Towards the Theory-Guided Design of Help Systems for Programming and Modelling Tasks

Claus Möbus, Knut Pitschke, Olaf Schröder

University of Oldenburg, Dept. of Computational Science
Unit on Tutoring and Learning Systems,
D - W2900 Oldenburg, Germany
Claus.Moebus@arbi.informatik.uni-oldenburg.de

Abstract. This paper describes an approach to the design of online help for programming tasks and modelling tasks, based on a theoretical framework of problem solving and learning. The framework leads to several design principles which are important to the problem of when and how to supply help information to a learner who is constructing a solution to a given problem. We will describe two example domains where we apply these design principles: The ABSYNT problem solving monitor supports learners with help and proposals for functional programming. The PETRI-HELP system currently under development is intended to support the learning of modelling with Petri nets.

1. Introduction

It has been well recognized that developing intelligent help systems raises difficult questions, like: How is help and instructional material to be designed? When should remedial information be supplied, and not be supplied? Why might the same information be useless to one person and helpful to another? Existing intelligent tutorial and help systems have not always provided satisfactory answers to such questions. For example, the information delivered to the learner may assume too little or too much knowledge, or the user interaction is too restrictive, or tutoring and help strategies are unprincipled and ad hoc [25]. These shortcomings are basically still true [24, 31]. In order to move forward, a theoretical framework is necessary which incorporates problem solving and learning. It should be detailed enough to enable specific design decisions and predictions. At the same time it should be general enough to be applicable to different domains.

We work on such a framework, ISPDL Theory (*i*mpasse - *s*uccess - *p*roblem solving - *d*riven *l*earning). It is an attempt to integrate the theoretical concepts of *impasse-driven learning* [7, 14, 15, 26, 27], *success driven learning* [1, 2, 4, 16, 33], and *problem solving phases* or action phases [11, 12]. One purpose of ISPDL Theory is to obtain a set of design criteria for intelligent help systems which support problem solvers while planning and constructing solutions. In order to make these design criteria as domain independent as possible, we apply them to two different domains: The ABSYNT problem solving monitor supplies help for functional programming. The PETRI-HELP system is designed to support Petri net planners. In this paper we will describe ISPDL Theory, the design principles implied, and how they are to be realized in ABSYNT and PETRI-HELP.

2. The ISPDL Theory of Problem Solving, Acquisition and Improvement of Knowledge

From empirical investigations we concluded that it is fruitful to describe learning in the domain of functional programming as an interplay of impasse- and success-driven learning. In particular, we developed a model based on these concepts which closely simulates the continuous stream of actions and verbalizations of a single subject while acquiring the interpreter knowledge about functional programs [23]. Further development led to the ISPDL Theory [20] which is intended to describe the stream of actions and cognitive processes occurring in problem solving situations. ISPDL Theory has three aspects:

• The distinction of *different problem solving phases* [11]. In the *deliberate* phase the problem solver considers several goals and finally chooses one. In the *plan* phase a solution plan is developed to obtain the goal. Subgoals are created and sequenced. Then the plan is executed, or *implemented*. Finally the problem solver *evaluates* the result.

• The *impasse driven acquisition of new knowledge*. In response to an impasse, the problem solver applies weak heuristics, like asking questions and looking for help [26, 27, 28]. Thus the learner obtains new information. As a result of this, the learner may overcome the impasse and acquire new knowledge. Thus impasses trigger the *acquisition* of knowledge. But the new information may cause a secondary problem [7, 10].

• The *success driven improvement of existing knowledge*. Successfully used knowledge is improved. By rule composition [2, 16, 22, 30], which can be based on the resolution method [20], the number of control decisions and subgoals to be set is reduced.

3. Principles for Help Design Based on ISPDL Theory

The ISPDL Theory motivates the following principles for providing help to the learner:

1. The help system should not interrupt the learner (see also [32]) but *offer* information, because according to the theory, information is only helpful at impasse time [26]. So information is only to be supplied on request by the learner. This principle is somewhat opposed to the principle of immediate feedback [3, 5]. But it is implied by the theory, and we think that it is important to let the learner develop her /his own solution ideas even if they seem strange from an expert point of view.

2. The learner must have the opportunity to obtain *detailed feedback and information* at every time impasses may arise in problem solving. Since different impasses are possible at *different levels* of problem solving, the system must offer support in the problem solving phases of *planning, implementation*, and *evaluation*.

3. The learner should be enabled to *make use of her/his pre-knowledge* as much as possible when asking for help, so the information provided as help does not suggest different solution plans and thus cause secondary impasses. Rather, help should accept the learner's solution plan and provide the learner with requested information as precisely as possible.

4. The provided information should be *tailored to the actual knowledge state of the learner*. If the information presupposes too much pre-knowledge, the learner will encounter a secondary impasse. This might lead to self explanation [8, 29] of the information obtained, but also to non-understanding and to negative emotions. If the information assumes too little pre-knowledge, then the learner will get bored by things already known. So whether information is helpful depends on the actual knowledge state of the learner. A *state model* is needed to represent online the actual hypothetical domain knowledge state of the learner. Its two main functions are to *control* the analysis of solution proposals of the learner, and to determine which help information to *choose* in case of several possibilities.

5. The state model should be embedded in a *process model*. The latter models the *processes of knowledge acquisition and modification, the application of weak heuristics and control processes*. It may make use of additional data, like verbalizations. One of the functions of the process model is to support the development of the more restricted state model which must be empirically valid since it is used for diagnosis and help generation.

6. It is necessary that *the learner is free in the choice and sequencing of her/his interactions* with the system. The more restricted the range of the learner's actions is, the less information can be obtained for inferring cognitive states (modelled by the state model) and processes.