

Computer-supported training of psycho-diagnostic skills

Westera, W., Hommes, M. A., Houtmans, M., & Kurvers, H. J. (2003). Computer-Supported Training of Psychodiagnostic Skills. Interactive Learning Environments, 11 (3), 215-231.

Postal address for all authors:

Dr. Wim Westera

Educational Technology Expertise Centre

Open University of the Netherlands

P.O. Box 2960

6401 DL Heerlen

The Netherlands

Telephone + 31 455762408

Fax: + 31 455762800

e-mail: wim.westera@ou.nl

Computer-supported training of psycho-diagnostic skills

Abstract

This paper outlines and discusses a self-contained multimedia-training program on psychological diagnostics. The program aims to play an intermediate role between theory and working practice, while offering students a simulated environment to prepare for practical work with real humans. The program simulates the entire diagnostic cycle in convincing, authentic cases while offering extended support facilities. Design principles of the program strongly rest on theories of constructivism and experiential learning; it combines freedom of movement for the students to allow explorative learning strategies, while it offers extended and intelligent feedback facilities that comment on the students' performances. The functional program design is presented and substantiated from a pedagogical perspective. Preliminary results of a survey amongst students are presented and discussed.

Keywords: media in education; multimedia/hypermedia systems; simulations; interactive learning environments; computer-based training; complex skills; competencies

Introduction

Diagnostic skills are of great importance for psychological practitioners, be it in choice of school system, choice of profession or career counselling. Because it always involves the assessment of an individual's qualities, capabilities and personality, psychological diagnostics may have far-reaching consequences for the way subjects arrange their lives. In psychology curricula quite some effort is given to diagnostic skills training. However, the complexity of psychologic diagnostics makes high demands to educators. Psychologic diagnostics involve the integration of various knowledge domains (learning theories, test and assessment theories, test selection, testing procedures, ethics, interviewing, occupational demands, etc.). Applying this knowledge in direct interaction with real human clients is not without a hitch.

As is the case in medical education, it is difficult to create sufficient opportunities for diagnostic skills training. Clearly, to let students carry out diagnoses at real clients may be quite delicate. If at all, such training contexts show a number of drawbacks that can hardly be avoided. Mostly, students are constrained to passive forms of apprenticeship learning: they observe expert behaviour rather than decide by themselves. Interruptions during a client interview to reflect on the quality of diagnostic decisions are hardly possible without affecting the quality of diagnostic services. Consequently, the learning efficacy of such practical work may be questioned, despite the one-to-one linking of students to experts.

This paper focuses on the educational design of the situational multimedia program DIAGNOST. This program aims to prepare students in a better way for their confrontation with real clients by simulating the full diagnostic cycle: the student taking up the role of psychologist has to diagnose a number of "built-in" clients in a systematic way. Here, the student is allowed to make mistakes and to find out what the effects of faulty decisions are. The program is assumed to bridge the gap between theory and practice; it prepares the students for their first confrontation with real clients and thus improves the quality and efficacy of this practical work. DIAGNOST is now being used in academic degree programmes in psychology of three universities in the Netherlands. Also the approach has been copied and adapted to allow for cases on remedial education on behalf of teacher training in the Netherlands. In addition, a variation of DIAGNOST has been developed in co-operation with various universities in Flanders (the Casekit project). To conclude this paper, we will briefly discuss preliminary findings with respect to the adequacy of the program and the appreciation by the students.

Educational context

The computer practical DIAGNOST has been developed by the Open University of the Netherlands in co-operation with the University of Groningen and the IJselland College. The program is part of a 120 hours distance education course on psychological diagnostics that is used in various (academic) degree programmes on psychology. The course aims to teach students how to carry out diagnostic examinations in accordance with scientific standards and methodologies. After having studied this course students will be able to reveal and formulate diagnostic questions to be examined, to select appropriate tests, to interpret the associated test results in view of the clients' situations, to draw conclusions about the diagnostics questions and to specify sensible measures and advices for intervention. Target groups are students and practitioners in psychology as well as mental health workers who want to upgrade their diagnostic skills. To this end, the course provides a basic endorsement of psychologic diagnosis as authorized by the Netherlands Psychology Institute (NIP). Subscribers to the course are assumed to possess some general knowledge on psychology, test theory and statistics as well as interview skills.

The 120-hours course consists of three parts: printed materials, the computer practical and face-to-face training sessions. During the first 65 hours the students study the self-instructive printed materials, which come out with relevant background knowledge (history of diagnostics, ethical problems, controversies, theories, models, psychometrics,

possibilities and restraints of tests on intelligence, personality, interest, capabilities and health determinants, standardisation and interpretation of test outcomes). In the subsequent 25 hours students have to work through the three cases of the computer practical. Here the students get acquainted with the diagnostic methodology and learn to apply the theoretical concepts in realistic situations. After this stage of individual preparation, students will extend their practical skills in face-to-face diagnostic sessions with fellow-students or eventually real clients (30 hours). Indeed, the computer practical is supposed to play an intermediate role between theory and working practice.

Diagnostic content

The diagnostic method that is subject of the course focuses on the testing of hypotheses; it is strongly based on the diagnostic framework of de Bruyn (Bruyn, Pameijer, Ruijsenaars & Aarle, 1995), though slightly simplified. The method is divided up into five subsequent phases, each of which has a different goal and each of which involves different activities.

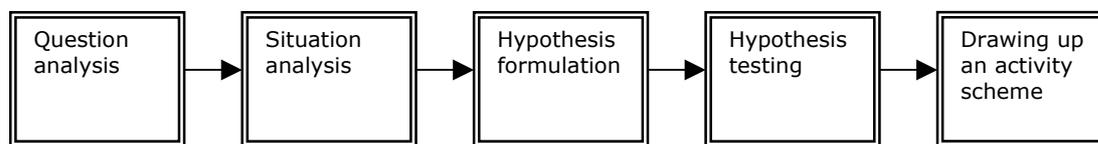


Figure 1. Subsequent phases in the diagnostic cycle.

In the phase of question analysis the diagnostician has to find out exactly what the client's help question is. In the phase of situation analysis the help question has to be translated into one or more concrete research questions to be investigated. Next, one or more hypotheses are formulated for each question and appropriate tests are selected. When test results have become available, it is necessary to decide on acceptance or rejection of the hypotheses. Finally, the client's help question has to be answered and relevant measures have to be proposed.

In each phase the diagnostician is assumed to work through a number of well-specified activities. This systematic, protocol-like approach aims to avoid impulsive and intuitive short-cuts and to improve the validity and reproducibility of the process.

Pedagogical principles and requirements

The program design is strongly based on theories of problem-based learning: these hold that learning is an active and constructive process that is facilitated by providing students with an adequate problem or case study that preferably reflects the problems they will have to face in real practice (Barrows & Tamblyn, 1980). This problem-based approach is seen as being highly motivating in that it reflects a natural way of learning. In accordance with theories for constructivist and experiential learning (Kolb, 1982), the acquisition of knowledge (and skills) is regarded as individual experiences that strongly depend on the characteristics of the learning environment, the learning activities and goals of the learner (Brown, Collins & Digid, 1989; Duffy & Jonassen, 1992). From this point of view it is indicated that the learning environment should be sufficiently rich and complex to allow for sensible experiences. It should also offer well-defined learning tasks addressing clear, operationalised learning objectives. The learning tasks should preferably allow for versatile, explorative and individualized elaborations. The learning itself is regarded as an active process of knowledge construction rather than a passive process of being exposed to information. It should include the active manipulation of attributes of the learning environment to facilitate acquisition of knowledge and understanding at operational levels (Westera, 1998; Westera, 1999).

It is clear that such design principles which support student control, free manipulation, independency, initiative and self-reliance fairly match the characteristics and needs of the student populations we described before. In addition to the domain related skills, the

program will strongly stimulate the acquisition of higher level cognitive skills (Flavell, 1973), such as critical thinking, creative thinking, analytical thinking, reasoning, dealing with conflicting data, reflection and evaluation.

In view of these pedagogical starting points, the program design should take into account the following issues:

How to incorporate real practice?

How to create sufficient freedom of movement?

How to track and assess the students' performances?

How to arrange effective student guidance and support?

How to realise a feasible user interface?

In the next sections we will explain how we have dealt with these questions in our program design.

How to incorporate real practice?

In the computer practical the students have to work their way through three authentic cases. The students, taking up the role of psychological diagnostician, are confronted with three clients who ask for psychological advice or assistance. Realistic interaction with the clients is simulated through the use of video streams. All three clients appear to have a different problem:

- Ellen Smit (29 years old) is going to lose her job as an office worker. She is not sure what kind of work she might like to do and she seems to lack insight into her capabilities.
- Edward Schövinck is a 19 years old schoolboy that has problems with choosing the right school system and a future profession. A diagnostic investigation is needed to clarify his desires and possibilities.
- Fransien Vermeer (31 years old) has been referred by the general practitioner for a psychological diagnosis. Her physical complaints could not be traced back to physical causes. A psychological component is suspected, though the nature of this psychological component is still unknown.

In each of the cases the students have to work through the whole diagnostic cycle and end up with a report with conclusions and proposed activities. Note that in the program the operational stage of actual testing is left out: after the students have decided on what tests should be applied, they instantly gain access to the associated test results. Indeed, the program focuses at the diagnostic methodology rather than at detailed operational instructions of particular tests.

How to create sufficient freedom of movement?

The prescriptive nature of the diagnostic method inevitably reduces the student's freedom of movement. Indeed, the five phases in the diagnostic cycle make up a logical sequence of successive steps that cannot be violated. So, paradoxically, the methodological content itself seems to conflict with the desired explorative pedagogy. To create sufficient room for exploration we have abandoned the idea of offering the students a full step by step approach of the diagnostic process. We think that psychological diagnostics is in no way to be equated with the unthinking execution of a number of pre-programmed steps, as might be the case with recipes in a cookery book. Instead, it makes heavy demands upon analytical skills, reasoning skills and decision taking. As we will demonstrate below, the DIAGNOST-program provides substantial freedom of movement for the students without affecting the nature of the diagnostic protocol.

While maintaining the overall, logical structure of subsequent phases, we allow the students to choose their preferred order of tasks within each of the phases (figure 2).

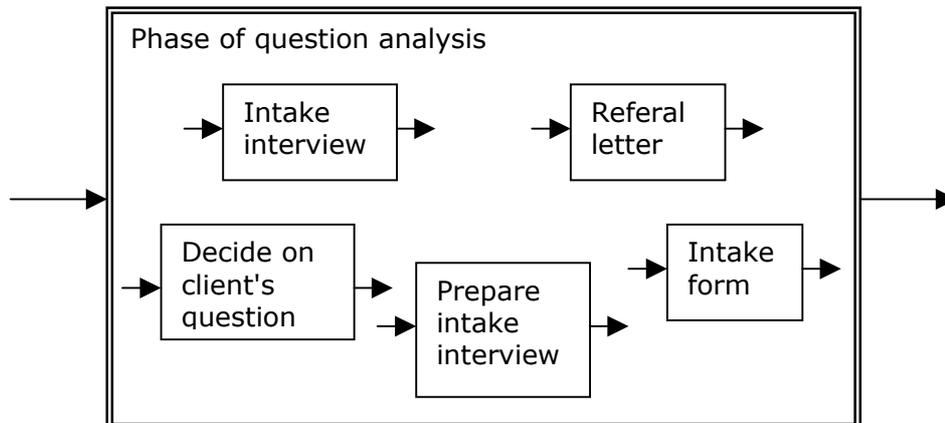


Figure 2. Diagnostic tasks in the phase of question analysis.

Although in many cases the recommended methodological approach would prescribe a fixed order of these separate activities, the students are allowed to deviate from it. For instance, students may try out to define the client's question even before the intake interview has been carried out or before the client records have been consulted. Although permitted, such inconvenient approaches cannot remain uncommented. Therefore, we have defined criteria for each activity to decide whether or not starting the activity is expedient. If it is expedient, the student will get a specific assignment (e.g. "Try to find out..."); if not, a persuasive advice to better start another activity first will appear on the screen (e.g. "You 'ld better first read the intake form, because..."). Mostly, students are allowed to be precocious and to persist in their a-typical choices. Naturally, the program puts these choices on record to allow evaluation of the students' performances. This open approach substantially enhances the students' freedom of movement: the program allows the students to make mistakes and invites them to adopt explorative attitudes.

When all activities within a particular phase have been completed successfully, the program switches to the next phase and offers the students a new set of activities. While each new phase comprises the activities of the previous phases, the environment gradually becomes more complex. In contrast with real working practice, the students are allowed to return to earlier activities at will to reconsider their decisions (which would be highly unpractical in reality).

How to track and assess the students' performances?

To be able to monitor and assess the students' performances it is necessary for the program to unambiguously record and evaluate the students' inputs. To this end, all of the diagnostic activities that the students has to carry out were classified into one out of four activity classes:

I. Consult client records

The students should have access to relevant documents in the client's file, i.e. the client's intake form and the referral letter. At a certain stage, also documents with test results will be available. Completion of such task can be easily checked.

II. Interview the client

Direct confrontations with the clients are essential in psychological diagnoses. In the program the interview is represented by on-camera video streams (Mpeg). Rather than linear observations, the video streams make up interactive sessions

controlled by the students: the students may repeatedly choose from a number of alternative questions after which the program accordingly plays back the associated answers. This allows the students to experience a sense of reality, while they can manipulate the flow of the interview and fully profit the impact of audiovisual stimuli. Students are asked to ask relevant questions, to interpret verbal information and non-verbal cues and to decide what topics and cues should be examined further. Adequate feedback can be offered for each question.

III. Take decisions on what and how to examine

Frequently, the students have to decide how to continue the examination. For instance, they have to decide on what tests should be used, what criteria should apply, etcetera. To facilitate such decisions the students is offered a number of alternatives from which they choose one or more. So, in the program, decision taking is equivalent with selecting items from a list. This allows for providing adequate feedback for each decision of the student.

IV. Note down findings and conclusions

Usually, the output of the diagnostic process is a report summing up the findings and conclusions. Free text input by the student would be the ideal solution here. Yet, because of the complexity and unreliability of algorithms for syntactic and semantic analysis, we chose to use multiselect lists with pre-defined texts. So, writing notes or drawing conclusions is carried out by selecting relevant text strings from a list of alternatives. Provided that the alternatives comprise a number of resembling, tempting or even faulty texts, the "writing" of a diagnostic report is far from being straightforward and students will still experience satisfactory freedom of movement. Again, feedback on each choice is easily generated.

Using this set of activity classes, any single decision to be taken by the student can be commented on unambiguously by the program. To allow sensible assessment of the students' performances, case authors have been asked to classify and rate the quality of each single student's decision. For each decision the error type and the permissibility that is associated with the particular student action are marked. Furthermore the quality of the student's choice is rated in a number between -10 to +10. Averages and frequencies of particular error-types can be used to generate comments on the students' behaviours.

How to arrange effective student guidance and support?

This problem concerns the efficacy of learning: how to realize a computer program, which offers a complex, open and student-controlled learning environment, without affecting the efficacy of the learning process? That is, how to combine ill-structured, multidimensional problems on the one hand and effective support and feedback on the other to prevent disorientation of the students? How to support these activities in an intelligible way without emphasis on closeness, tutor guidance and control (cf. traditional approaches for computer-aided learning like drill and practice, or teaching dialogues, tutorials)? Scardamalia (Scardamalia, Bereiter, McLean, Swallow & Woodruff, 1989) suggests that this can only be accomplished by providing the students with suitable tools and information that allow them to analyse and decide what to do. Nathan (Nathan & Resnick, 1994) intentionally uses the term 'unintelligent tutor' here: the tutor should not claim omniscience, students themselves should take the intelligent decisions.

To arrange sufficient guidance and support in the complex, open and student-controlled learning environment, we have used a simple model that describes both support at a strategic level and support at a tactic level. Also, we have discerned four levels of feedback that can be set by the student. Furthermore, various additional support facilities have been made available.

Strategic support

Strategic support is associated with comments on the start, the progress and the termination of the separate student activities. Each of the (some 20) activities that make up the full diagnostic cycle is consistently equipped with production-rule based expert comments.

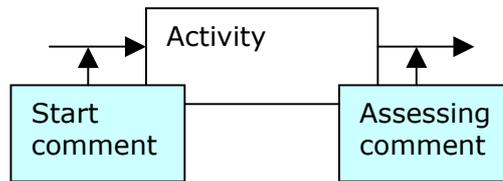


Figure 3. The "activity" as the basis for strategic comments.

Whenever the student starts an activity, the associated instructional message is generated and displayed. Such "start comment" may range from an assignment including operational objectives on the one hand to a ban that blocks the activity on the other hand. Mostly, a cautious advice is given based on the status of other activities. Similarly, for each activity the program regularly evaluates the student's performance and checks whether it meets the minimum requirements for completion. If not so, a corrective comment becomes available indicating what gaps still have to be filled. At the higher level of diagnostic phases the same strategic procedure is used to assess the student's progress and to generate the automated transition to the next phase.

Tactic support

Tactic support is associated with comments on specific choices and decisions at micro level (figure 4).

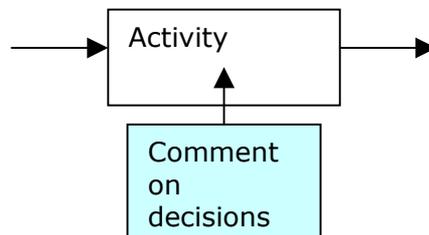


Figure 4. The "activity" as the basis for tactic comments.

Such micro feedbacks would typically look like:

- "This not the right moment to ask a question like this..."
- "No, this is not a well-formed hypothesis..."
- "Indeed, this test is a good choice to investigate this hypothesis..."
- "Yes, this text would be a correct conclusion from the test results..."

Feedback levels

In accordance with the idea of learner control, it is important to notice that the students may set the level of feedback they want to receive. The feedback texts that are presented to the student are composed of 4 levels of detail. The levels refer to:

- level 1. Minimum support, necessary to be able to proceed.
- level 2. A comment with regard to (in)correctness.
- level 3. A comment stating the reasons for (in)correctness.
- level 4. The right answer, a specific explanation or instruction.

As might be inferred from the definitions, the feedback texts are cumulative. When the student sets the feedback level, the associated feedback text is assembled out of the

particular level and all lower levels (for example, at level 3 the program displays the cumulative texts of levels 1, 2 and 3). Occasionally, the program may intervene and suggest the student to change the level of feedback to be presented.

Additional learner support

In addition to the strategic and tactic comments the program offers diverse facilities to enhance its self-instructive nature. These facilities include:

- A glossary with relevant terms in the field of psychological diagnostics; relevant terms can be printed directly.
- A hypertext structure that directly links textual expressions to its description in the glossary (pop-up).
- A status recovery system that accounts for recovery of deselected items. Having worked through part of a case, the deselection of a data root (for instance the initial help question) would discard all associated choices in the subsequent data tree (for instance on hypotheses, tests, conclusions etc). The status recovery system assures that in case of reselection of the root the whole data tree is recovered.
- A set of status files to be saved on hard disk or diskette: these files preserve the actual state of progress and allow students to interrupt and resume their sessions at will.
- A report: all actual decisions and choices are automatically recorded and listed in a report. This generated report becomes available as a standard editable text file.
- A tutorial on normalisation of test scores: since the interpretation of test scores is quite delicate a separate tutorial is available for instruction on this topic. Students that show problems during the cases are advised to work through the tutorial.
- An extended archive of texts on the psychological diagnostics. These include descriptions and specifications of various tests and test methods.
- A help facility: the program features a basic help facility on the operation of the program and its components.

How to realise a feasible user interface?

The challenge of this complex learning environment was to design a simple, straightforward user interface. The user interface is a full-colour graphical environment that represents the diagnostician's consultation room: here the student can carry out all relevant activities that are part of the diagnostic cycle.

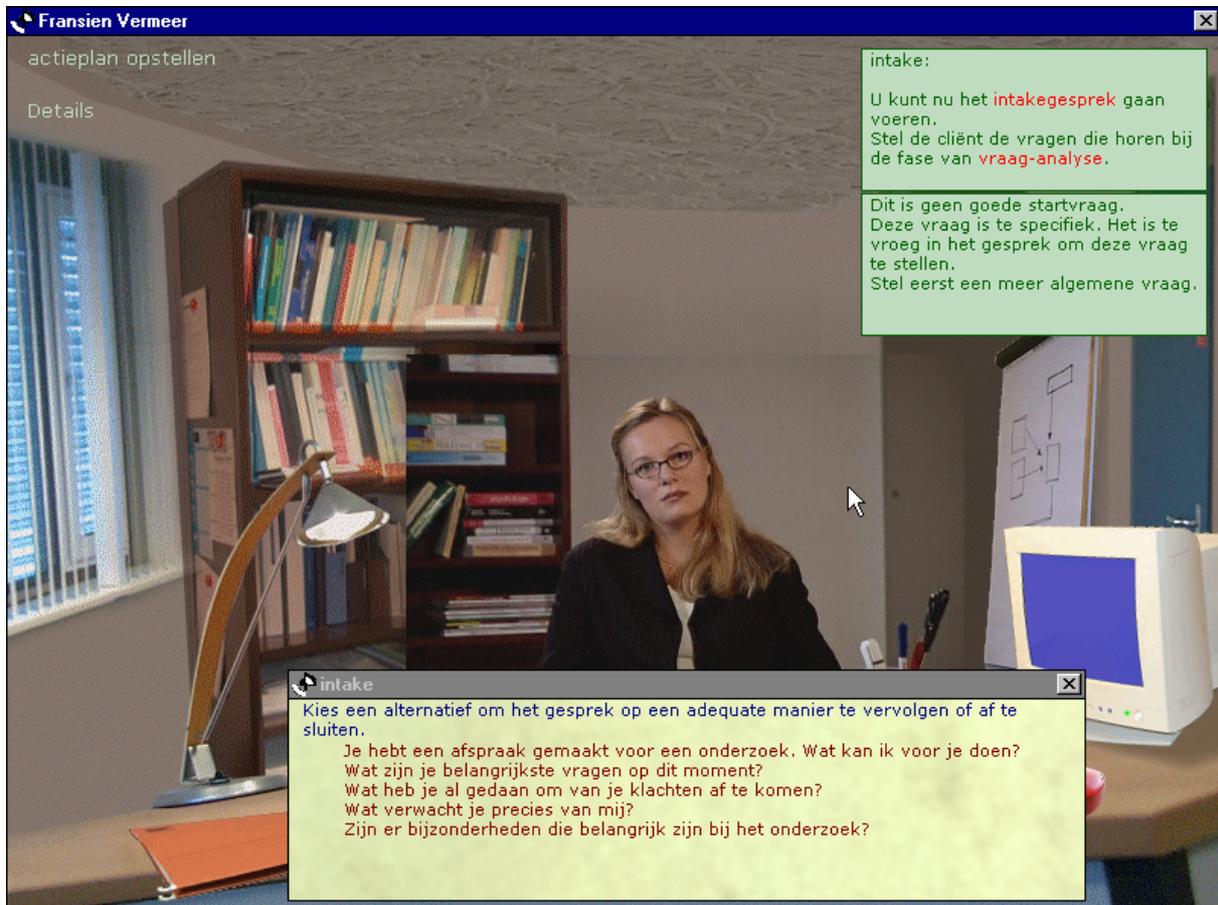


Figure 5. Snapshot of the user interface of DIAGNOST.

In accordance with the object-oriented programming paradigm all human computer interaction is materialized through concrete, visible objects: a portfolio gives access to the client's records, a pencil box is used to make notes or draw conclusions, the chair is used to invite the client to an interview, the white board is used to set up the diagnostic investigation etc. At this level of user-computer interaction, the program uses a hovering mechanism: moving the mouse pointer over one of the presented objects (chair, file, the white board etcetera) starts a modest animation (i.e. the chair wobbles slightly, the file opens itself etc.). Subsequent clicking with the mouse presents a menu with activities to choose from. Due to the intuitive character of the user interface, virtually no pre-operational instructions are needed, while still facilitating substantial student control. In figure 5, the student has just chosen to carry out the intake interview. Note that the video player (displaying the client) has been fully integrated with the background graphics. At the top right of the screen the assignment associated with the interview is given: a strategic comment, stating the operationalised learning objectives, which is in accordance with the notions of problem-based learning (Brown, Collins & Digid, 1989; Duffy & Jonassen, 1992). The student has just selected one of the questions from the list. This will trigger the video to play the relevant scene. Below the assignment a specific (tactic) comment on this choice has popped up. It is interesting to mention that while

students are deciding how to continue the interview, the program displays a random sequence of client images waiting for a response.

Program development and instrumentation

To develop this program, a number of specialists have been involved to cover case development and authoring, diagnostic methodology, educational design, video scripting and production, graphic design and programming: indeed, multimedia development demands multidisciplinary co-operation.

The program has been designed case-independent; that is, code and data are strictly separated. This allows extension with other cases or even extension to other disciplines, provided that sufficient authentic data are available. Indeed, additional cases have been developed on the domain of remedial education on behalf of teacher training in the Netherlands. In addition, a variation of DIAGNOST has been developed in co-operation with some universities in Flanders (the Casekit project). After completion of the functional program design, program development and content development were organised as two parallel processes, each of which had their own production cycles. To arrange the frequent updating and distribution of program code and data amongst the team members of the diverse institutes, we used an FTP-based client-server application. Each of the team members was given partial authoring rights for editing and uploading their contribution to the program. Apart from the program kernel, some 20 separate data entry tools have been developed for each of the student activities. For each option feedback texts can be entered, if necessary provided with production rules that will only fire when the specified conditions are met. These production rules are Boolean conditions that use the status of previous activities (completed or not) and choices made within any activities (selected or not selected). All program components have been developed in Asymetrix's ToolBook II Instructor 5.01. To warrant the integrity of the various program components and case data, part of the program testing has been automated.

Conclusions

During program development, extensive testing and formative evaluation have been carried out as part of the quality assurance cycle of the Open University of the Netherlands. By now, the program has been utilised by some hundred students as part of the academic degree program on psychology of the Open University of the Netherlands. To evaluate the program's practicability we have carried out additional interviews and questionnaires amongst the first cohorts of students (43 students). In particular, we focussed our survey on testing our starting points with respect to pedagogy, its implementation in the program and the appreciation by the students. Below, we will briefly discuss preliminary findings with respect to our claims.

- An intermediate role between theory and working practice
Students massively endorse the importance of the program in the preparation to real practical work. Its intermediate role is highly valued, in particular the authenticity and credibility of the three cases, the quality and depth of the contents, the possibility to make and restore mistakes and the detailed built-in support. After having completed the three cases, students report an increased self-confidence, increased diagnostic knowledge and skills and an increased efficacy of practical work.
- Constructivist pedagogy
Students consider the authentic cases captivating, highly motivating and adequate with respect to the learning objectives. In addition, students judge the program to be quite appropriate as a self-instructive educational means. They also indicate that the program strongly stimulates an explorative attitude; it strongly supports the students' autonomy and self-reliance. While they often seem to adopt a trial and error strategy to find out the best way to proceed, they frequently consult the built-in comments to reconsider their decisions. Occasionally, the intrinsic logical order of the diagnostic method seems to conflict with the idea of exploration. Yet, the development team

seems to have found the right balance between a fixed protocol and complete openness.

- Support mechanisms

Students make extensive use of both tactic comments and strategic comments. The quality and adequacy of the comments are highly commended. While students are allowed to overrule the program and set the feedback level by themselves (this is in accordance with the idea of self-reliant, independent learners), the majority of the students report to use the extended feedback comments (level 4). This pattern may easily reduce the students' alertness and stimulate a kind of "laziness". We consider to extend the program with an (intelligent) assessment algorithm that weighs the amount of support that a student has "consumed", combined with the speed of working, the number of faulty decisions etcetera, to arrive at an overall opinion on the student's performance. When applicable, intermediate (strategic) comments might urge the students to reduce the amount of retrieved feedback. In addition to the tactic and strategic comments, the students report to use the glossary and associated hotlinks quite frequently. Occasionally, background materials and operating instructions are consulted.

From formative evaluation it was concluded that, occasionally, additional support is necessary: if the program perceives that insufficient convergence occurs (i.e. the student is intensely lost) a strict intervention is generated and displayed to guide the student to completion of the activity. Such intervention would only fire in case of serious problems.

- The user interface

Despite the complexity of the diagnostic method and the extended features of the program, students seem to be able to work through the cases very well without noticeable prior operating instructions. The consultation room metaphor implemented with animated "hovers" on functional objects seems to be straightforward. The graphic design, featuring fully integrated video images of the clients, has an immersive effect and strongly supports the suggestion of real practice. Regular strategic comments provide sufficient clarity about objectives and activities. An important constraint of the program is the absence of free text inputs: students have to compose their texts (conclusions) by regularly selecting relevant sentences from lists. Some students report a preference to formulate their own hypotheses. However, students seem to be prepared to put up with this restriction, because it enables customized feedbacks. In addition, students indicate that the selection of correct statements from an extended list (up to some dozens of statements) is far from straightforward, especially when faulty ones are only subtle deviations from correct ones. Furthermore, students always are entitled to write down their own conclusions; in fact they appear to do so by editing the generated report.

- Other comments

The far majority of the students completed the cases successfully, though about 60 % met with some technical problems. A small number of students indicated not to be able to complete the cases due to technical problems: installation problems, program crashes, performance problems, sound problems or problems with status recovery. Most of the problems could be traced back to incorrect data or program code, which could be easily fixed. Some bugs appeared to be dependent on the user's hardware and software configurations. It should be noted that students mostly used their home computer, while often not knowing their system's specifications or the system requirements. A technical help desk set up by the course team proved to be of great value. Some concern was expressed about RSI. The course team has decided to add directions to prevent RSI.

In conclusion, we claim that the development team has succeeded in finding the right balance between the pursued student's freedom and autonomy on the one hand, and the

need for an effective guidance and support mechanism on the other. In particular, the built-in intelligent commenting routines allow the students to explore the domain and gradually improve their performances without being discouraged or getting stuck.

References

- Barrows, H.N. & Tamblyn, R.M. (1980). *Problem-Based Learning, an Approach to Medical Education*, Berlin/Heidelberg: Springer Verlag.
- Brown, J.S., Collins, A. & Duguid, P. (1989). Situated Cognition and the Culture of Learning, *Educational Researcher*, 18, 32-41.
- Bruyn, E.E.J. de, Pameijer, N.K., Ruijssenaars, A.J.J.M. & Aarle, E.J.M. van (1995). *Diagnostische besluitvorming: Handleiding bij het doorlopen van de diagnostische cyclus*, Leuven/Amersfoort: Acco.
- Duffy, T.M. & Jonassen, D.H. (1992). *Constructivism and the Technology of Instruction: A Conversation*, Hillsdale NJ: Lawrence Erlbaum Associates.
- Flavell, J. H. (1973). Metacognitive Aspects of Problem Solving. In L.B. Resnick (ed.), *The Nature of Intelligence*, Hillsdale NJ: Lawrence Erlbaum Associates.
- Kolb, D. A. (1982). *Experience, Learning, Development: the Theory of Experiential Learning*, Englewood Cliffs NJ: Prentice Hall.
- Nathan, M.J. & Resnick, L.B. (1994). Less Can Be More: Unintelligent Tutoring Based on Psychological Theories and Experimentation. In S.Vosniadou, & E. de Corte (eds.) *Technology-Based Learning Environments* (193-199), Springer Verlag, Berlin/Heidelberg.
- Scardamalia, M., Bereiter, C., McLean, R., Swallow, J. & Woodruff, E. (1989). Computer-Supported Intentional Learning Environments, *Journal of Educational Computing Research* 5, 51-68.
- Westera W. (1998). Paradoxes in Open, Networked Learning Environments: Towards a Paradigm Shift, *Educational Technology*, **39**, 1, 17-24.
- Westera, W. & Niesink, R. M. (1999). Non-directive, Self-instructive Media in the Field of Behavioural Toxicology and Addiction, *Medical teacher*, **21**, 4, 405-408.