

# Users' Learning of Principles of Computer Operations

*Elena Stamatova and Jens Kaasbøll*  
*University of Oslo, Oslo, Norway*

[elenasta@ifi.uio.no](mailto:elenasta@ifi.uio.no) [jensj@ifi.uio.no](mailto:jensj@ifi.uio.no)

## Abstract

In order to develop computer competence, users need to understand principles of its operation in addition to memorizing where a particular operation is located in menus. Understanding a principle ease transfer of learning, hence the understanding helps learning a new application. These principles, e.g. functional dependency, are based in computer science, and are thereby precise and general. Two cycles of the Action-Process-Model (Aharoni, 2000) of learning abstract concepts are used to characterize the learning process of such principles. Observations and tests of students in a high school show discrimination errors and forgetting already in the first of six stages of learning. Nevertheless, the majority of the students arrived at the fourth stage of learning functional dependency. In order to improve learning, teachers need to provide both intermediate and general concepts so that the learners can verbalize their actions and hence improve their understanding.

**Keywords:** learning, user training, transfer, abstraction, concepts, computer programs, applications.

## Introduction

Users need to know the technology in addition to how it can be utilized in their work. The latter has been emphasized since Orlikowski's (1992) studies of the non use of groupware due to lack of training in how computer tools can be used for supporting users' tasks. Novice users still complain "I had my file in Word, but now it's gone." Similar superficial learning has been observed among adults, as stated in a special issue of the Communications of the ACM in 1995 (Compeau, Olfman, Sein, & Webster, 1995). The call for improved user understanding was repeated by Sein et al. in 1998, but no reports on progress of research into user understanding has been reported. They noticed that training often focused on skills without understanding and suggested that also conceptual understanding of IT tools should be included. However, they did not specify which concepts to include.

Transfer of learning of principles is more likely than of discrete facts (Ormrod, 2004). This is the

---

Material published as part of this publication, either on-line or in print, is copyrighted by the Informing Science Institute. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission and payment of a fee. Contact [Publisher@InformingScience.org](mailto:Publisher@InformingScience.org) to request redistribution permission.

reason to assume that transfer of learning from one program to another or to the next version of a program is enhanced when understanding basic concepts, upon which most computer programs are built. The following mechanisms constitute basic principles that are found in many frequently used programs:

## Users' Learning of Principles

- Hierarchical data structures (file systems and web-sites) and the pointers (aliases and links) that may give access to a lower level in the hierarchy or other hierarchies.
- Template-instances relations (e.g., style-paragraph) and the possibility of modifying the template and thus impose global changes in the file.
- Data types have accompanying operations (e.g., number-calculation and file type-program), so that data should be stored by a type that is appropriate for processing.
- Functional dependencies (e.g., pointers to a web-page) and the usefulness of storing data only one place.
- Generalization-specialization relations (e.g., styles that inherit properties from other), and the benefit of storing descriptors only one place.
- Separation of structure and form (e.g., between the structure of a web-page and its layout), so that a structured set of data can be displayed with different formats.
- The layered architecture of software, implying that one type of data can be understood and processed in another way at a lower level, including the option of processing program code as data.
- The separation of hardware and software, opening for utilizing different types of hardware (e.g., hard disk and memory chip) for the same purpose and utilizing the same hardware (e.g., a music player) for different purposes (e.g., storage and video).

Template-instances and functional dependency constitute the two principles, selected for study in this paper.

The principle of functional dependency occurs frequently in spreadsheets, where formulas are dependent on the numbers entered in other cells. In text editors, there are references of various kinds, including table of contents, and this is also the principle that allows for entering data items only once in relational databases.

The principle template-instances is also frequently occurring in programming, e.g., when creating objects as instances of the class. It appears in text editors as the relation between a style and paragraphs, in presentation programs as master slides and slides, in spread sheets the type of cells can be selected, style sheets determine the layout of web-pages, and in all programs, the type of page layout can globally determine the appearance of individual sheets of paper in the print outs.

The research questions, which we address, are as follows:

1. How can students' learning of the principle template-instances and the principle of functional dependency in computer applications be characterised?
2. How can they transfer understanding of a principle generated from the use of one computer application to other applications?

Finding an answer to the question: 'How can we improve teaching so that students have better possibility to transfer the knowledge?' could be an expected consequence of the study, which is dependable on circumstances. Hence it could be considered as research contribution.

The setting of our study is attending classes in Information and Communication Technology subject, taught at an international high school. The population studied consists of 16 years old students, where four of the children were selected for in depth observation. At the field we focused on how students use a principle in practice, relating the cognitive theory as a resource/mediator for conducting the study.

The coming sections are arranged as follows: ‘Problem Area Concepts and Literature Review’ acquaints the reader with the concepts studied as well as presents the theoretical background for conducting the study; ‘Research Design and Research Methods’ introduces the methods used to search for possible answers to the research questions; ‘Research Findings and Discussion’ presents the fieldwork data and discussion; and the last section ‘Conclusion and Implications for Teaching’ gives a resume about the results of this preliminary study.

## Problem Area Concepts and Literature Review

Since computer principles are to be learned, the study draws upon research from the field of computer science education. Also literature on transfer of learning from the pedagogical sciences is presented, after the first sub-section ‘Problem Area Concepts’.

### **Problem Area Concepts**

A **computer application** is an executable program capable of performing a specialized function such as games, educational programs, and communications software. Word processors, databases, and spreadsheets are also examples of applications.

**Principle** is a basic generalization that is accepted as true and that can be used as basis for reasoning or conduct (WorldWebOnline, 2005). For the architecture of applications, the principles are decided prior to construction, unlike principles of nature, which are to be uncovered.

**The principle template-instances** in PowerPoint refers to the fact that people can generate the slides in their presentation based on the master slide, i.e. all slides in the presentation get the same text characteristics, background colour and special effects as the ones in the template master slide. This principle is also used in other applications, one of which could be found in style-paragraph in text processors, which will be explored in our further work.

**Functional dependency** is the most significant concept in normalization and the one most applicable to Document Engineering, as mentioned by Glushko and McGrath (2005). In general terms, dependency describes the impact on one object of change to another. Functional dependency is a specialized form of dependency. It means that if the value of one component changes when the value of another component changes, the former is dependent on the latter. Formally stated this is:

Given an object X, property A of X is functionally dependent on property B of X, if and only if each A-value in X has associated with it precisely one B-value in X (at any one time).

In transactional documents we find rich sets of dependency rules that apply to components that deal with organizations, addresses, taxes, payments, deliveries and personal details. Dependency rules may even be explicitly stated in documents such as forms. These may indicate the relationships of components using instructions for their completion. For example, an instruction like “If you are the owner of the vehicle, complete section 17,” identifies one part of the form as being owner information and section 17 as being dependent on the status of the applicant.

The principle of functional dependency identifies essential semantic components and reusable patterns, according to Glushko & McGrath (2005). In more narrative types of documents, dependencies among components are often expressed in cross-references, footnotes, and other kinds of links between content.

The way, in which students learn and might transfer understanding of this principle, will be evaluated further according to our main fieldwork.

## **Concept Learning in Computer Science**

Aharoni (2000) investigated students' learning of data structures: "The three thinking types express stages of abstraction presented in the **Actions-Process-Object model**, which is considered today as one of the central models of concept formation and is discussed by many researchers. According to this model, the formation of a new concept of a mathematical entity begins with *actions* performed on some physical or mental objects. When the learner gains the ability to refer to these actions using symbols and in input/output manner, without carrying out the specific actions themselves, we say that the actions were transformed into a *process*. The last stage is transforming the process into an *object*: The learner can now refer to the entity much like she refers to a physical one. She sees it as a whole static entity, recognizable at a glance, which may be thought of without any reference to its process view. Now the new formed object may be used as input to new actions, which will be transformed to a process, then to a new 'more abstract' object, and so forth. This whole procedure is, in fact, one procedure of abstraction."

The three stages are intended characterising learning of mathematical concepts. Principles behind computer applications do not appear in terms of formal languages for the users, but their semantics is as precise and universal as those of mathematics. Therefore, these stages are adopted as a basis for the study of learning of computer principles.

Based on the Actions-Process-Object model, we identified six stages in the students' process of learning the principles. Learning functional dependency could involve first learning the principle of dependency between cells and a graph in a spread sheet, while learning the general principles applying to all applications requires a second round of abstraction. For each of the two principles studied the first three stages (t. e. first, second and third stage) are forming the first cycle of the Actions-Process-Object model of abstraction, and the second three stages (t. e. forth, fifth and sixth stage) - are forming the second cycle of this model.

**Regarding the principle of functional dependency**, the *actions stage of table-chart dependency* (the first stage) might be the students' ability to operate the menu choices for inserting a chart. In the *process stage of table-chart dependency* (the second stage), the student should be able to talk about the actions and explain preconditions and results, e.g. that a formula computes a number based on the values of other cells. In the *object stage of table-chart dependency* (the third stage), the learner should be able to use the concepts of formula and hyperlink when discussing the construction of a spreadsheet of a document.

The concept of functional dependency is more general than those of formula and hyperlink, and hence we can apply here the second cycle of the Actions-Process-Object Model. Recognising similarities is a first step in generalisation, so the actions of recognising the dependency principle in formulas and hyperlinks, regardless of which application they appear in, will constitute the forth stage of the students' process of learning this principle, called *actions stage of functional dependency*. In the *process stage of functional dependency*, (fifth stage) the learner should be able to use the term "functional dependency" for talking about the commonalities of the two concepts. In the third *object stage of functional dependency* (sixth stage), the student should be able to compare the concept of functional dependency against other principles of computer operation.

**For the principle template-instance**, the actions of using the master slide in order to format the slides in a presentation application constitute the first stage, called *actions stage of master-slides relation*. Second, explaining what happens including the results at the slides is the *process stage of master-slides relation*. In the *object stage of master-slides relation* (the third stage) the learner should be able to discuss when to use the master slide and when to make exceptions.

The *actions stage of template-instances relation* (the forth stage) would be to recognize the template-instance association also in other functions, e.g., print layout-page or style-paragraph in text

processors, regardless of which application they appear in. In the *process stage of template-instances relation* (fifth stage), the learner should know the term “template-instance” and express similarities and differences between how the principles appear in programs and functions. Third, when template-instance is objectified, the students should be able to compare it with functional dependency and other principles. The latter is the *object stage of template-instances relation* (sixth stage).

## **Transfer**

According to Atkinson Atkinson, Smith, Bem, and Nolen-Hoeksema (2000) organisms generalize what they have learned and generalization can be curbed by **discrimination training**. If a young child is reinforced by her parents for petting the family dog, she will soon generalize this petting response to other dogs. Since this can be dangerous (the neighbours might have a vicious dog), the child’s parents may provide some discrimination training so that she is reinforced when she pets the family dog but not the neighbour’s. Discrimination training will be effective to the extent that there is a discriminative stimulus that clearly distinguishes cases, in which the response should be suppressed. Our young child will have an easier time learning which dog to pet if her parents point to an aspect of dogs that signals friendliness, for example. In general, a discriminative stimulus will be useful to the extent that its presence predicts that the response will not be followed by reinforcement (or vice versa).

We believe that the students’ ability to discriminate one principle from another is very important and hence the topic of discrimination is also relevant for the study.

As mentioned by Ormrod (2004) when something you learn in one situation affects how you learn or perform in another situation, transfer has occurred. Sometimes you transfer knowledge and skills you have previously learned to solve a problem, hence **problem solving is a form of transfer**. The author defines transfer as an essential component of human functioning because people continually encounter new situations and draw on previously acquired knowledge and skills to deal with them. Without transfer people would have to learn from scratch about how to behave in every new circumstance and would spend much of their time in trial-and-error learning. There are two types of transfer: specific and general. In **specific transfer** the original learning task and the transfer task overlap in some way. For example, having knowledge about the anatomy of a human should help a veterinary learn the anatomy of a dog, because the two species have many parallel anatomical features. In **general transfer**, the original task and the transfer task are different in both content and structure. For example, if knowledge of Latin helps a student learn physics, or if the study habit a student develops in a physics course facilitates the learning of sociology, then general transfer is occurring.

In this paper we will concentrate on specific transfer of understanding of principles from one computer program to another. One of the factors that can help us predict when transfer is more likely to occur is stated in the previously referenced book by Ormrod (2004). **It is that principles are more easily transferred than discrete facts**. For example, when students are trying to understand such current events as revolutions and international wars, general principles from history - for instance, the principle that two groups of people will engage in battle when other attempts at reaching a mutually satisfying state of affairs have failed – are probably more applicable than precise knowledge of World War II battles. And when students study geography, general map interpretation skills – for instance, determining why various features are located where they are – are probably more useful than the names and locations of specific rivers and state capitals.

We relate the fourth, fifth and sixth stages from the abstract stages in the process of student learning (introduced in the previous sub-section), to the concept of transfer, regarding transferring un-

derstanding of a principle generated from the use of one computer application to other applications.

## **Research Design and Research Methods**

This section is structured into two sub-sections as follows:

### ***Research Design***

With the purpose to find out how students learn principles (our first research question), we identified abstract stages in the process of learning for the students and related the cognitive theory to the six stages identified. We focused on Year 10 students, who were attending ICT subject and writing assignments for this subject, where four of the children were selected for in depth observation. We carried out observations of these students' participation at class, as well as the process of writing of their projects in relation to the abstract stages mentioned. We have also asked the teacher for collaboration in the way that she introduces the two principles, explains their ideas and also how they work in various computer programs as part of her teaching. Additionally, we conducted in the beginning of the school term an interview with her, and then continued carrying out the observations.

We were not allowed to interview the students, however, we had permission to observe the students and tape-record their classes. Hence we decided to concentrate on their work at class and their assignments in order to see how they perform. We were also permitted to add questions to the students' exam.

### ***Research Methods***

The rationale for using both qualitative and quantitative data is that a useful questionnaire of students' experience in transferring understanding of principles could best be developed only after a preliminary exploration of how students learn.

Before the observation we sent our short biographical documents to the teacher, who introduced us to the class: Year10. Then on the first day of the observation we gave to the students the Beginning Test, designed after conducting the interview with the teacher. The test which of a questionnaire, containing background questions and open-ended questions, and also some exercises in Word and PowerPoint (PP) closely related to the focus of our study. The purpose of the test was to define the level of understanding of the students. Additionally, after the test we made a summary of the results and choose to observe four of the students closely. The criteria (control elements) for choosing exactly these four students from the number of 22 students are stated as follows: previous experience with Word and PowerPoint, gender, and English language knowledge. The four students were two females (named A and M), and two males (L and O). Before we conducted the observation on the second day in class, we arranged with the teacher that she mentions for the first time the idea of the principle of functional dependency (without naming it), as giving an example in connection to the fact that when students insert graph, they will use the command: Insert→Picture→Chart. Then, she explained that, if they change the datasheet (table) connected to the chart, the graph itself will automatically be changed. Hence the actual numbers in the graph are dependent on the cells in the datasheet. This example, which the teacher gave, was arranged to be shown to the students in order to check later if, after the teacher mentions the idea of the principle of functional dependency, students will be able to understand and apply it in their assignment, and to what extent?

On the second day of observation, the students were given an individual first assignment (t. e. assignment 1: to write a document in Word, in which they needed to type a text provided by the teacher, and to insert in this Word document one table and one graph, written first in Excel). Then

we tested two methods for observing, either closely observing the four students, or observing the students who asked questions. At this point we decided to use both methods for observation in relation to the fact that the students, not only from the two groups, but globally, were collaborating with each other and with the teacher during the whole time, and there was continuous interaction going on both within one group and outside it, which was interesting to follow. During the observation of the students' work on assignment 2, we used mostly the second method for observation, because all the students were introduced to the idea of the principle of functional dependency in Excel, without the teacher mentioning the concept, and it was valuable to watch what students do and say, and how do they understand or not understand the idea of this principle, in relation to the six abstraction stages. While later on at assignment 3, we used mostly the first method for observation. That was because at that time we have already explained thoroughly the two principles, with their names, only to the female group. Hence we would like to observe this group closely: how do they learn or not learn the principles and to what extent they are able to transfer their understanding.

Before assignment 3 the teacher presented a short lecture about PowerPoint to all students, in which she has added and mentioned the terms 'template-instances' and 'functional dependency' by giving several examples of the two terms. For example, she related the term 'template-instances' to the command View→Master→Slide Master, as well as the command File→New→Templates→On my computer. Additionally, she related the term 'functional dependency' to the command 'Insert→Hyperlink and under Link to click Existing File or Web Page,' explaining that two separate files linked together are functionally dependent on each other. After that she presented assignment 3, which was writing a presentation with the use of PowerPoint.

## Research Findings and Discussion

This section presents the empirical findings of the study, structured into three sub-sections, as well as discussion of them. Additionally the fourth sub-section of this section gives an overview of the four multiple-choice questions given to the students at the end of the term and summarizes the results in eight graphs.

### ***Case 1: Copying Instead of Inserting***

One of the requirements for assignment 1 was to insert an Excel chart in Word. However, **O did not insert the chart into his Word document, but copied it**, thereby removing the option of creating a graph dependent on the table.

Having in mind that the teacher has written in her evaluation requirements that the students need to 'insert Excel chart into their Word document', we might think that O has not understood the difference between copying a chart and inserting a chart. This situation could be related to the first stage from the six abstract stages of the process of students' learning t. e. *actions stage of table-chart dependency*. Hence we might say that this student has not grasped the point of the assignment in connection to the principle of functional dependency.

One possible explanation for that O has not inserted the Excel chart in Word but has copied it, can be that he has not read the evaluation list, which requires inserting. Another explanation might be that the student has read the evaluation list but has not interpreted it correctly.

Furthermore, the student has used a command, conveying the idea of the principle of functional dependency - Insert→Chart, in Excel - at the moment of creating the chart based on the data table in Excel. However, he has not recognized that he needed to use another command, conveying the idea of the principle of functional dependency - Insert→Object→From File and pointing to his Excel Worksheet containing the chart, in Word. If he had used this second command, the Excel

chart in Word would be dependent on the data table in Excel, when the Excel data table worksheet is open.

In order to characterize the students' understanding of the commands which they have used in relation to the two principles studied, we identified two possible sidetracks and one aid-track. The first sidetrack we called '*discrimination error*' which refers to lack of a discriminative stimulus that clearly distinguishes cases (Atkinson et al., 2000). In other words, this is the situation when the student cannot recognize an aspect of the command, which signals about its principle as different from the principle of another command. The student has recognized the first time that he needed to use the following command conveying the idea of the principle of functional dependency: Insert→Chart in Excel, but has not recognized that he needed to use another command in order to obtain the dependency in Word. This is the first (of two) example regarding the principle of functional dependency.

### Case 2: Numbers Instead of Formulas

In this sub-section we will present second example regarding the principle of functional dependency, as we look at L's assignment 2.

Assignment 2, given on the third day of our observation, was to graph the following functions all on the same set of x and y axes:

$$y=2*x+3 \quad y=2*x-3 \quad y=-2*x+3 \quad y=-2*x-3 \quad y=1/2*x+3 \quad y=4*x+3$$

as students need to use the values of x from -3 to 3. Additionally, they had several questions to answer related to their Mathematics subject.

One of the comments by the teacher to L's spreadsheet was "**Excel formulas were not used.**" Figure 1 illustrates part of L's work on assignment 2:

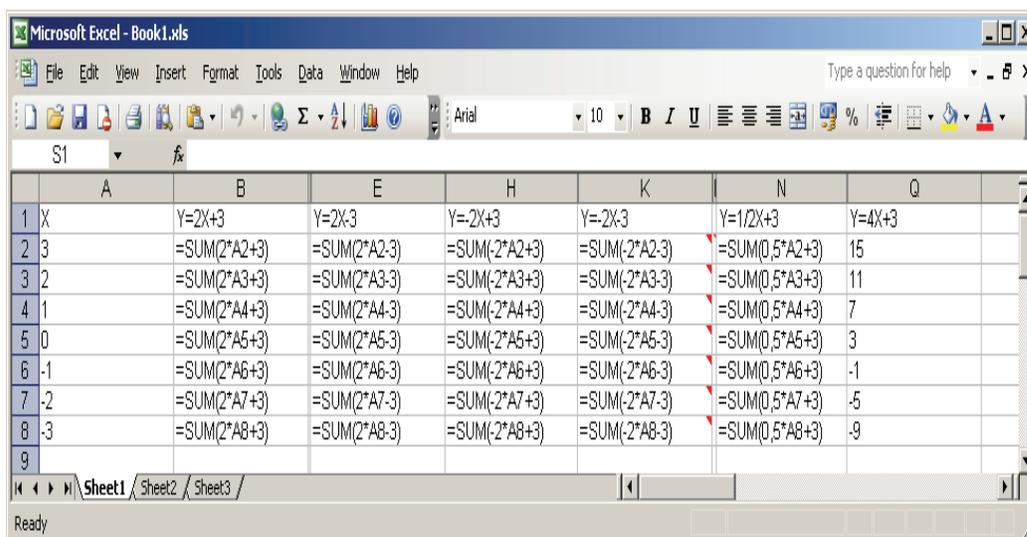


Figure 1: Screenshot, showing L's work on assignment 2, done in Excel.

After having a look at L's assignment 2 file, we can say that he has not used Excel formula in order to calculate the last function values:  $y=4*x+3$ . He had calculated it by pen and pencil, and then directly written the results in Excel.

That might be related to the idea that he had not completely understood the principle of the Excel command required to be used, and hence it might be an obstacle for him to learn this command. This case could be related again (as case 1) to the *actions stage of table-chart dependency* (the first abstraction stage). As L has used the command for the first five steps and after that has stopped using it for the last sixth step, this might be defined by us as the second sidetrack, called: *'forgetting'*. This sidetrack is interpreted in the given context as using the principle of the command for several steps to a certain moment, and after that stopping to use it in the next steps. In other words, the student remembers the principle to a certain moment, and then he/she forgets it. We could say that it might force the student L to understand the principle until a certain moment, which depends on circumstances. We believe that before this certain moment L has used the idea of the principle of functional dependency, but after that – has not. The sidetrack 'forgetting' could hinder O from reaching the process stage of table-chart dependency.

### **Case 3: Applying Master Slide with Exceptions**

In this sub-section we will present an example of abstraction of the principle template-instances.

During the observation of the female group's work on assignment 3 we followed closely student A, who used the command: 'View→Master→Slide Master' and then clicked on: 'Close Master View'. She used it in order to make the background and some other settings of all slides the same, based on the master slide, of which there was only one in the version of PowerPoint that the students were using. She was concentrated on editing the title slide of the presentation, while, at the same time, the other girl, called M, was making one of the other slides, which was going to be added as number 4 at their presentation, by typing text in a blank presentation sheet.

1. The first student A made the file a.jpg filled with the text: Alfred Nobel, written several times. After that she opened PowerPoint and used the command: Format→Background. Then from the list she chose: Fill Effects→Texture pressed the button: 'Other Texture' and browsed to her file: a.jpg. Then she pressed: 'Apply', which applied the background to the first slide, see Figure 2. When she chose the background and some other settings, A inserted three pictures of Alfred Nobel (Figure 2) by using the command: Insert→Picture. Then she saw that the background is visible under the picture (the picture and the background are mixed) and said: "Oh, no." The other student M asks her: "What is the difference?" Then she replied: "I do not know, it is not anything cool of it. Come on, Master, going. Oh..." (Here the whole dialog on the tape is difficult to hear because of noise from construction work from outside) That is why she tried to change the background with another one but still it did not work. Then she came back to the previous background. After that, A started creating the next two slides after the first one (t. e. slides number 2 and 3) by using the command: Insert→New Slide (they came with automatic background fill), as for each of the two slides she changed the background by following the steps: first using the command: Format→Background, and then from the list she chose: Fill Effects→Texture pressed the button: 'Other Texture' and browsed to her file: a.jpg. Then she pressed: 'Apply', which applied this background to the second and third slide (t. e. the first three slides had the same background defined by her). After that she pressed Insert→New Slide, and **the inserted fourth slide was with automatic background fill** (t. e. the fourth slide had automatic/white background). She did not change this time the background of this slide, because she had probably in mind that later on in this slide (number 4) will be copied the information from the slide prepared by the other student M (when M is ready with it, please look at Figure 4). After that A inserted again a new slide (number 5, please, look at Figure 3) and started changing its background. She used the command: Format→Background. Then from the list she chose: Background Fill: black (t. e. **the fifth slide had black background**).

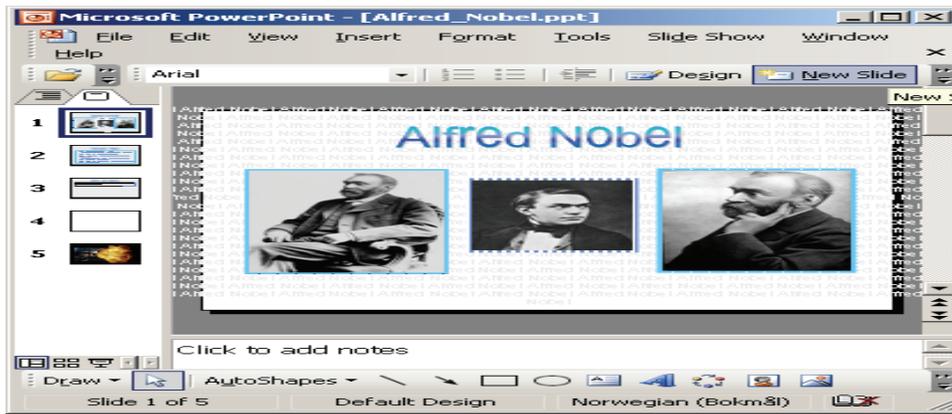


Figure 2: Screenshot, showing A's work on assignment 3, done in PowerPoint (this is slide 1 from her presentation; slide 2 and slide 3 have the same background fill as slide 1).



Figure 3: Screenshot, showing A's work on assignment 3, done in PP (this is slide 5 from her presentation; its background fill is changed).

2. M wrote slide number 4. The following Figure 4 illustrates her work on assignment 3:

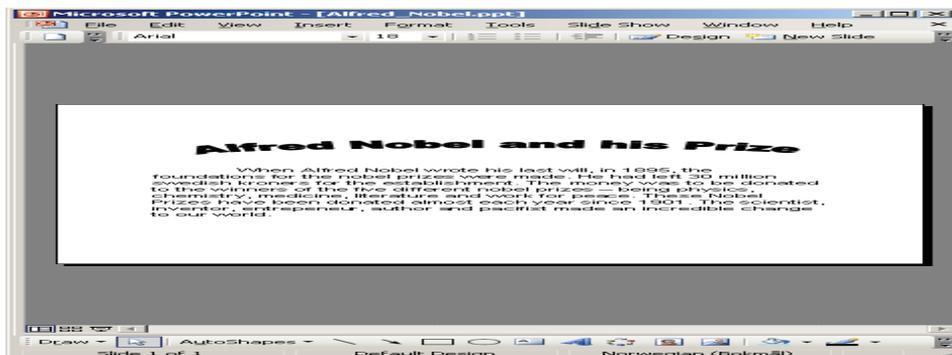


Figure 4: Screenshot, showing M's work on assignment 3, done in PP.

In this case A might have changed the background of slides number 4 and 5 deliberately having control over this change, and she probably would like to make the presentation with different backgrounds of these two slides, according to her taste. Additionally, she demonstrated not only that she understands that the command: ‘View→Master→Slide Master’ applies the background of the Master slide to all other slides, but also that one can change after that the background of these slides in the way he/she would like to have them.

As mentioned before, in order to characterize the extent, to which A understands the idea of the principle template-instances, we defined one aid-track, called: ‘*exception-handling*’. This aid-track describes the situation when one uses a command, conveying the idea of the principle template-instances, in several steps, and does not use it in other steps deliberately. In other words, he/she makes exceptions from using this command in the other steps, being aware of the contexts, in which one need to use this command and those, in which he/she do not need to use it. In our case, A has started using the command, conveying the idea of the principle right in the beginning, then she decided to make exceptions from using this command and has done this correctly showing understanding regarding the principle until the end of her assignment. She did not discuss in which contexts she could use the command and in which not, so she did not demonstrate the third stage of treating master-slide as an object. But in order to figure out that she needed an exception and also how to do it, she must have conceived the *process stage of master-slides relation*.

Table 1 reviews the relation between the three cases and the abstract stages, as follows:

**Table 1: Cases – stages relation**

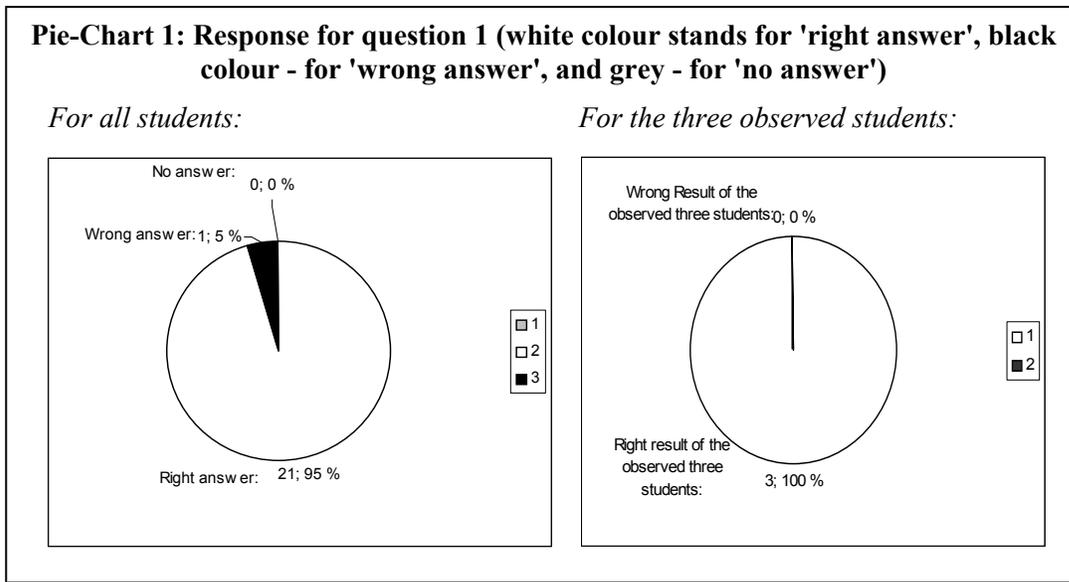
Case 1	Case 2	Case 3
<i>actions stage of table-chart dependency</i>	<i>actions stage of table-chart dependency</i>	<i>process stage of master-slides relation</i>
Side-track discrimination error	Side-track forgetting	Aid-track exception

## Test

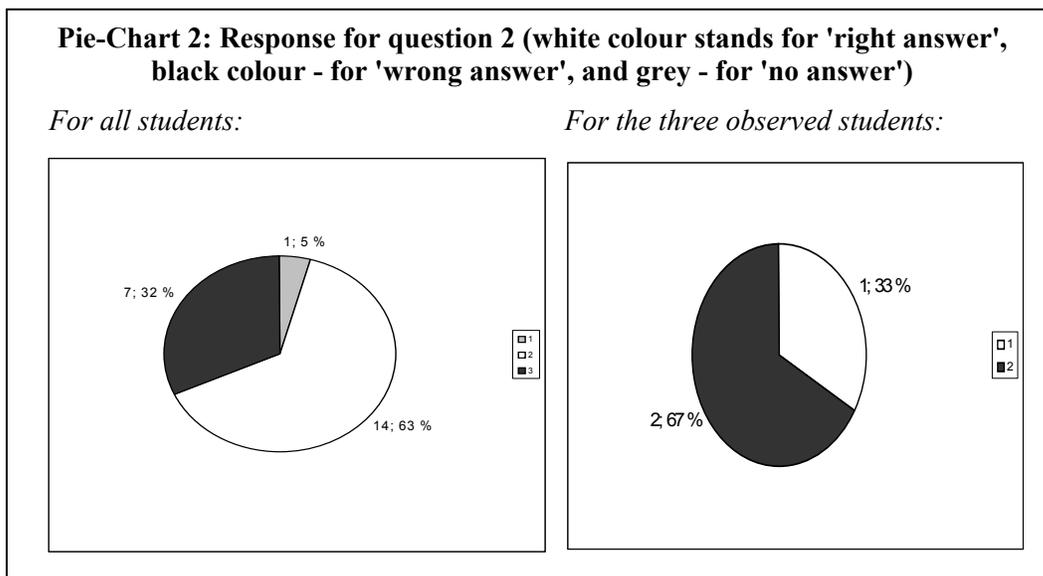
Before the end of the school term we added four multiple-choice questions, related to the two principles studied, to the exam of the whole class and gave it to the twenty-two students. The questions and the pie charts presenting results (please, see Pie-Chart 1, 2, 3 and 4), are as follows:

1. In *Word*, select on the menu: **Insert/Picture/Chart**. What happens if you change the Data-sheet (table) connected to the chart?
  - a) The chart itself will not be changed.
  - b) This changes the chart type.
  - c) The chart itself will automatically be changed.

This first question refers to students’ understanding of the mentioned command in *Word* in relation to the principle of functional dependency. As the majority of the students (and also all observed students) have given a right answer to it, we claim that the students easily operate the command, conveying this principle, and are able to explain what this command do, i.e. for this command they have reached the *process stage of table-chart dependency* (the second stage).

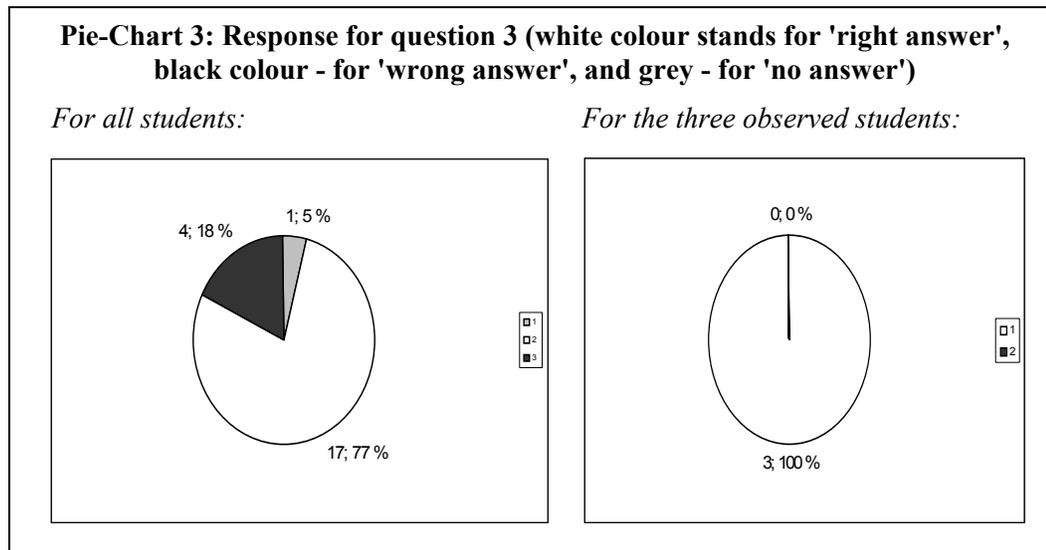


2. Open *Word* in one window, select on the menu: Insert/Hyperlink and under Link to click Existing File or Web Page. Open *Excel* in another window, select on the menu: Insert/Picture/From File. Are there similarities between the two commands mentioned above (from *Word* and *Excel*)?
  - a) Yes. The principle is the same (functional dependency).
  - b) No. The principle is not the same.
  - c) Yes. The principle is the same (template-instances).



The second question refers to transfer of the principle of functional dependency from one computer application to another. As two-third of the students (and one of the three observed students) have given the right answer, we claim that these students have reached the forth stage t. e. *actions stage of functional dependency*. However, the rest of the students have not reached this stage.

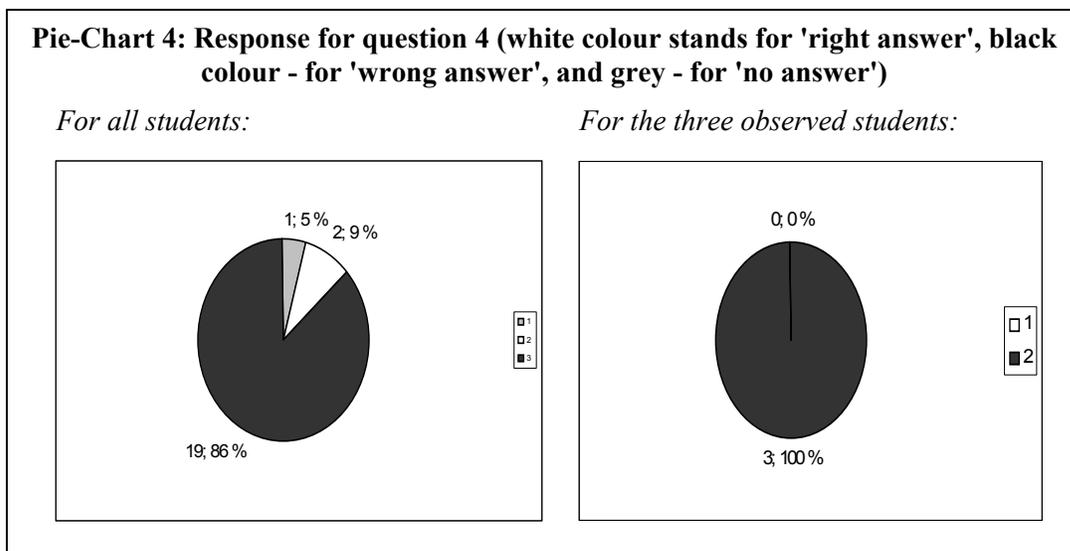
3. In *PowerPoint*, when selecting on the menu: **View/Master/Slide Master**, which of the following happens?
- Slide Master** controls the look for all slides regarding: only Font type.
  - Slide Master** controls the look for all slides regarding: only Background colour.
  - Slide Master** controls the look for all slides regarding: only Bullets.
  - All the above.
  - Slide Master** controls the look for all slides regarding: only Font type and Background colour.



The third question is related to students' understanding of the mentioned command in *PowerPoint* based on the principle template-instances. As the majority of the students (as well as all observed students) gave a right answer to this question, we could claim that for this command they have reached the *process stage of master-slides relation* (t. e. the second stage).

4. Open *PowerPoint* in one window, select on the menu: **View/Master/Slide Master**. Open *Word* in another window, select on the menu: **File/New**, then in the opened task pane **Templates** click **On My Computer**. Are there similarities between the two commands mentioned above (from *PowerPoint* and *Word*)?
- Yes. The principle is the same (functional dependency).
  - Yes. The principle is the same (template-instances).
  - No. The principle is not the same.
  - Yes. The principle is the same (both functional dependency and template-instances).

The fourth question concerns transfer of the principle template-instances from one computer application to another. As 86% of the students have given wrong answer to it (and all three observed students also), we claim that the students have not reached the fourth stage *actions stage of template-instances relation*.



## Conclusion and Implications for Teaching

### Conclusion

This paper presented two examples of using commands, conveying the idea of the principle of functional dependency as well as one example of using a command, conveying the idea of the principle template-instances, and explained potential reasons for them. As a discussion of these examples we will relate again the six defined stages based on the Actions-Process-Object model.

Having in mind the data from the three assignments, we can point out that for the observed students the concept 'actions' is performed in all three cases. These 'actions' are not transformed into 'process' in the first two cases because of a sidetrack, such as 'discrimination error' or 'forgetting', which hinders the student from understanding of the command used in relation to the principle of functional dependency. While only in the third case the student has showed understanding of the command used in relation to the principle template-instances, owing to the aid-track 'exception-handling', which indicates that she is at the second level. However, 'process' is not transformed into 'object' in neither of the cases t. e. the third stage is not reached. Additionally, the results of our test showed that most of the students, including the three closely observed, were at the second, process stage at the end of the term. Furthermore, two-third of the students demonstrated competence of the action stage of functional dependency (fourth stage). However, few of the students could do the same with template-instances in the exam question.

### Implications for Teaching

Learning ladders of the type that Aharoni (2000) have suggested constitute useful guidelines for teachers both when helping students working with the applications and when planning teaching sessions.

During the action stage learning, teaching can point to possible discrimination errors so students become aware of them prior to practice. When supervising, teachers can remind learners in order to prevent forgetting. In order to reach the process stage, students need to verbalise their actions, e.g., by creating exceptions like illustrated in the third case. For reaching the object stage of table-chart dependency, the students need to learn both the term and the contents of the concept. Such concepts are seldom found in course literature for user training, so teachers and researchers have

to invent these missing links in the chain of user competence. Recognizing and discussing the differences between two commands and deciding when to use one command and when not to use it can be useful exercises during the object stage.

In order to reach the action stage of functional dependency, the learners should consolidate the object stage of table-chart dependency or another type of functional dependency. Thereafter, the learners need to get acquainted with other types of the principle, e.g., the dependence between a cell value and the functions involving this cell in a spreadsheet or an automatically generated table of contents in a text processor. The process stage can be learnt after introducing the concept and discussing the similarities and differences between the various types of functional dependency relations. The object stage, where functional dependency should be compared with other types of relations, constitutes an advanced understanding that may be above the level needed for the common user.

This paper is a preliminary study into the investigation of how the research questions might be explored. The six stages identified in the paper form a model, which later on will be used to do further observations.

## References

- Aharoni, D. (2000). Cogito, ergo sum! Cognitive processes of students dealing with data structures. *ACM SIGCSE Bulletin* 32(1), 26 – 30.
- Atkinson, R. L., Atkinson R. C., Smith E. E., Bem D. J., & Nolen-Hoeksema S. (2000). *Hilgard's introduction to psychology* (Thirteenth ed.). Fort Worth: Harcourt College Publishers.
- Compeau, D., Olfman, L., Sein, M., & Webster, J. (1995). End-user training and learning. *Communications of the ACM*, 38(7), 24-26.
- Glushko, R. & McGrath, T. (2005). *Document engineering: Analyzing and designing documents for business informatics and web services*. Cambridge, Mass.: MIT Press. Chapter 13, from which we quoted, is retrieved November 10, 2006, from <http://www.ischool.berkeley.edu/~glushko/andrea/> (please, open the link: Chapter13-FINAL.doc).
- Orlikowski, W. (1992). Learning from notes: Organizational issues in groupware implementation. Computer Supported Cooperative Work, *Proceedings of the 1992 ACM Conference on Computer-supported Cooperative Work*, Canada, pp. 362 – 369.
- Ormrod, J. (2004) *Human learning* (Fourth ed.). Upper Saddle River, NJ, USA: Pearson
- Sein, M.K., Bostrom, R.P., & Olfman, L. (1998). Re-conceptualizing IT training for the workforce of the future. *Proceedings of the Conference on Computer Personnel Research*, Boston, Massachusetts, United States, pp. 233-241.
- WorldWebOnline Website (2005). Dictionary and thesaurus. Retrieved November 16, 2006 from <http://www.wordwebonline.com/en/PRINCIPLE>

## Biographies

**Elena Stamatova** is currently last year Master Degree student in Information Systems at the Department of Informatics, University of Oslo, Norway. Her research interests are in the area of learning and ICT.

**Jens Kaasbøll** is a professor at the Department of Informatics, University of Oslo. He has published in the areas of information systems development, object-oriented modeling, and computer science education, including a recent book "Comprehensive Object-Oriented Learning: The Learner's Perspective." He is the leader of a Masters program on information systems at the Eduardo Mondlane University in Mozambique and has been a member of a team who developed the curriculum for computer science education in Norwegian high schools.