causal links*. Each of them links a step that satisfies a

M. Bauer, P.J. Gmytrasiewicz, and J. Vassileva (Eds.): UM 2001, LNAI 2109, pp. 243–246, 2001.

condition with the plan's goal or with a step that requires this condition for its executability. Any further elaboration of the plan must not to destroy these links since this may jeopardize the plan's executability. If the user cannot execute a step of a plan directly, he must *know how* to decompose it into smaller sub-steps. The presentation process decides on the basis of the user model whether or not the user needs additional or correcting explanations. This user model extends the terminological representation of plan operators (so called *plan concepts*) described in [2] by decompositions that may be assigned to plan concepts. Similar to HTN-Planning (e.g. [1]), these decompositions are (simple) plans. Their steps are instantiated plan concepts like the steps of the generated plan. We assume that the generated plan considered all the conditions and dependencies that are relevant in the domain. Therefore we may ignore individual preconditions and effects of the steps in a decomposition and look at a decomposition as a recipe whose executability depends on the precondition of that plan concept only to which it belongs.

Whether or not the *current user* can execute *at least one* decomposition of a plan concept depends on his capabilities with regard to the steps of the decomposition. We call a decomposition *usable* (for the current user) if he is able to perform all its steps, and each step either is *atomic* or the system can provide a usable decomposition for it. *Atomic* means that the system assumes that the user has extensive competence in performing the plan concept (and its instantiations) or can execute it directly. A user has *reliable* knowledge of a plan concept if he does not know any false decomposition (compared to the system's knowledge) but at least one usable decomposition that contains atomic steps only, or steps only for which the user has *reliable* knowledge. The absence of wrong assumptions prevents the user from choosing an erroneous way to perform the plan concept. For all steps of the plan

---

present-plan (*p*:plan)
(p1) set *effort* initially to (| *steps* (*p*) | + 2 * | *causal-links* (*p*) |) * k_{pres}
(p2) set *presentation* initially to *linearize*(*p*)
(p3) for each *s* ∈ *steps* (*p*) add the name of *plan-concept*(*s*) to s in *presentation*
(p4) for each *pc* ∈ *plan-concepts* (*p*)
(p5) add result of *present-plan-concept*(*pc*) to *presentation*, *effort*

present-plan-concept (*pc*:plan-concept)
(c1) initially set *presentation* to {} and *effort* to 0
(c2) if user doesn't know *pc*
(c3) set *presentation* to *descript*(*pc*) and *effort* to *effort*(*descript*(*pc*))
(c4) if not (*atomic*(*pc*) or user has reliable knowledge of *pc*)
(c5) add result of *present-recipe-for-plan-concept*(*pc*) to *presentation*, *effort*

present-recipe-for-plan-concept (*pc*:plan-concept)
(r1) let *ec* be the set of usable decompositions of *pc* acc. to system's knowledge
(r2) if *ec* = {} → set *effort* to ∞ (plan presentation fails)
(r3) else set *presentation*, *effort* to results of *present-plan* (*d*) where
(r4) $d \in ec \wedge effort(present\text{-}plan(d)) = \min_{(d' \in ec)} effort(present\text{-}plan(d'))$

Fig. 1. Plan presentation algorithm

for which the user has reliable or atomic knowledge, he does not need any explanations. For all other steps, the presentation process of our system provides a usable decomposition.

The pseudo code of Fig. 1 summarizes the procedure for determining the knowledge that the user needs to successfully execute a plan *p*. Each function returns the values *presentation* (a data structure for the contents of the presentation) and *effort* (a numerical value for the amount of information that is unknown to the user). The algorithm starts by computing the effort for presenting all steps and causal links of the plan (p1). Since causal links are complex components, each contributes two basic effort units k_{pres} to this value. A linearization of the plan's data structure (p2) with a name for each plan step (p3) forms a basis for the presentation. For each plan concept that was used for the plan (p4), the user will obtain a canned description if he does not know it (c2,c3), and user specific information about how to perform it if he does not have reliable knowledge (c4,c5). This information is determined by selecting from the set of usable decompositions of the plan concept (r1) the decomposition with the minimal effort (r4). The presentations of these decompositions are computed by the function *present-plan*. The depth of these recursive calls is limited by the depth of the decomposition hierarchy of the plan concepts. Line (r2) handles the case in which the system cannot give a usable explanation. This possibility is a consequence of the incomplete knowledge of how to perform plan concepts. The subsequent enhanced plan generation process excludes such *unexplainable* plan concepts from further consideration.

Up to here our discussion separated the two phases of user-tailored advice. We can however identify two starting points for considering presentation aspects already during plan generation, to obtain plans that lead to better presentations. First, we exclude unexplainable non-atomic plan concepts from the planning process, because their presence in a plan always leads to a rejection of the presentation. Moreover, we influence the decisions of the planner so that it prefers plan concepts with low presentation effort for the current user. This effort may be computed by the function *present-plan-concept* (see above) before the planning process starts. As a result, plans with low presentation effort are *preferred*, but we do not loose any solution. Specifically, we do not loose the opportunity to give explanations to users. This technique of biasing the planning process can also be exploited for taking into account users' preferences for performing certain plan concepts, their practice in performing them, or the general probability of performing them successfully.

3 Summary and Further Development

We described the presentation of an already generated domain plan to a user. We claim that the user needs structural information to comprehend the plan's functionality and also knowledge of how to perform the plan steps. Although the user may already have such knowledge available, it might be wrong, incomplete or unusable, i.e. he cannot act along it. The presentation process exploits a model of the user's knowledge and capabilities to complement or correct the user's knowledge, in order to

enable him to execute the plan. Finally we modified the plan generation process of our earlier work to prefer plans that lead to presentations with little unknown information for the user.

While the presentation process described so far considered the user's knowledge as well as his capabilities, it did not take the user's (likely) inferences into account, and thus may produce lengthy results. Our approach to tackle this problem is an improvement of Young's work [5]. It takes again a model of the user's planning knowledge and capabilities into account and is described in more detail in [3].

References

1. K. Erol, J. Hendler and D. Nau. Semantics for hierarchical task-network planning. Tech. Report CS-TR-3239, Computer Science Dept., Univ. of Maryland, 1994.
2. D. Küpper and A. Kobsa. User-tailored plan generation. In *User Modeling: Proc. of the 7th International Conference, UM99*, pages 45-54, Banff, Canada, 1999.
3. D. Küpper. Benutzermodellierung für benutzerspezifische Plangenerierung und -präsentation. Dissertation, Dept. Mathematics and Comp. Science, Univ. Essen, Germany, forthcoming.
4. B. Webber, N. Badler, B. DiEugenio, C. Geib, L. Levison and M. Moore. Instructions, Intentions and Expectations. *Artificial Intelligence 73*(1-2):253-269, 1995.
5. R. M. Young. Using Grice's maxim of quantity to select the content of plan descriptions. *Artificial Intelligence 115*(2):215-256, 1999.