A DIALOG MODEL FOR OFFERING TASK COMPLETION FOR COMPLEX DOMAINS

Marc Hofmann and Manfred Lang

Institute for Human-Machine Communication
Technical University of Munich, D-80290 Munich, Germany
{hofmann, lang}@ei.tum.de

ABSTRACT

We present an approach towards dialog modeling for a plan recognition based adaptive user assistance system for complex domains. The assistance system uses a probabilistic plan recognizer's output for offering complete or partial task completion. Therefore the dialog model not only considers the most likely plan, but also takes plans into account that are similar in content. Hence the chance of offering user adequate assistance is increased. In addition we introduce a user interface for an intuitive offering of assistance.

1. INTRODUCTION

Plan recognition based user assistance systems usually follow the most likely plan for adapting the dialog and the user interface to the users' needs. However complex domains yield the risk of the plan recognizer [Hof01] evaluating a number of plans with nearly equal evaluation measures causing problems for the dialog model. We present a user interface and a dialog model for offering appropriate task completion, which stays abreast of the fact that plans with similar evaluation measures are often similar in content. Therefore the dialog model consists of a number of criterions on four different levels: the plan level, the sub plan level and the action level and command level.

The interface and the dialog model have been developed for a user assistance system to offer task completion for a standard UNIX shell. The input vector for the dialog model is created by a probabilistic plan recognizer, which evaluates all potential plans regarding the previous user actions with every new action the user takes. Each plan is assigned an evaluation measure reflecting the belief that the user follows that plan. The plan library consists of a number of UNIX tasks, which might be executed in the current file system using standard UNIX commands. The UNIX domain with its complexity, variety of plans and the virtually infinite number of ways to achieve a goal is an ideal testground for the dialog model.

2. METHODOLOGY

2.1. User interface

The user interface consists of a standard UNIX shell for the users' inputs and system responses, as well as four buttons, which will be assigned dynamically by the dialog model. By clicking on one of the buttons the user has the chance to complete his task automatically or at least to execute appropriate sub plans and UNIX actions.

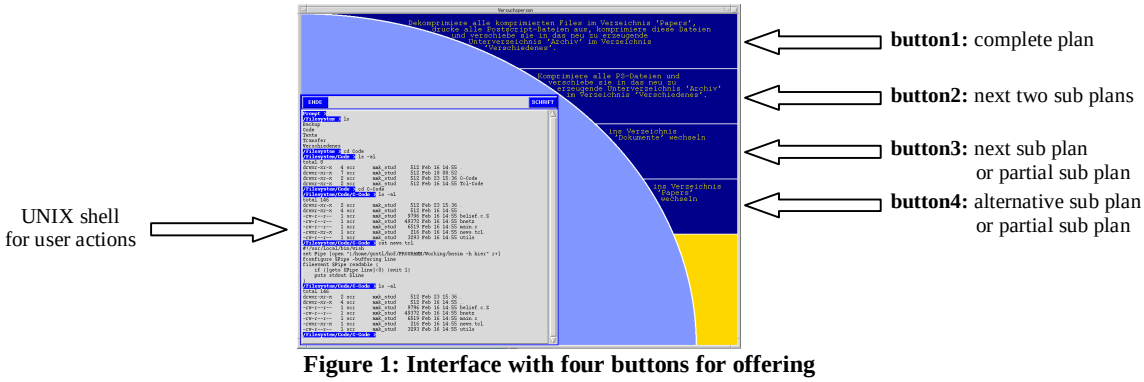

Figure 1: Interface with four buttons for offering task completion or partial task completion

From: Proc. HCI 2001
Lawrence Erlbaum Ass. NJ

Fig. 1 shows a screenshot of the interface. The size of the buttons reflects the level the help refers to. For example pressing the top button means help on the plan level, i.e. the completion of the whole plan, which is assigned to the button. The button assignments change dynamically controlled by the dialog model. Regarding the design of the interface, the assignment of the buttons and the verbalization of the help texts various usability studies have been made.

2.2. Dialog model

The basic idea behind the dialog model is creating a dynamic button assignment, which stays abreast of a preferably large number of plans. In case of similarly likely evaluation measures for various plans, the dialog model is not supposed to decide for the plan with the highest evaluation measure, instead it should offer help that sets the focus on reducing the risk that the user cannot be offered any kind of appropriate assistance. Therefore the dialog model uses four criterions, which will now be illustrated:

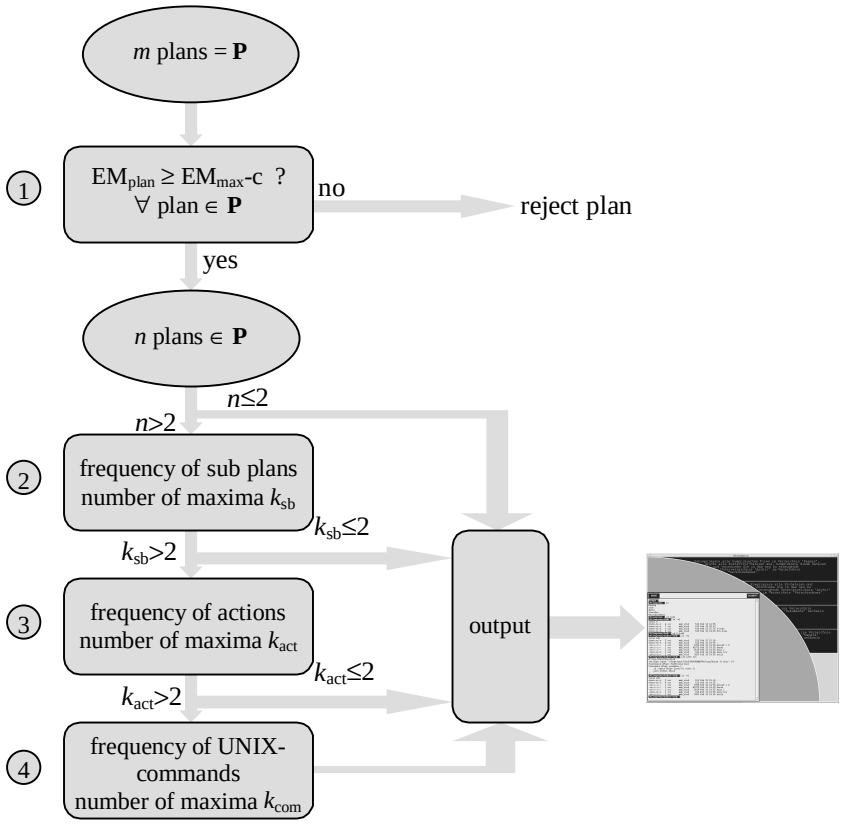

Figure 2: Basic structure of the dialog model

Fig. 2 pictures the basic structure of the algorithm with the criteria ① to ④:

① Criterion one refers to the plan level. At first each plan of the plan hypotheses space **P** with an evaluation measure EM_{plan} beneath a certain threshold will be rejected to consider only the *n*-best plan hypotheses. For determining the decision threshold we subtract a constant *c* from the maximal observed evaluation measure EM_{max}. We found 10 per cent of the maximal evaluation measure to be a reasonable value. If just two or less plans remain, the completion of the most likely plan and the next sub plans of both plans are directly assigned to the buttons of the interface. If there are more than two plans left, the remaining plans will be analysed according to criterion two.

② Criterion two refers to the sub plan level. Each plan consists of a number of sub plans. Similar evaluation measures of plans in most cases mean common sub plans or single actions. The number of occurrences of each sub plan in the remaining plans will be calculated. If there are two sub plans in the majority, their completion will be assigned to the buttons. If there are more than two maxima, the algorithm steps one level deeper in hierarchy and continues on the action level with the next criterion.

③ Criterion three refers to the action level. The algorithm determines the number of occurrences of each action in potential successive sub plans; an action therefore consists of a UNIX command and its parameters. This step can be interpreted as creating action intersections between sub plans. If there are two actions in the majority, their execution will be assigned to the buttons, if there are more than two the algorithm steps to criterion four.

④ Criterion four finally refers to UNIX commands. The number of occurrences of UNIX commands will be determined. The two commands that appear in most potential succeeding sub plans will be assigned to the buttons. If the user clicks on one of these buttons it is very valuable information for the new plan recognition process. This situation reflects the cooperative relation between the user and the assistance system.

3. RESULTS

The user assistance system, i.e. especially the user interface and the dialog model, has been evaluated in usability studies with a number of experimental subjects. 100 per cent of the subjects with little UNIX experience evaluated the system as very helpful, the system was also stated to be helpful by occasional UNIX users. Fig.3 shows the evaluation results. It has been expected that experts would judge the system to be not helpful, as experts are not the target group for that kind of assistance system.

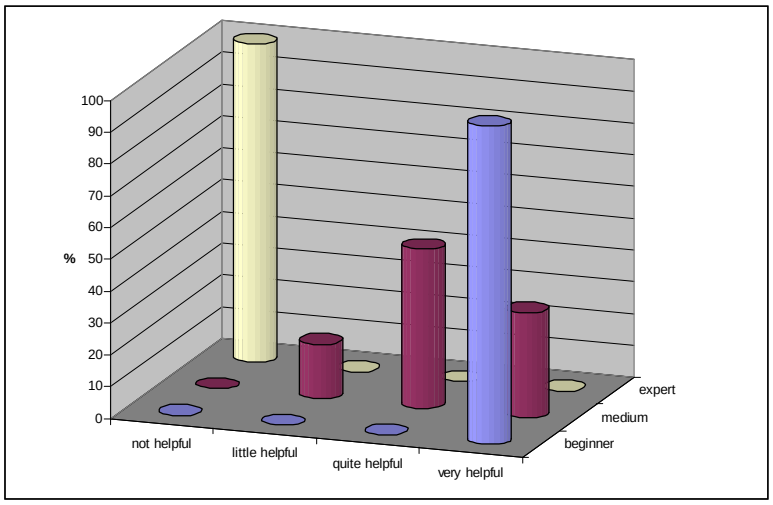

Figure 3: Evaluation of the user assistance system

4. CONCLUSIONS

The dialog model and the user interface proved to work well. Our next step is the development of an intelligent text generator for creating users adaptive and appropriate verbalizations for the button assignments.

REFERENCE

[Hof01] Hofmann, M. and Lang, M.: ”User Appropriate Plan Recognition for Adaptive Interfaces“, Proceedings HCII 2001 (New Orleans, Lousiana, USA), (this conference)