# Guided Exploration: an Inductive Minimalist Approach for Teaching Tool-related Concepts and Techniques

CHRISTIAN KÖPPE
HU University of Applied Sciences, Utrecht, the Netherlands
and
RICK RODIN
Pace University, New York, USA

In this paper we introduce Guided Exploration as an inductive teaching approach. It is based on Minimalism and makes use of the pattern format. Guided Exploration addresses a couple of problems when teaching tool-related concepts and techniques, like how to address different student learning styles and how to address the issue that most students do not read the provided material as expected before starting to work. It also puts the focus on the concepts to be learned.

We describe the design of a first course at our university where we applied Guided Exploration. The evaluation gives promising results and shows that at least some of the known problems are addressed by this approach.

Categories and Subject Descriptors: K.3.2 [**Computers and Education**]: Computer and Information Science Education —*Computer science education*

General Terms: Design, Education

Additional Key Words and Phrases: Educational Patterns, Learning Styles, Inductive Teaching

**ACM Reference Format:**

Christian Köppe and Rick Rodin. 2013. Guided Exploration: an Inductive Minimalist Approach for Teaching Tool-related Concepts and Techniques. CSERC '13 Proceedings of Third Computer Science Education Research Conference, Arnhem, the Netherlands, 9 pages.

## 1. INTRODUCTION

One focus of Computer Science Education (CSE) is the teaching of tools together with the concepts and techniques related to these tools as described in the *Software Engineering Tools and Methods* knowledge area of the Software Engineering Body of Knowledge (SWEBOK) [IEEE Computer Society 2004]. This is also explicitly included in the *Computer Science Curriculum 2008* as expected capabilities and skills for computer science students: "Methods and tools. Deploy appropriate theory, practices, and tools for the specification, design, implementation, and maintenance as well as the evaluation of computer-based systems." [Interim Review Task Force 2008].

How these tools can be integrated in the curriculum has been discussed before, e.g. by Boloix and Robillard [Boloix and Robillard 1998]. They suggest a theory-tool course as part of a two-courses-in-sequence approach (theory-tool and project). The learnability of a tool is thereby an important aspect of the choice for tools. Boloix and Robillard rely on the tutorials and documentation provided by the tool vendors. This approach; making use of the available documentation, can also be observed at other institutions, including the university of the first author. What we often observe is that students do not at all or only marginally use the provided material and regularly experience such a course as "learning the tool" only instead of learning the concepts *and* the tool. They mostly struggle with technical, tool-related problems, whereby a thorough understanding of the techniques and concepts would help them with solving these technical problems faster.

We believe that one of the reasons of why the material is not used widely by the students and why they often fail to learn the concepts behind the tool is the way the provided material has been designed. Conventional manuals and training guides, known as "The Systems Approach" to instruction, address themselves to the logical and hierarchical decomposition of overall instructional objectives and the decomposition of these into enabling objectives. Each objective is incorporated into a systemic lesson structure incorporating specific "events" of instruction, such as gaining the student's attention, informing the student of the objective, eliciting performance, enhancing retention, training transfer, etc.

The factors that would motivate a student to work through such well-sequenced but pragmatically fragmented lessons are never discussed and are simply assumed to be present. Students are directed to practice and perform under the assumption that they will be happy to demonstrate their ability. Human problem solving is neither confined to nor always well served by typical manuals organized around a hierarchical system designed to control the direction of the training. Going from high-level objectives down to specific exercises leads to fragmented sessions. Training activities in such manuals are often carefully chosen, but not meaningful to the users. People tend to lose their motivation to work through such material and taking this orientation seriously raises deep questions about the adequacy of the Systems Approach [Carroll 1990].

The work of John Carroll [Carroll 1990] includes research giving strong support to the idea that people do not read big training manuals, and the more comprehensive such manuals are the less useful they become. This applies even to people who express a sincere commitment to reading the manual before they start; his research

Authors' addresses: christian.koppe@hu.nl, rick.rodin@verizon.net

*Conference CSERC '13*, 4-5 April 2013, Arnhem, The Netherlands

demonstrates that even such people do not read the manuals. Therefore it is not a good idea to give someone a task and say you must read the entire manual before starting.

Fuhrman et al. [Fuhrman et al. 2012] addressed this tool learnability challenge by providing only well targeted pointers to specific documentation artifacts in the lab assignment descriptions. Furthermore, the lab assignments were targeted tasks and the students learned the tool through performing these tasks and not only by completing some provided tutorials. As further support they supplied the students with small running examples, highlighting a specific aspect of a technology. Even though this is an improvement, some problems still remain. Students e.g. often do not follow the prescribed order of tasks. Another observation shared by many educators is that if some tasks are graded and others aren't, then students start with the graded ones, even if the experience of the other tasks would be helpful for the finishing of the graded ones. The same observation can be made for reading tasks. This is not a problem in itself, but indicates that the task design does not match the way students often follow.

For course development often one approach is chosen, but choosing one approach makes it hard to take the variety of learning styles into account. Students have different learning strategies, like e.g. overview first vs. examples and details first [Sun and Zhang 2004]. Entwistle, who summarizes important aspects of the work from Pask in [Entwistle 2001], describes these two as *holist* and *serialist* learners respectively. Holists prefer personal organisation and a broad view, while serialists prefer step-by-step and tightly structured learning [Entwistle 2001]. Offering only one way of learning in a course has potentially a negative effect on the learning outcomes of student population parts.

We believe that some of these problems can be addressed by combining different well-known approaches: inductive teaching, minimalism, and educational patterns. In this paper we will describe Guided Exploration as such a combination, discuss some basic features of it, and share our experience of applying it to the design of a course on the rapid application development (RAD) tool Oracle APEX[1] and its underlying concepts.

In the next section we will provide some background information on inductive methods, the minimalist approach, and patterns for education. We then describe how the combination of these leads to Guided Exploration as teaching approach. This approach was used for setting up a part of a course at our university, the design of this course is explained next. The results of the course evaluation are presented in the following section, and the paper concludes with a summary and suggestions for future work.

## 2. BACKGROUND

### 2.1 Inductive Teaching and Learning

The traditional approach to engineering instructions is deductive. First the theory is presented, followed by the application of these theories. This approach has also been applied for teaching tools and concepts, e.g. in [Boloix and Robillard 1998]. An alternative approach is inductive teaching and learning, and research supports its effectiveness as reported by Prince and Felder [Prince and Felder 2006]. They describe inductive teaching and learning as follows:

> "Instead of beginning with general principles and eventually getting to applications, the instruction begins with specifics — a set of observations or experimental data to interpret, a case study to analyze, or a complex real-world problem to solve. As the students attempt to analyze the data or scenario or solve the problem, they generate a need for facts, rules, procedures, and guiding principles, at which point they are either presented with the needed information or helped to discover it for themselves." [Prince and Felder 2006]

Different inductive teaching methods like inquiry learning, problem-based learning, project-based learning, or just-in-time teaching all share these common features:

—Questions or problems provide context for learning.

—Students discover course content for themselves.

—Learner-centered.

—Active learning.

### 2.2 Minimalist Training Design

Minimalism is a style of writing training materials; materials that allow learners to start immediately on meaningfully realistic tasks, reduce the amount of reading and other passive activity in training, and help make errors and error recovery less traumatic and more pedagogically productive [Carroll 1990]. Minimalism emerged in the early 1990s, around the same time as Patterns and Pattern Languages became more known in software design, and research supports Minimalism's effectiveness [Carroll 1990].

Carroll [1990] presents three fundamental aspects of the minimalist instructional approach, along with the concept of "Exploration as Instruction". The three aspects of minimalism are:

(1) Allowing learners to start immediately on meaningfully realistic tasks.

(2) Reducing the amount of reading and other passive activity in training.

(3) Helping to make errors and error recovery less traumatic and more pedagogically productive.

This list can be shortened to the following three principles of minimalism: "More to Do", "Less to Read", and "Help with Errors". According to Carroll, "when one is designing instructional materials what is important is the key idea of minimizing the extent to which instructional materials obstruct learning. The starting point for such design is an eclectic synthesis of available design elements marshaled to address instructional objectives and usability objectives. There is no deductive theory, that is, given a set of minimalist principles we cannot just crank out a training manual" [Carroll 1990]. Such material "requires a high degree of modularity, a structure of small, self-contained units" [Carroll 1990].

### 2.3 (Design) Patterns

Patterns (and pattern languages) were first introduced by the architect Christopher Alexander and his colleagues in *A Pattern Language: Towns, Buildings, Construction*. According to them, "Each pattern describes a problem that occurs over and over again in our environment, and then describes the core solution to that problem, in such a way that you can use the solution a million times over, without ever doing it the same way twice." [Alexander et al. 1977].

Patterns are "eclectic design elements" [Carroll 1990] addressing the principles of "Less to Read" and "More to Do". While conventional Pattern forms vary, one common characteristic is that they are brief, often a single page or less. Also, Patterns support skimming and reading in any order, or at least in a very flexible order. Manns

[1] http://apex.oracle.com/i/index.html

and Rising likewise refer to this aspect of patterns when discussing appropriate pattern naming: “You will find that after quickly reading the pattern summaries that you will understand what the pattern reference means, even if you haven’t read all the information” [Manns and Rising 2005], and “We’re not prescribing a fixed sequence for using our collection of patterns...” [Manns and Rising 2005]. Skimming short patterns, or even just their names or descriptions, in any order combines “more to do” with “less to read”, allowing one to start sooner while reducing the amount of reading one must do. The entire Fearless Change pattern language is oriented around providing readers with a “small package of patterns” [Manns and Rising 2005] for each stage of one’s “journey” into organizational change. In these ways Patterns accomplish two of the three basic principles of minimalism. The whole point being, as with Minimalist Documentation, with Patterns you don’t have to read an entire dense manual before starting, one simply starts right away with what they specifically need with which to succeed.

Patterns provide a Resulting Context supporting the “Help with Errors” dimension of minimalism. Manns & Rising relate the following situation “A fellow from one company said, ’I think that anyone who uses this solution should be aware that there are some problems with it. Should that go in the Solution?’ ’Actually,’ Linda replied, ’there is an additional section for concerns like that, it’s called Resulting Context’” [Manns and Rising 2005]. Patterns contain a wealth of information supporting their solution and providing help where there are “problems with it.” In this manner they help make error recovery easier, less traumatic, and more informative.

Pedagogical patterns describe proven solutions for designing different aspects of education. One of the first publications in this field was *Patterns for Classroom Education* [Anthony 1996], which already addressed questions like how to address the needs of all kinds of students. The Pedagogical Patterns Project [ppp 2012] introduced the pattern approach to a broader audience and collected patterns in the areas — among others — of active learning, feedback, and experiential learning [Pedagogical Patterns Editorial Board 2012]. Recent activities in the field of educational patterns focus on teaching specific content, like patterns for teaching design patterns [Köppe 2013], or applying pattern mining for including professional skills in a software engineering curriculum [Köppe 2012].

Rick Rodin [Rodin 2012] introduced a pattern collection mined from the principles behind Carroll’s approach to minimalism so that these principles can be used for developing minimalist training materials based on the pattern format. The target audiences of these patterns are organizations having a need to train individuals on new or rapidly changing tasks, helping them to acquire the skills they need as quickly as possible. Using the eight “principles” of minimalism captured in these Patterns, one can “define” minimalist instruction. These Patterns serve as guides to a process that will create the minimalist training objects.

A “Training Object” is a small, self-contained, and self-describing digital resource used for training that has a specific purpose [Gupta et al. 2010]. In Minimalism, the “Guided Exploration Card” is the Training Object. The results of the patterns for developing minimalist guided exploration objects are P-Forms, which will be the Training Object and will support Guided Exploration, but will be different from traditional Guided Exploration Cards.

A P-Form is a piece of documentation structured like a Pattern or Proto-Pattern but containing new or emerging knowledge, not existing established knowledge. The name P-Form is an abbreviation of “pattern form”, as they make use of the pattern format but are no patterns. Like Patterns, P-Forms follow Minimalism and are a form of Minimalist documentation. The basic elements of a P-Form and their intention are described in following list:

—**Name** - a unique title that covers the resulting thing and the process which leads to it.

—**Context** - defines the applicability of the P-Form and scopes the environment.

—**Problem** - states the essence of the addressed problem.

—**Forces** - define the problem in more detail.

—**Solution** - the common aspect of implementations which solve the problem in the described context and balance or resolve the forces.

—**Resulting Context** - describes the context which emerges after the application of the solution, including the benefits and liabilities.

—**What if I get stuck** - hints on how to solve common errors and how to recover if an error has been made. To improve “error handling and recovery”, this section is separated out from the resulting context, where it usually can be found in patterns.

The suggested benefits of using P-Forms over Guided Exploration Cards are based on the theory that P-Forms can harness the power of Patterns and Pattern Languages to convey rich information, and thus represent an improved mode of knowledge sharing and management. As Alexander says in his *Timeless Way of Building* [Alexander 1979], the patterns are there whether we verbalize them or not. Software designers have known for a long time that by using Patterns we can come to a more conscious realization of what it is we are doing, helping us to communicate better — more precisely — and ultimately contributing to a greater implicit understanding of good design. Therefore the finished form of the Minimal Manual is preferred to be a set of P-Forms instead of Guided Exploration cards.

## 3. GUIDED EXPLORATION AS INDUCTIVE TEACHING APPROACH

P-Forms in combination with Guided Exploration and other principles of Minimalism can be used in the design of training materials developed to teach students of various learning styles the concepts, theory, practice and tools of Software Engineering. In the subject area of Software Engineering and Computing the impact of cheaper, faster information processing and communications has flattened decision making structures, making them more distributed, and accelerates the need for much more rapid knowledge and skill acquisition, and with this the demand for skilled workers [Brynjolfsson and McAfee 2012].

Carroll’s work, which is based on extensive research, demonstrates that it is better to create the smallest possible instructional guide that enables self-directed training taking advantage of what people already know, letting them get started quickly (without having to read a lot first), and letting their mistakes be teaching opportunities.

Guided Exploration is in its nature inductive, it shows the same important features as other inductive methods. The implementation of GE is mainly comprised of a set of P-Forms. These P-Forms contain all concepts to be learned and applied by the students. They follow hereby, as described in the previous section, a specific format. The motivation for each P-Form is formulated as a problem that occurs in a specific context. Besides that, the problems described in P-Forms are mostly of a lower scope. This is similar to problem-based learning and one of the general features of an inductive method. The forces section helps with understanding the problem in more detail.

The solution tells on a conceptual level what needs to be done to solve the problem and what the resulting context will be after implementing the solution. Not telling the students specifically what to do, as most tutorials do, but by describing the solution on a higher level and in combination with the problem and the context, the student is exposed to the underlying concepts of the tool to be learned, not only the techniques to follow. This is significantly different from most approaches for integrating tools in a curriculum, and in our opinion one of the valuable features of Guided Exploration.

Some P-Forms include links to other P-Forms. This shows the students the relations between concepts and their interdependencies. It supports the students to recognize these dependencies and to take them into account in the order of P-Form application, even though the students are encouraged to start with whatever P-Form they want.

GE forces the students to actively work on the subject The passive surface learning approach that can be found with some students does not work. As the students do not get an extensive introduction to the tool, but have to immediately start working with it, they likely will run into some problems and hereby discover the need for more information. This information can be provided as part of the "What if I get stuck" section, but also in optional lectures using just-in-time teaching. These lectures are hereby triggered by the student needs and on demand only.

## 4. COURSE DESIGN BASED ON GUIDED EXPLORATION

We decided to use an existing course we already had experience with and which showed some of the problems mentioned earlier. The course design was not completely changed, as there was no experience with this new approach and its applicability in an educational context. We wanted to guarantee that, in case of an obvious failure of the new approach we had a back-up. We therefore included all important aspects of GE, but also kept a few of the original parts of the course like the lecture slides, the functional and technical description of the assignment, and the final grading criteria. In this section we describe the mixed design and the decisions that led to it.

The course we used for applying Guided Exploration was Model Driven Development Tooling. It is part of the second year of an undergraduate Computer Science program with a specialization in Software Engineering. One part of this course focused on rapid application development of administration systems. The learning objectives of this part were defined as follows:

—The student has knowledge of the basic building blocks of administration systems.

—The student is able to realize a medium complex system with the 4GL tool Oracle APEX.

—The student is able to implement business rules in APEX with PL/SQL.

The end terms of the re-designed course stayed the same as with the previous version. The students were free to choose their own way towards these end terms, meaning that they were allowed and encouraged to apply the provided P-Forms in the order they found the most important, interesting, natural, or easy. However, a standard route was provided too for the students who have problems with such less-structured assignments. These possibilities were also told to the students. We expected that this design would suit both the holist and the serialist learners.

The students were also free to choose between a system contrived by themselves or a standard system provided by the teacher in the form of functional and technical requirements. In the first case the students had to describe the required functionality themselves, taking the required concepts and techniques into account. As the same P-Forms are used for both kinds of systems, they implicitly also show that the concepts and techniques they contain are not specific for one system, but for a style of systems (in our case administration systems).

The tool used in this course was Oracle APEX. This tool was either not mentioned at all or only mentioned in one place in the P-Forms as part of the realization of the minimalist principle of "error handling and recovery", namely the "What if I get stuck/Hints" section. In this way it was possible to put more focus on the real learning objectives - the concepts and techniques - as these are stated in the P-Forms (and not the APEX specific How-To's).

An exemplary P-Form is shown in Figure 1. Please note that we added another section called "Fulfillment criteria", which contains the information about the requirements needed for successfully completing this P-Form. These criteria were later used for grading, so the students did know how they will be graded.

The P-Forms were mapped to the functional requirements coming from the earlier version of the course. An exempt from this mapping is shown in this list:

—Use group defined navigation (2-level tab) - RESPECT USER ROLES, GOOD NAMING, NAVIGATION SUPPORT

—User friendly maintenance of all application data tables - HELP WITH VALUE SELECTION, VALIDATE YOUR DATA, USE DEFAULT VALUES, SELECTION REPORT, ADMINISTRATE COMPLEX DATA, ADMINISTRATE SIMPLE DATA, GOOD NAMING, NAVIGATION SUPPORT

—...

The information about the P-Forms was intentionally provided in different ways:

(1) A list was given with an example order of applications of P-Forms. This order was determined by one possible logical flow of P-Form applications.

(2) A list was given of all required P-Forms in descending alphabetical order of the titles (with the only two not-graded P-Forms "Use Sample Application" and "Functionality Description" preceding this list). The same order was also applied for publishing the P-Forms on the Learning Management System (LMS).

The problem that students tend to focus mainly on the tasks which are graded is narrowed through the list of required P-Forms, meaning that starting with the first task, the students are working on parts of the assignment which will be graded.

The P-Forms are presented in a short and concise way, not extending the size of one page. This follows the minimalist principle of "less to read". However, all P-Forms contain links to the standard documentation and therefore offer the possibility for further reading.

We decided to combine this approach with "teaching on demand". After a short introduction to the topic the students were encouraged to start working on the assignment. Any questions that arose had to be posted on a forum provided in the LMS or had to be asked to the teacher during the practical work. The content of

RESPECT USER ROLES

**Context**: Your application has different types of users, and some parts of the application are only allowed to be used by specific users.

**Problem**: Providing all functionality to all users gives some of the users the possibility to work with parts of the application they're not allowed to work with.

**Forces**:

—Different stakeholders are often interested in different data. An employee who's responsible for entering data might not be interested in monthly reports on sales, and a manager might not be interested in the amounts of items of a specific order, but in the total sales numbers per month.

—Something with user rights/who's working with the application

**Solution**: Identify all different user roles and their interests. Respect the user roles by adapting the application so that only the parts of the application are shown or available which are relevant for the current user or he/she is allowed to use.
The implementation is dependent of the specific user roles and their interests and rights. A data entry employee should not have the opportunity to add new employees or to run management reports. Standard ways of implementing this are:

—Deactivating or removing menu items,

—A check when the user tries to go to a page if he or she is allowed to go to that page,

—Deactivating or removing page items, like e.g. text-fields or selection lists,

—Setting a default value of an item based on the user role,

—Changing properties of an item, like e.g. from optional to required.

**Resulting Context**: The user only sees the parts of the application he or she is interested in. If the user is not allowed to use certain application parts, then these are not available or reachable for the user.

**What if I get stuck/Hints**:

—Remember: Authentication and Authorization (Authentication has to do with granting access only for known-users. Authorization is granting access to parts of the application.)

—For documentation see UserGuide, chapter 13, paragraph "Establishing User Identity Through Authentication" and "Providing Security Through Authorization".

—Authentication is defined and used on application level. Authorization is defined on application, page, region and/or item level.

**Fulfillment Criteria**:

—No application parts are available — or reachable — for a user which do not match the interests of the user or do fit the rights the user has.

Fig. 1: Example P-Form: RESPECT USER ROLES

the following short lessons was determined by the questions asked, and therefore provided the information needed by the students just in time. The teachers role was mainly that of a supervisor or tutor, putting the guidance into the students' hands.

We included one P-Form called USE SAMPLE APPLICATION. It contained the following problem and solution:

> **Problem**: You don't know what kind of applications, in terms of functional requirements can be built with APEX.
> **Solution**: Use the provided example application for exploring the possibilities of APEX and recognizing the possible functionality. Walk through all menu items and try to use all parts of the application.

The intention of this P-Form was two-fold: we wanted to offer students preferring the holist learning strategy a good way to get an overview of the possibilities of APEX and to show them what, more or less, was expected from them. On the other hand we wanted to make the students aware of the possibility of exploring an existing application and getting information on how this was implemented which they could then use for their own assignment. This P-Form was optional and not graded.

## 5. EVALUATION

In total there were 45 students enrolled in this course. The first part of the evaluation was formed by the observations of the teachers. Furthermore, we collected data by asking the students to weekly answer the following questions:

| Team | Order of P-Forms | Answers to “Why did you apply the P-Forms in this order?” |
|---|---|---|
| T1 | 1,2,4,19,9,17,14 | “We found this order quite intuitive.” |
| T2 | 1,2,3,4,5,7,19,8,11,9,14,17,13,12,10,15 | “We applied the P-Forms in this order because we needed it for the functionalities we made.” |
| T3 | 2,3,4,14,5 | “We applied them in this order because it was a quite natural order.” |
| T4 | 3,2 | “We did the P-Form Functionality Description first, because we felt it was necessary to get that one done first. You cannot start a project without knowing the requirements, because if you do, you do not know what you are building and you will deliver an unusable application.” |
| T5 | 2,4,19,14,5 | “First we must have the database, otherwise we could not work with APEX.” ... “So that there will be a structure in the app” |

Table I. : Reasons for different order of P-Form application

—Which P-Forms did you implement during the last week and in which order (please be as specific as possible)?
—Why did you apply the P-Forms in this order?
—Which problems did you experience regarding these P-Forms and how did you solve them?
—Which P-Forms (or parts of them) were no problem for you and easy to realize?
—What other resources than the provided ones did you use?

At the end of the course the students had to fill in a questionnaire. As this was directly given and collected during the first meeting of the course preceding the described one, where most of the students were present, the number of responses was relatively high (n=42).

## 5.1 Teacher Observations

35 students finished the course in the first instance. 23 students chose the given auction system, 12 students chose their own system and therefore also applied the P-Form FUNCTIONALITY DESCRIPTION. As there was only a final deadline for the assignment and no obligatory deliverable during the assignment period, some students did not start immediately to work on the P-Forms, but postponed it to a later moment.

The final grades for this course part were an average of 8.1 (on scale from 1 to 10, where 10 is the highest possible grade), the solutions were of a high quality and better than the results from the previous years' course (average of 7.4). One reason might be that the students knew in this course precisely the requirements for grading, as these were explicitly included in the P-Forms. A few students also mentioned in the feedback that they used the P-Forms mainly for checking if they fulfilled all requirements.

The provided forum on the LMS was not extensively used, there were in total 7 questions. But more questions were asked during the practical work and also actively inventoried by the teacher. These questions were used for determining the content of the next lecture as part of the “teaching on demand” and providing the information just in time. An interesting observation here was that this demand-oriented order of presented content was different from the chosen order in the earlier edition of the course, even though the same content was presented. This indicates that indeed the order of content in the earlier version, which was basically determined using the system approach, did not fit the information needs of the students well. As the material for the on-demand lectures was reused from the earlier course version, the adaptation of the Guided Exploration in the new course design was no problem — no new material had to be developed for the lectures.

The P-Forms helped the students to find an entry-point to the assignment. Whatever they chose was also taken into account for grading, and therefore of value for the students. In the beginning we observed that some students had difficulties relating the P-Forms to the required functionality of the system to be built. These difficulties might have been caused by this approach being different from what they were used to in all earlier courses of their study. After starting to work with the P-Forms, these difficulties diminished and the students experienced this way of working mostly as easy, which is supported by the responses in the questionnaire.

## 5.2 Results from Weekly Questions

In total 12 student teams (total 22 students, they had to work in pairs, but some worked alone) responded to the questions. The number of responses decreased over time, which means that from most teams there was not a complete order of P-Form applications known, but mostly only the first ones. However, some interesting observations can be made based on the acquired data.

3 of the 12 teams followed the provided default order, 7 teams followed a different order, 1 team did not make use of P-Forms for the implementation, and 1 team only provided information over the first applied P-Form. All teams started with the first P-Forms as provided in the document, but then they went on with a diverging order of P-Forms. 5 teams that used a (partly) different order explicitly stated reasons for their choices, these are summarized in Table I.

The quotes in combination with the different orders of P-Forms suggest that some of the students were consciously thinking about what a good and logical next step would be (instead of just aimlessly going on) and that these logical steps differ per student (team). This suggests that the degree of freedom in terms of order of P-Form application does not go against what the students see as natural or logical, and therefore supports them in the “flow” of working through the assignment. The different order of P-Form applications and the comparable good results indicate that this part of Guided Exploration — “reading in any order” — can be successfully applied. It gives the students a high degree of freedom in terms of how they reach the final goal, and a certain population of students makes use of this freedom. This is also supported by the answers to the questionnaire.

Many students mentioned that the P-Forms generally were easy to implement. When they encountered problems, they asked the teacher so that topics were taught in the next lecture. This shows that the chosen “teaching on demand” approach — or just-in-time teaching — was well perceived by the students. One student responded that the provided additional material (the Oracle APEX documentation) was too complex and that the internet and the teacher were used instead for getting the required information.

| | $\bar{x}$ | M | $\sigma$ |
|---|---|---|---|
| Q1: I liked using the P-Forms. | 3.21 | 3.00 | 1.10 |
| Q2: I would have preferred more classical lectures and exercises. | 2.95 | 3.00 | 1.23 |
| Q3: I understood the concepts of administration systems. | 3.98 | 4.00 | .56 |
| Q4: I would like to have more courses that use P-Forms. | 2.79 | 3.00 | 1.05 |
| Q5: I learned APEX very well. | 3.26 | 3.00 | .86 |
| Q6: I liked to have a well-structured order of tasks as with the default order of the P-Forms. | 3.55 | 4.00 | .97 |
| Q7: I liked that I was able to choose my own order of P-Form applications. | 3.60 | 4.00 | .99 |
| Q8: The P-Forms helped me to understand what I was doing. | 3.17 | 3.00 | 1.10 |
| Q9: My new knowledge will help me to implement better administration systems in the future. | 3.07 | 3.00 | 1.05 |
| Q10: I can use APEX easily for implementing other administration systems. | 3.07 | 3.00 | 1.07 |
| Q11: The P-Forms were easy to understand. | 3.71 | 4.00 | .97 |
| Q12: I missed a book or reader in this course. | 2.90 | 3.00 | 1.27 |

Table II. : Results questionnaire (n=42)

| age | n | $\bar{x}$ |
|---|---|---|
| <=20 | 20 | 2.80 |
| 21/22 | 14 | 3.43 |
| >=23 | 8 | 3.88 |

Table III. : "I liked using the P-Forms" - Mean per age

Two students responded that they did not use the P-Forms to guide the implementation. They took the functionality description of the system to be built and then used the P-Forms only for determining if their systems fulfilled the described assessment criteria.

The data show that from the 12 responding teams only 6 started with the P-Form USE SAMPLE APPLICATION. The other 6 teams started directly with implementing functionality. This might be an indication of different learning strategies — we expected that especially holist learners will make use of this P-Form — but the data do not further support this.

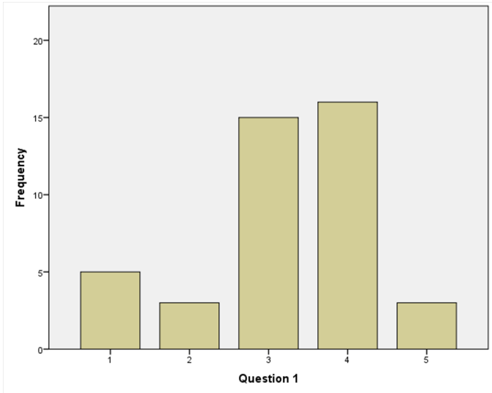


Fig. 2: Question 1: I liked using the P-Forms

## 5.3 Results from Questionnaire

Table II gives an overview of the questions and the results from the questionnaire. For the answers we used a Likert scale from 1 to 5, where 1 stands for strongly disagree and 5 for strongly agree. We also asked for the age, sex, and pre-education of the students. The three types of pre-education mentioned were HAVO (higher general continued education), MBO (middle-level applied education), and VWO (pre-university secondary education). Please note that we did not find a significant difference between male (n=40) and female (n=2) students, although the number of female students is too low to allow any generalizing of this finding. In the next paragraphs we will interpret the results per question.

**"I liked using the P-Forms".** Even though the mean is not very high, the standard deviation shows a high diversity of responses. Figure 2 clearly shows the majority of students either slightly liked the P-Forms or absolutely disliked it. One reason for this absolute disliking might be that this was the first time that the students were exposed to an inductive teaching approach. Research has shown that students learn more effectively when the presented material matches their preferred learning style [Entwistle 2001].

An interesting observation that was made is that the liking of the P-Form approach increases with age, as shown in Table III. This suggests that older students are more likely to respond positively to Guided Exploration than younger students.

**"I would have preferred more classical lectures and exercises".** The overall mean of 2.95 does not clearly show any preference. However, the high standard deviation indicates a broad variation in answers. Figure 3 shows that most of the students either agree or disagree. Further examination showed significant differences between the two classes, but not between pre-educations. We expect that this is related to the average age per class, as the class which agreed more on that statement (mean=3.29) had an average age of 20.4 years, while the class showing more disagreement (mean=2.5) had an average age of 21.7 years. This could be interpreted as that our chosen approach is more fitting to older students, which correlates with the findings from the 1st question. Further research might be needed here to determine an appropriate starting point in the curriculum, related to the age of the students, when incorporating Guided Exploration.

**"I understood the concepts of administration systems".** This question was directly related to one of the learning objectives of the course. The mean of 3.98, the small standard deviation of 0.56, and the fact that none of the students disagreed show that at least in the perception of the students this objective has been realized. This is also supported by the final grades for this assignment with an average grade of 8.1, compared to 7.4 last year (on a scale of 1-10 with 10 being the highest possible grade).

**"I would like to have more courses that use P-Forms".** The overall mean of 2.79 shows a slight disagreement. Even though most students liked the P-Forms, they did not want to have more courses using them. The only exception here were students with VWO pre-education with a mean of 3.57. This pre-education is

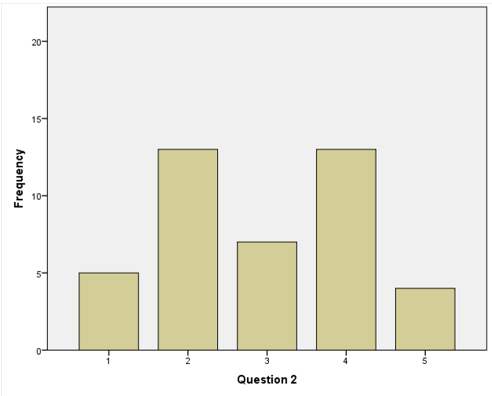


Fig. 3: Question 2: I would have preferred more classical lectures and exercises

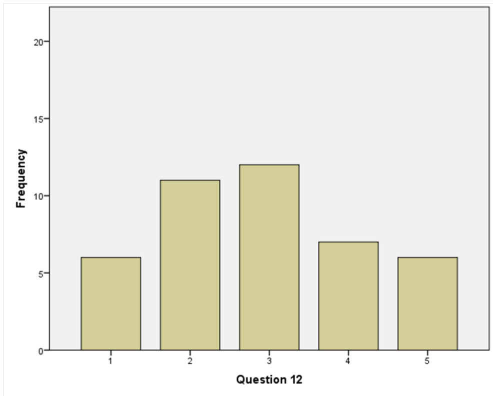


Fig. 4: Question 12: I missed a book or reader in this course

| age | n | $\bar{x}$ |
|---|---|---|
| ¡=20 | 20 | 2.85 |
| 21/22 | 14 | 3.29 |
| ¿=23 | 8 | 3.75 |

Table IV. : ”The P-Forms helped me to understand what I was doing” - Mean per age

more oriented towards an academic career, and the result could therefore be seen as an indication that Guided Exploration might be appropriate for academic education. Further research is a needed to support this assumption.

**”I learned APEX very well”.** This is directly related to one of the learning objectives. The mean of 3.26 and the absence of strong-disagree responses suggest that this objective was partially achieved. This is again supported by the quality of the final results and the corresponding grades.

**”I liked to have a well-structured order of tasks as with the default order of the P-Forms”.** and **”I liked that I was able to choose my own order of P-Form applications”.** Both questions have a relatively high mean of 3.55 and 3.6 respectively, which shows in our opinion that offering different ways of approaching the assignment was well received by the students. This is an important aspect of Guided Exploration (and minimalism), and could also be applied in other courses.

**”The P-Forms helped me to understand what I was doing”.** The overall result suggests a slight agreement, but bigger differences can be found when examining the answers per age and pre-education. Table IV shows the average answer per age. This shows that especially older students experienced the P-Forms as helpful for their understanding.

Students with an MBO pre-education agreed the most on this question (mean=3.56), followed by VWO (3.14) and HAVO (2.88). This result is unexpected, and we have no possible explanation for it. Further research is needed to more clearly evaluate this part of Guided Exploration.

**”My new knowledge will help me to implement better administration systems in the future”.** and **”I can use APEX easily for implementing other administration systems”.** The overall responses are quite similar for both questions, both having a mean of 3.07. However, further examination of data shows that students with MBO pre-education agreed more on the first question (mean=3.17) than on the second question (mean=2.83). For students with HAVO pre-education it is exactly the other way around (with means of 2.81 and 3.13 respectively). Students with VWO pre-education agreed stronger (means of 3.57 and 3.43). Again, it seems that pre-education can be of influence on the outcomes of the Guided Exploration approach. Also the age seems to play a role, as students with an age of 23 or higher tendentially agreed on both statements (means of 3.73 and 3.27).

**”The P-Forms were easy to understand”.** Most of the students agreed on that. We assume that one reason for this is the uniform structure of the P-Forms, which allows the students to easily identify the relevant parts. Even though some students initially had trouble with understanding the P-Forms, once they had grasped the concept behind them it was easy for the students to apply the other P-Forms too.

**”I missed a book or reader in this course”.** Survey results support that a course book or reader was not missed by a small but significant population of respondents, as shown in Figure 4. But the broad distribution of the values ($\sigma$=1.27) suggests that more material should be provided, but that this material should be additional and optional.

## 6. CONCLUSION

In this paper we presented Guided Exploration as an inductive approach to teaching tool-related concepts and techniques. We described how we used it to design a course on the development of administration systems using Oracle APEX. The evaluation of this course shows promising results.

In general Guided Exploration and the P-Forms as instructional material were tendentially well perceived by the students and the learning objectives were achieved in a slightly better way than with the old course design. However, the data also lead to some other conclusions and suggestions, and in consequence to further necessary research.

The success of parts of Guided Exploration seems to be dependent on the age of the students and their pre-education. This requires further research, including the question how Guided Exploration could be adapted to fit the learning styles and the level of all students best.

Guided Exploration as a pedagogical approach does not solve the well-known problem that students wait with working on the assignment until the deadline approaches. One possible solution is to make the implementation of 2-3 P-Forms per week required and, if necessary, part of the grading. This is also the implementation of the CONTINUOUS ACTIVITY pattern [Köppe 2011]. One important aspect is that it does not matter which of the P-Forms were implemented, as this follows the principle of "reading in any order".

The aspect of using mistakes as learning opportunities was not explicitly addressed and evaluated. This is part of future work.

## REFERENCES


2012. Pedagogical patterns project. (2012). http://www.pedagogicalpatterns.org/

Christopher Alexander. 1979. *The Timeless Way of Building* (later prin ed.). Oxford University Press, New York.

Christopher Alexander, Sara Ishikawa, and Murray Silverstein. 1977. *A Pattern Language: Towns, Buildings, Construction (Center for Environmental Structure Series)*. Oxford University Press.

Dana L. G. Anthony. 1996. Patterns for classroom education. In *Pattern Languages of Program Design 2*, John Vlissides, James Coplien, and Norman Kerth (Eds.). Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 391–406. http://dl.acm.org/citation.cfm?id=231958.232961

G. Boloix and P.N. Robillard. 1998. CASE tool learnability in a software engineering course. *IEEE Transactions on Education* 41, 3 (1998), 185–193. DOI:http://dx.doi.org/10.1109/13.704544

Erik Brynjolfsson and Andrew McAfee. 2012. *Race Against the Machine: How the Digital Revolution is Accelerating Innovation, Driving Productivity, and Irreversibly Transforming Employment and the Economy*. Digital Frontier Press. 98 pages. http://www.amazon.com/Race-Against-Machine-Accelerating-Productivity/dp/0984725113

John M. Carroll. 1990. *The Nurnberg Funnel: Designing Minimalist Instruction for Practical Computer Skill*. MIT Press, Cambridge, MA, USA. http://dl.acm.org/citation.cfm?id=80371

Noel Entwistle. 2001. Styles of learning and approaches to studying in higher education. *Kybernetes* 30, 5/6 (Jan. 2001), 593–603. DOI:http://dx.doi.org/10.1108/03684920110391823

Christopher Fuhrman, Roger Champagne, and Alain April. 2012. Integrating tools and frameworks in undergraduate software engineering curriculum. (June 2012), 1195–1204. http://dl.acm.org/citation.cfm?id=2337223.2337380

Saurabh Gupta, Robert P. Bostrom, and Mark Huber. 2010. End-user training methods: what we know, need to know. *ACM SIGMIS Database* 41, 4 (Nov. 2010), 9. DOI:http://dx.doi.org/10.1145/1899639.1899641

IEEE Computer Society. 2004. *Software Engineering Body of Knowledge (SWEBOK)*. IEEE Computer Society. http://www.swebok.org/

Interim Review Task Force. 2008. *Computer science curriculum 2008: An interim revision of cs 2001*. Technical Report. ACM/IEEE. http://scholar.google.nl/scholar?q="computer+science+curriculum+2008"&btnG=&hl=nl&as_sdt=0,5#2

Christian Köppe. 2011. Continuous Activity - A Pedagogical Pattern for Active Learning. In *Proceedings of the 16th European Conference on Pattern Languages of Programs - EuroPLoP '11*, Vol. 2011. ACM Press, Irsee, Germany. DOI:http://dx.doi.org/10.1145/2396716.2396719

Christian Köppe. 2012. Using pattern mining for competency-focused education. In *Proceedings of Second Computer Science Education Research Conference - CSERC '12*. ACM Press, Wroclaw, Poland, 23–26. DOI:http://dx.doi.org/10.1145/2421277.2421280

Christian Köppe. 2013. A Pattern Language for Teaching Design Patterns. In *Transactions on Pattern Languages of Programs III*, James Noble, Ralph Johnson, Uwe Zdun, and Eugene Wallingford (Eds.). Springer, Berlin, Heidelberg, 24–54.

Mary Lynn Manns and Linda Rising. 2005. *Fearless Change: Patterns for Introducing New Ideas*. Addison-Wesley. http://2www.transitionnetwork.org/sites/default/files/ABCchange2010.pdf

Pedagogical Patterns Editorial Board. 2012. *Pedagogical Patterns: Advice for Educators*. Joseph Bergin Software Tools. 230 pages.

Michael J Prince and Richard M Felder. 2006. Inductive teaching and learning methods: Definitions, comparisons, and research bases. *Journal of Engineering Education* 95, 2 (2006), 123–138. http://www.it.uu.se/edu/course/homepage/cosulearning/st11/reading/ITLM.pdf

Rick Rodin. 2012. Meta Patterns for Developing Minimalist Guided Exploration Objects. In *Proceedings of the 19th Pattern Languages of Programs conference, PLoP'12*. Tucson, AZ, USA.

Ron Sun and Xi Zhang. 2004. Top-down versus bottom-up learning in cognitive skill acquisition. *Cognitive Systems Research* 5, 1 (March 2004), 63–89. DOI:http://dx.doi.org/10.1016/j.cogsys.2003.07.001