2009

Innovative Pedagogical Analysis Of A Distributed Systems Course

Rasmus Ulslev Pedersen
Copenhagen Business School, rup.inf@cbs.dk

Michael Pedersen
Copenhagen Business School, mp.ll@cbs.dk

INNOVATIVE PEDAGOGICAL ANALYSIS OF A DISTRIBUTED SYSTEMS COURSE

Rasmus Ulslev Pedersen
Department of Informatics
Copenhagen Business School
rup.inf@cbs.dk

Michael Pedersen
CBS Learning Lab
Copenhagen Business School
mp.ll@cbs.dk

Abstract:

IS teaching is often part of an interdisciplinary program. Teaching in an interdisciplinary program can be challenging because many core subjects carry a large body of so-called prerequisites with it. We analyze one such class taught in an interdisciplinary IT/IS program at a large business school. The class is on distributed systems, and we show how these challenges have been addressed in the present program. Furthermore, we develop a toolbox of methods and ideas for future teaching of similar programming classes taught in interdisciplinary IS programs.

Keywords: Interdisiplinary teaching, distributed systems, challenges, new approaches

I. INTRODUCTION AND PROBLEM

In this paper we will draw on a discipline called Distributed Systems (DS). It is a subject which we have taught many times. We want to discuss the teaching tools we have used this time, and a group of new things which we would consider using the next time the subject is taught. So the kernel and thus what we contribute to IS education in this regard are some innovative ways to teach in the future. A technical subject such as distributed systems (DS) is a little different than other subjects at Copenhagen Business School, because it is a core computer science subject taught in an interdisciplinary program.

The task is built up so that we give a little background for the subject to begin with. Then we discuss the course development in terms of the student reviews, and how this will undergo analysis before the next semester starts again. In addition, we show a simple model of communication, in a class to illustrate that it is a good idea of dialogue-generating activities in the classroom. Afterwards comes the section on the didactic (teaching method) issues we have identified this time and during previous times. Last but not least, comes the section about the new things we plan to use in subsequent semesters. In the last section, we will use visualization to make it easier to show what we mean.

II. BACKGROUND OF THE PROFESSION

Compared to other institutions, the distributed systems class is a classic course such as, for example, object oriented design and development. However, it not the whole truth. On an interdisciplinary study like the one at Copenhagen Business School (Business & IT), such a class much cover more than just the core subjects of DS. It must also cover the prerequisites that lead up to the DS class. It must cover subjects such as:

1. Network Communications
2. Operating systems
3. Databases
4. Distributed systems

It is almost an art to find a rhythm in such a course which has the right title, but which in practice covers an incredible amount of the discipline. We solve this particular by reviewing the prerequisites that needs to be in place before the actual review of a DS book chapter. The future of the class is promising. There are more and more distributed systems around us. Consider, for instance social networking sites or Google's search engine or the new cloud computing concept. Another reason that we believe makes the class promising, are the new didactic findings that we present later in this paper. They should ensure a good variety of teaching and exercises, and effective learning.

III. COURSE DEVELOPMENT FOR DISTRIBUTED SYSTEMS

We have observed a trend among some of the young people entering the study program. It is that they are essentially what we would call a "Windows generation". They have sometimes an expectation that programming, and other things in education, are as easy as typing in the Microsoft Word editor. Of course it is not quite the case, and the students realize that the stability of Microsoft Word is a result of many hours of programming by Microsoft. 10 years ago, when Windows crashed every now and again, it opened (the hard way) people's eyes up to the fact that computer systems were not infallible. But we cannot (and probably should not) change the new generations, but we must be more aware that they expect that things will work the first time. This is an exciting challenge. Already now we can state that at least one of the new teaching approaches we present will addresses this.

IV. MODEL FOR DISSEMINATION

We will show a graph of how the typical training room looks. The idea behind this is to show what tools a teacher normally have available in a classroom. Moreover, as we show, the majority of students bring laptops to class. Traditionally, teachers do not use this fact to something productive, but rather let the students sit and possibly doing other things. We think that the laptops as a resource and an opportunity to enrich the teaching situation.

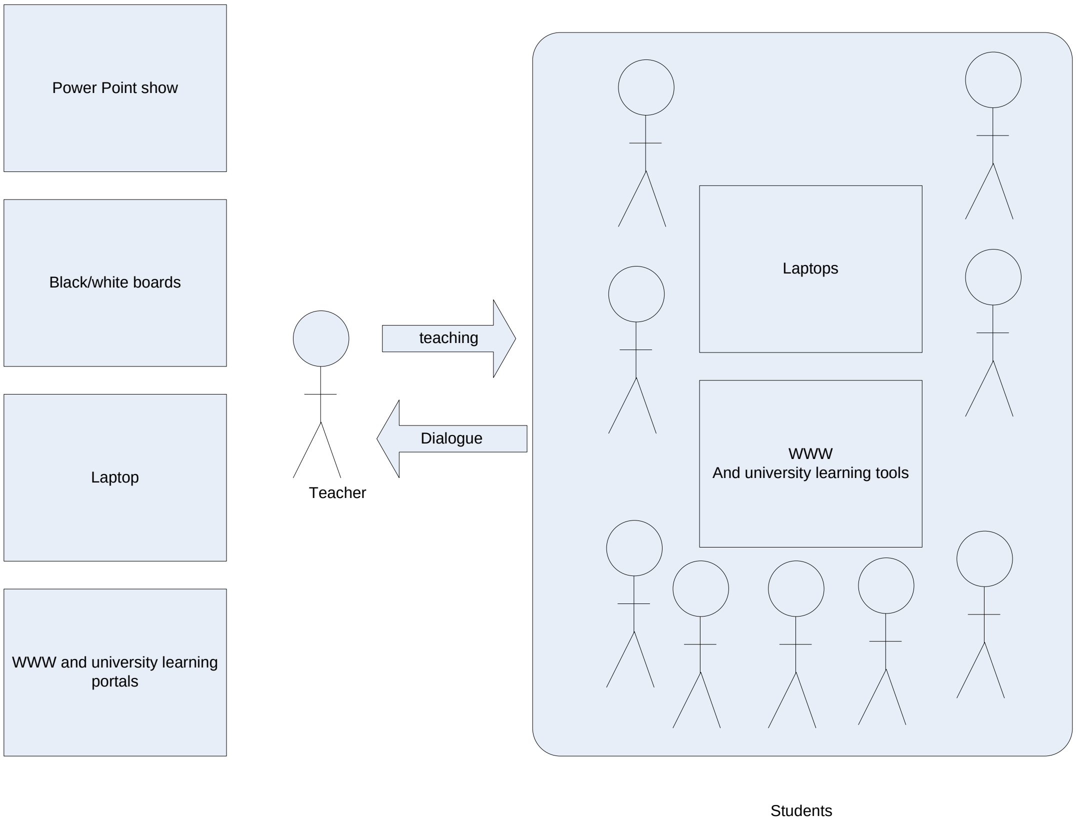


Figure 1. A classroom with a teacher and a group of students (shown as a box).

We will refer back to this figure from the following sections on current and future didactic "tools".

V. CURRENT DIDACTIC TOOLBOX: "VERSION 1"

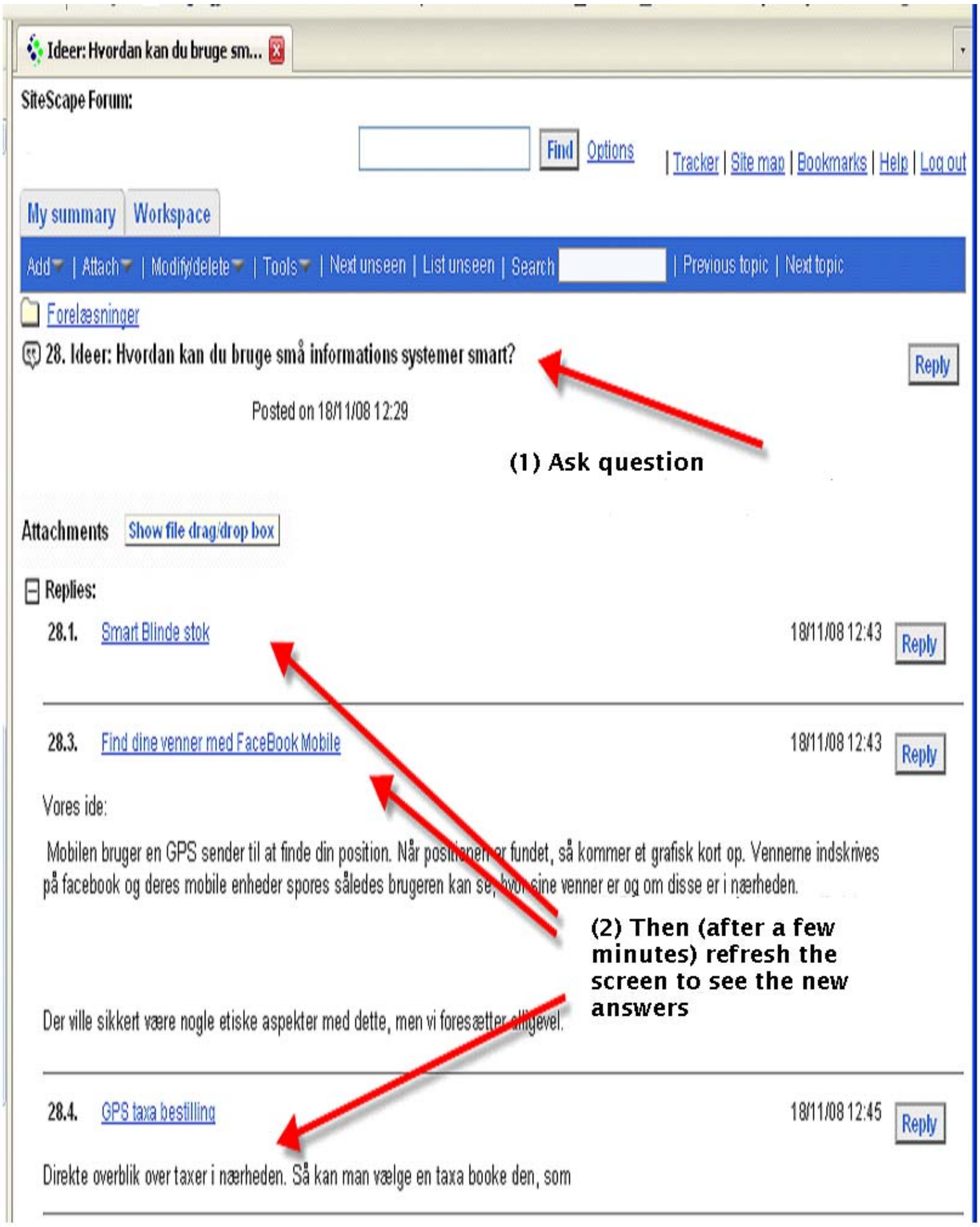
In this section, we will tell you about the things that we have used in teaching up to this point. Two of the four things are something that can be called new, and the rest is normally used by teachers.

NEW: SITESCAPE PAIR-CHAT

The idea here is to use Sitescape as dialogue tool. Sitescape is CBS's current teaching platform, which is similar to the Moodle system. On the teaching figure, we have shown that students bring laptops to class. This can be positively exploited by using Sitescape in the classroom. The background to this teaching instrument is that so-called "pair-chats" or "buzzs", where two students discuss something for a short while before being called upon to share the results. A "pair-chat" is in brief that the teacher asks the students to two and two take a few minutes about a topic and then be ready for a dialogue in class. The problem with the traditional pair-chat is that only a minority of students is heard afterwards. Furthermore, it is only the most outspoken who might dare to say anything afterwards.

The solution is the so-called Sitescape buzz session, which is something we invented. It introduces the topic and the teacher writes the question in a Sitescape forum thread. The teacher let the projector show the question on the screen, and in the meantime ask the students to reply to topic. When time is up, then the teacher presses the refresh button in their browser.

Below you see a finished Sitescape buzz session meeting:


Ideer: Hvordan kan du bruge sm...
SiteScape Forum:
Find
Options
| Tracker | Site map | Bookmarks | Help | Log out
My summary
Workspace
Add | Attach | Modify/delete | Tools | Next unseen | List unseen | Search | Previous topic | Next topic
Forelæsninger
Reply
Attachments
Show file drag/drop box
Replies:

28. Ideer: Hvordan kan du bruge små informations systemer smart?

Posted on 18/11/08 12:29

(1) Ask question

28.1. Smart Blinde stok 18/11/08 12:43

28.3. Find dine venner med FaceBook Mobile 18/11/08 12:43

Vores ide:

Mobilen bruger en GPS sender til at finde din position. Når positionen er fundet, så kommer et grafisk kort op. Vennerne indskrives på facebook og deres mobile enheder spores således brugeren kan se, hvor sine venner er og om disse er i nærheden.

Der ville sikkert være nogle etiske aspekter med dette, men vi foresætter alligevel.

(2) Then (after a few minutes) refresh the screen to see the new answers

28.4. GPS taxa bestilling 18/11/08 12:45

Direkte overblik over taxer i nærheden. Så kan man vælge en taxa booke den, som

Figure 2 Here, the teacher wrote a question on Sitescape during the lecture. There

are some responses from students. The text is edited on the trailing edge to show the point of this paper.

We think this is a good teaching tool. We already write our syllabus / course descriptions in this system, and here we use it for something new.

OVERVIEW DRAWING

The students obviously not the picture a teacher have in connection with learning new material. Therefore we have made a kind of red thread for them. It is a drawing showing the entire class material in a single Power Point slide. What we then do is to show the Powerpoint each time the class starts, telling the students two things:

1. Where have we been
2. Where we will go today

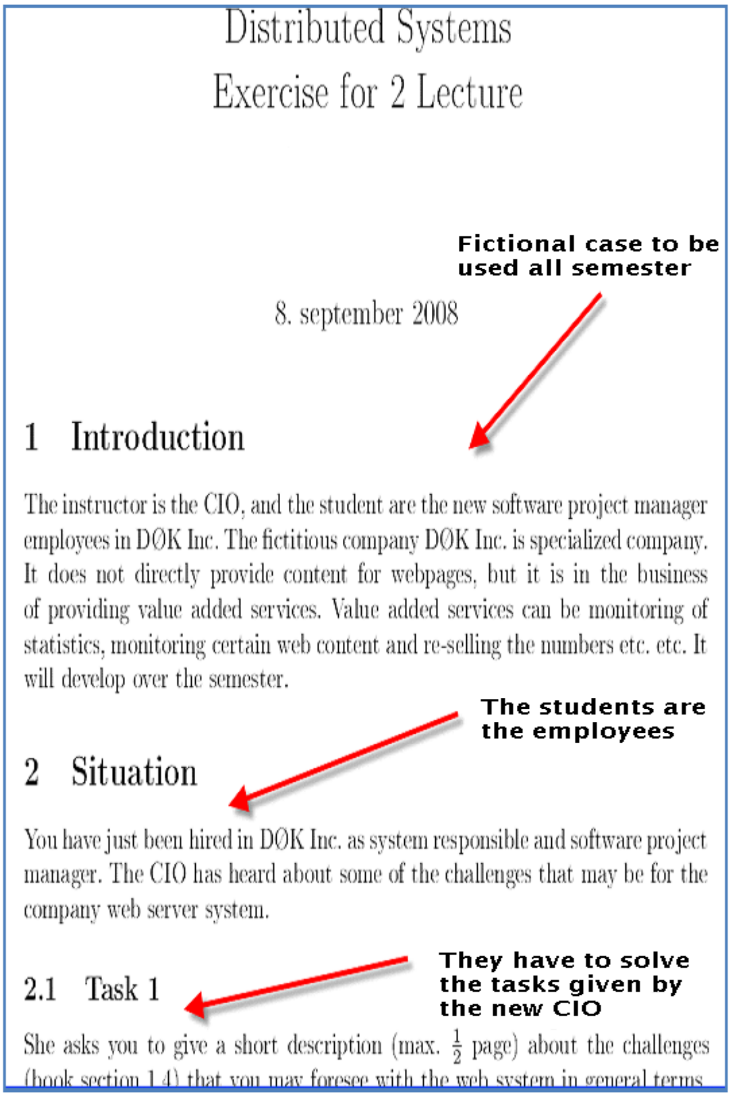
The tool works well and if we consider the term learning, then this slide will be imprinted into their brains for many years. That is not the case of the many Power Points slides that are displayed in the class.

Variation: We have previously started the class by drawing parts of this figure on the blackboard, and introduced the day's curriculum from the figure. The advantage are two things: (1) the figure is on the board throughout the lecture and we can refer back to it, and (2) it gives some variation since we do not need to turn on the projector until maybe 15 minutes inside the lecture.

NEW: EXERCISES WITH THE TEACHER AS INSTRUCTOR

We have also been their instructor, which we normally use students for. The reason for this, is that it is relatively difficult curriculum, and it is difficult to obtain two good instructors. We will most likely continue with it because it gave something unique to have the students in the exercises sessions. Namely the fact that we could make a much more nuanced approach to either push the brake and the accelerator than we normally have been able to those times when we have been teaching.

To conclude our approach to the exercises, we show here an outline of an exercise. The special thing is that we write exercises ourselves and present them as a case. This also gives control over the teaching momentum.



Distributed Systems

Exercise for 2 Lecture

Fictional case to be used all semester

8. september 2008

1 Introduction

The instructor is the CIO, and the student are the new software project manager employees in DØK Inc. The fictitious company DØK Inc. is specialized company. It does not directly provide content for webpages, but it is in the business of providing value added services. Value added services can be monitoring of statistics, monitoring certain web content and re-selling the numbers etc. etc. It will develop over the semester.

The students are the employees

2 Situation

You have just been hired in DØK Inc. as system responsible and software project manager. The CIO has heard about some of the challenges that may be for the company web server system.

2.1 Task 1

They have to solve the tasks given by the new CIO

She asks you to give a short description (max. $\frac{1}{2}$ page) about the challenges (book section 1.4) that you may foresee with the web system in general terms

Figure 3 Exercise from DS profession that is created as a case study.

VI. DIDACTIC TOOL VERSION 2

In this section we will present the new approaches that we work toward using. We think there is a significant innovative element to the new ideas, and it is a good things to have when the future teaching situation changes these years due to the many new multimedia opportunities.

So now we will introduce the new things can be done: It includes. Live programming, team programming, Power Point with-life and memory, virtual lab, and pod-cast.

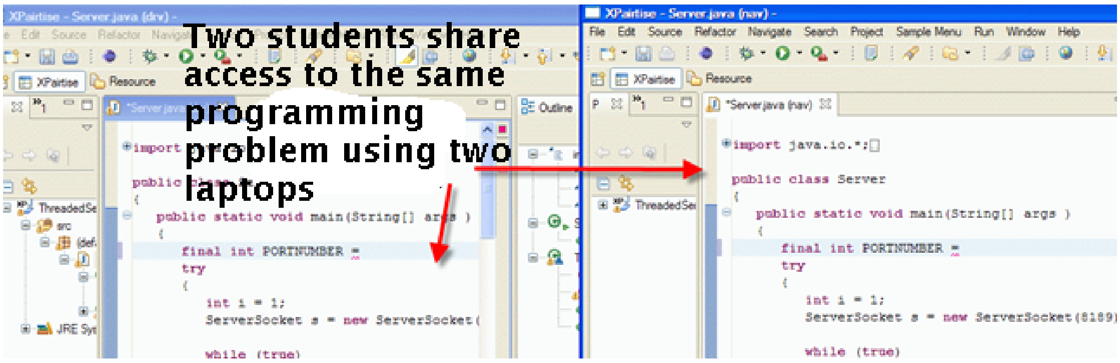
VII. NEW: TEAM PROGRAMMING

We can do more to enhance the programming experience by letting students work together in pairs. We have found a new way of doing this which makes up with status quo:

Status quo: “Team programming” currently takes place by two students sharing a single laptop.
Team Programming: You can program on two different computers and share the same screen.

This means that means that all students can gain practical experience. Normally it is unfortunately so that a few specialize in programming and other just look at it. Within programming, actual keyboard time is what counts most

.
Below we have two links to software that can be used for team-programming:
http://www.eclipseplugincentral.com/Web_Links-index-req-viewlink-cid-927.html
http://www.eclipseplugincentral.com/Web_Links-index-req-viewlink-cid-915.html


XPairtise - Server.java (drv) -
XPairtise - Server.java (nav) -
File Edit Source Refactor Navigate Search Project Sample Menu Run Window Help
XPairtise
Resource
Outline
*Server.java (nav)
ThreadedSe
src
JRE Sys
Two students share access to the same programming problem using two laptops

```
import java.io.*;

public class Server
{
    public static void main(String[] args )
    {
        final int PORTNUMBER =
        try
        {
            int i = 1;
            ServerSocket s = new ServerSocket(8189)

            while (true)
```

Figure 4 Team programming is carried out by two (or more) working together on the same files, but from different computer.

There is also a built-in chat in this system so that you can discuss the things to be done. If this will work in the DS class then it is a good contribution to new teaching methods.

NEW: LIVE PROGRAMMING IN LECTURE

This is true for much of the programming instruction we have received; the teacher only talks about it. I.e. the teacher says what must be done, but fails to show how it's actually done.

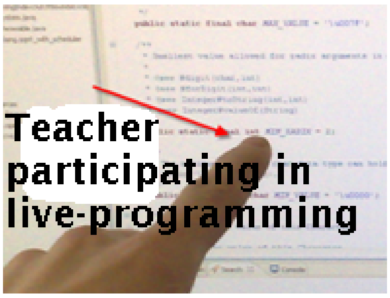
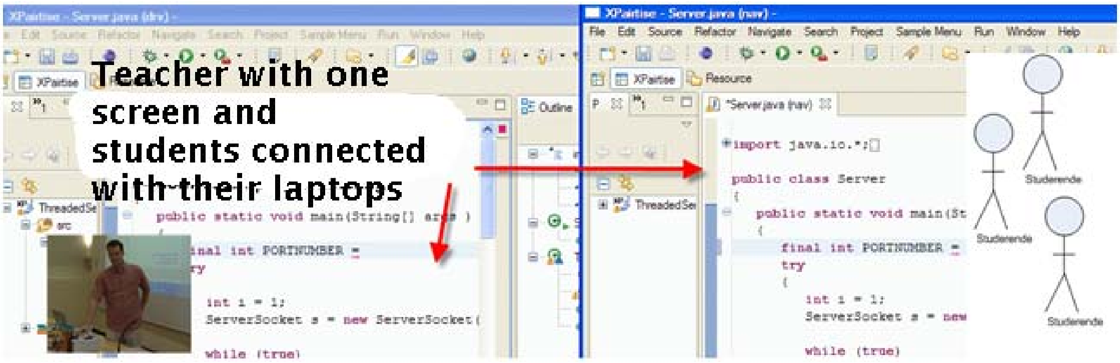


Figure 5 Live programming idea: Programming in front of the students in the classroom

The idea of live programming is that the teacher is switching from the static and boring Power Points to sit at the presenter desk and perform a small program task live. This is comparable with the little TV program hit with the melody: It is our impression that this would be a productive complement to the theoretical analysis of a given programming problem. By seeing how the teacher clicks with the mouse, the students will learn how a given integrated development environment works from the menus and similar things.

One last interesting variation is to combine team-programming this idea of Live programming. It works such that the teacher starts his editor, and make progress with the program while the projector shows it all. Then the teacher stops at certain points and asks a student to connect to the editor and make a complete line or write something else in the program using his own laptop. Our feeling with this is that it will provide an extremely high attention in class, since all students can bid into the live programming session.



```
import java.io.*;

public class Server
{
    public static void main(St
    {
        final int PORTNUMBER =
        try
        {
            int i = 1;
            ServerSocket s = new

            while (true)
```

Figure 6 Here are the variation with Live programming with active participation of the students.

These things are new, and it would be beneficial with the a discussion in the IS educational community to address these new opportunities.

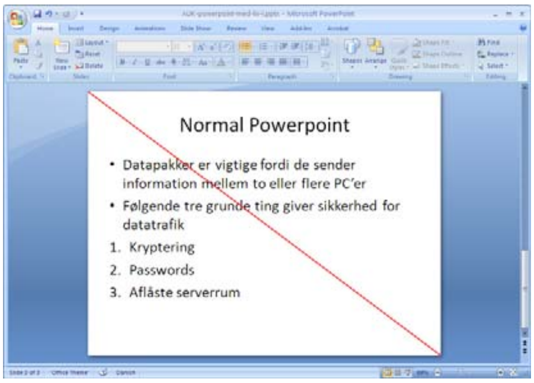
New: PowerPoint with Life

Most college classes would want that even more students showed up. The matter is that some students choose to spend time working on the side. We, as teachers, do not make the case better as teachers by uploading Power Points to Sitescape (or or university’s learning system). We can actually reward the students who come to class by giving them information not found on the delivered Power Points. Here are two things we found:

1. Deliberately have "fill in the blank" in our Power Points

2. Draw text and figures on empty Power Points with a Tablet PC or digitizer

The first idea is to save some of the important points by having blanks in our slides. It is something you can ask the students about, and then possibly even provide the answers. That way you do not serve the entire syllabus in the slides. Here is an illustration of a "before" and "after" slide.

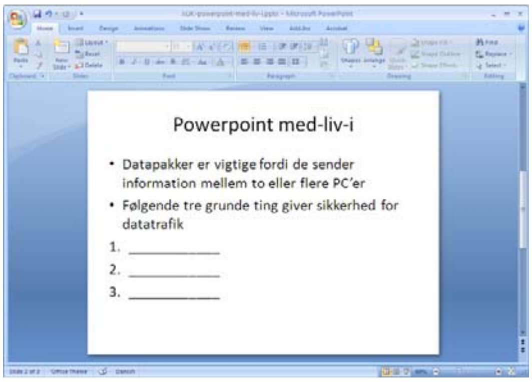


Normal Powerpoint

- Datapakker er vigtige fordi de sender information mellem to eller flere PC'er
- Følgende tre grunde ting giver sikkerhed for datatrafik

1. Kryptering
2. Passwords
3. Aflåste serverrum

Figure 7 Normal: Slide with all the information served



Powerpoint med-liv-i

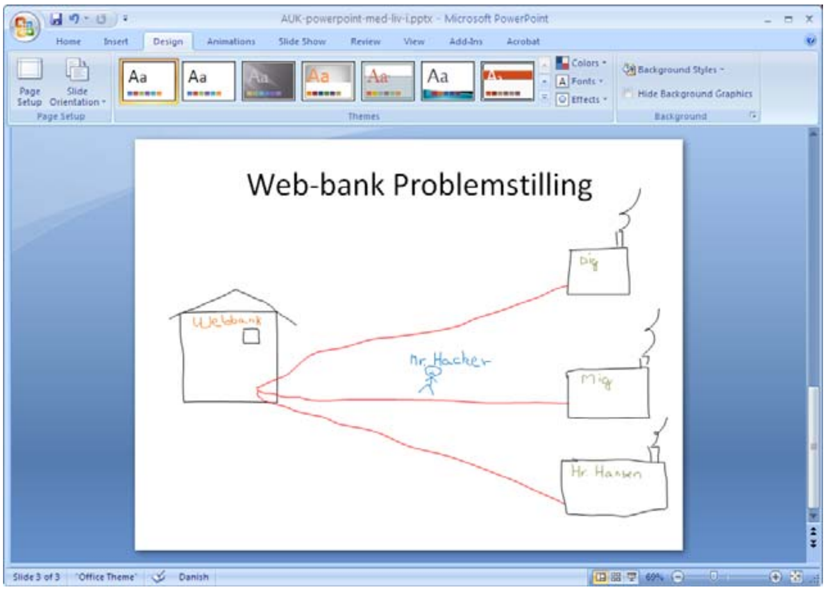
- Datapakker er vigtige fordi de sender information mellem to eller flere PC'er
- Følgende tre grunde ting giver sikkerhed for datatrafik

1. ___________
2. ___________
3. ___________

Figure 8 New: Hold back a little (information) and create a dialogue

Above is the first idea to hold back some information for an in-class dialogue.

The last idea is that you write on a blank Power Point in the same way as you would have used the blackboard. The new here is that you can then save the presentation after the event and the students can get it and thus have the drawing with their subsequent repetition of the syllabus.



Web-bank Problemstilling

Figure 9 The idea around to make drawings of his PowerPoints hours while in class. This drawing is done with a mouse, but can also be done (better) with a tablet or digitizer.

Many students think it is nice when the teacher himself doing something tailored material for hours. By using this idea, you can both meet the target, and simultaneously break the rhythm of monotone slide presentations.
We would think that in a normal 30 Power Point slides show for 90 minutes, there should be 2-5 "Power Points with-life".

NEW: HOMEWORK AND EXERCISES “WITH MEMORY”

A distributed systems teacher does not generally follow up on individual student exercises. There is not time as it is. We simply do not have the means see/measure how far people have come during practice hours. By using a particular system which in reality is a common disk drive for programmers, we can better keep track of how far the individual student is.
The idea is that the student submits the programming files on an ongoing basis in its own directory on the file server, and so the teacher could relatively easily see how far people are and possibly where there are problems.
The right way to do this in programming classes is called source code version control and it is the same as a common disk drive. The system which can be used is called Subversion. We installed it on our machine and we show both the management screen and the web interface which can see by the users.

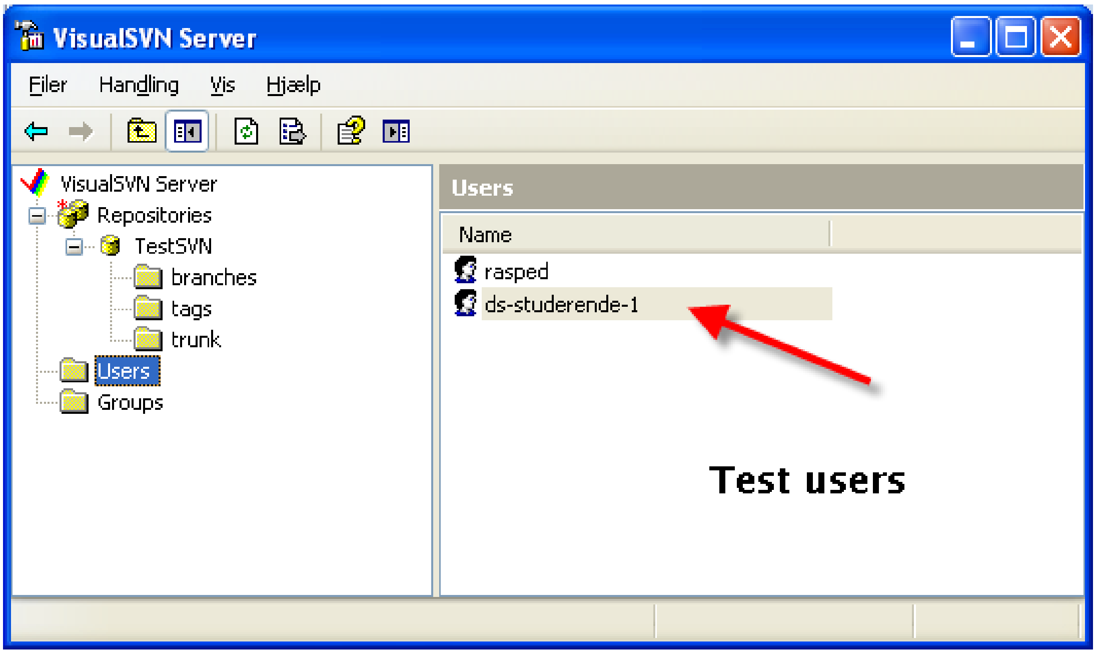
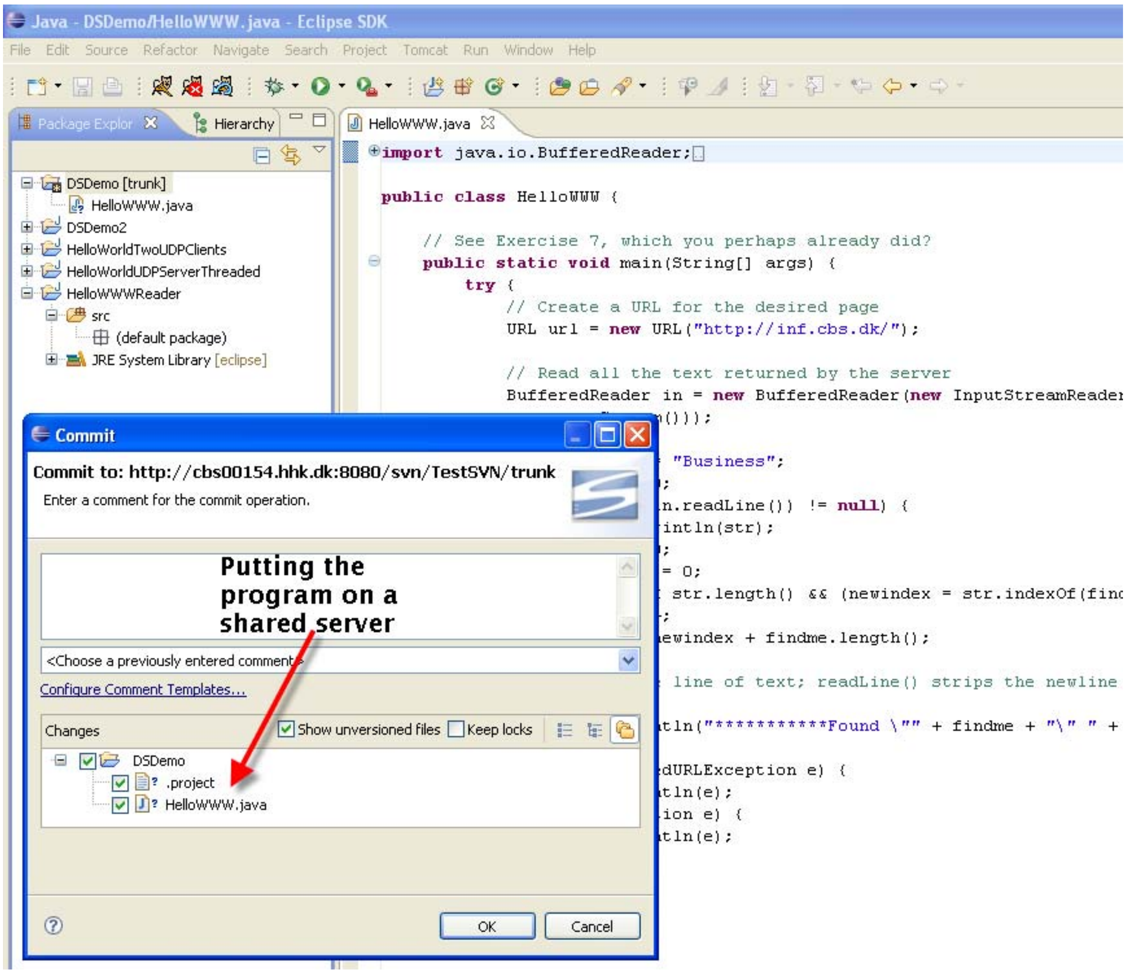


Figure 10 Configuring the shared server for homework and exercises.


Java - DSDemo/HelloWWW.java - Eclipse SDK
File Edit Source Refactor Navigate Search Project Tomcat Run Window Help
Package Explor
Hierarchy
HelloWWW.java
DSDemo [trunk]
HelloWWW.java
DSDemo2
HelloWorldTwoUDPClients
HelloWorldUDPServerThreaded
HelloWWWReader
src
(default package)
JRE System Library [eclipse]
Commit

```
import java.io.BufferedReader;

public class HelloWWW {

    // See Exercise 7, which you perhaps already did?
    public static void main(String[] args) {
        try {
            // Create a URL for the desired page
            URL url = new URL("http://inf.cbs.dk/");

            // Read all the text returned by the server
            BufferedReader in = new BufferedReader(new InputStreamReader
                              n()));

                                "Business";
                              ;
                              n.readLine()) != null) {
                              intln(str);
                              ;
                              = 0;
                              str.length() && (newindex = str.indexOf(finc
                              ;
                              ewindex + findme.length();

                              line of text; readLine() strips the newline

                              tln("***********Found \"" + findme + "\" " +

                              dURLException e) {
                              tln(e);
                              ion e) {
                              tln(e);
```

Commit to: http://cbs00154.hhk.dk:8080/svn/TestSVN/trunk

Enter a comment for the commit operation.

Putting the program on a shared server

<Choose a previously entered comment>

Configure Comment Templates...

Changes ☑ Show unversioned files ☐ Keep locks

- ☑ DSDemo
  - ☑ .project
  - ☑ HelloWWW.java

OK Cancel

Figure 11 Illustration of a students' use of the server from the programming system, we normally spend on information technology (Eclipse).

Above we have shown how it is possible to have a system which each student can use. It will also give a very clear (for both teacher and students) indication of how many are "in". So it is a teaching tool with more options than just those you think about initially.

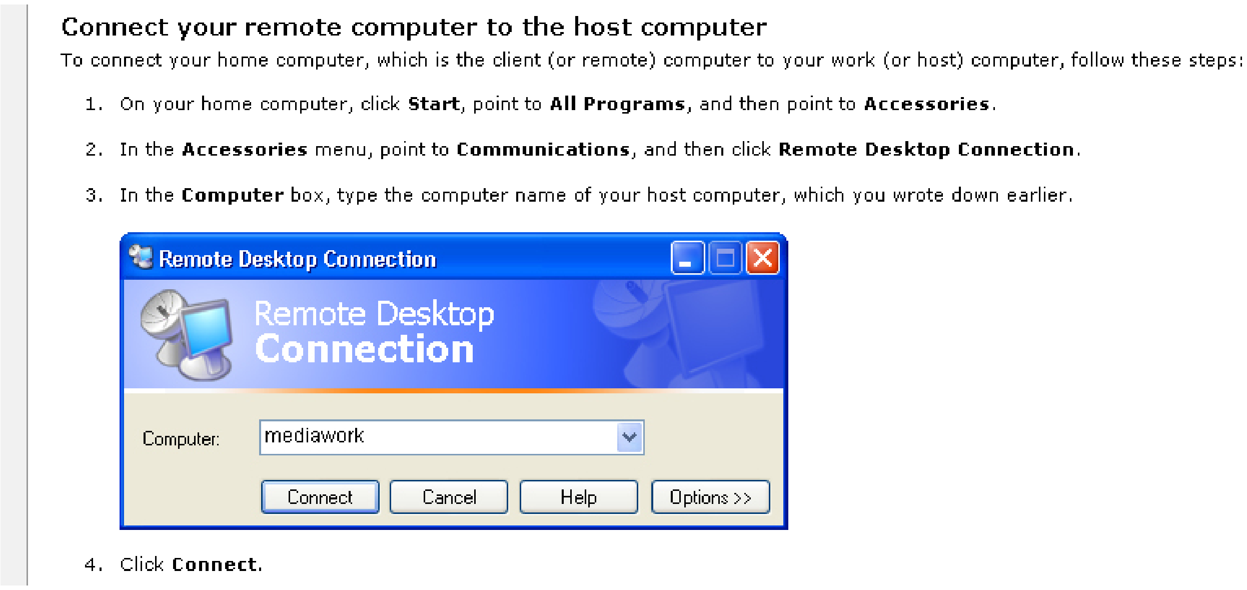
SCHEDULE

If we show the students a schedule of what to be learnt that day, and the points to be understood, and even the teaching tools to be use; they could get an overview of what will happen the next 2 x 45 min. We show a short example here:

Learning	Pointer	Didactic tools
Discuss the data packet	Encryption is useful Encryption is expensive	(1) Live Programming: Simple encryption (2) Sitescape-buzz: How can we ensure traffic (3) Exercise: Make your own program on the shared server
E.T.C.		

NEW: VIRTUAL LAB

The challenge in programming classes is that we are increasingly seeing is that students use Macintosh computers, and sometimes Linux. It is difficult for a teacher to master multiple operating systems, and even Windows Vista has teased us a few times.
So the idea to address this is that we in the DS class could use something called remote PC, which means that students (both Windows. Linux and Mac) gets a small piece of software they install. Then they can then make a visual remote access to a machine in our data-lab. The advantage is that the teachers can install all the software once on a single server, and then we know 99% that it works every time when they all see exactly the same system. This also addresses what we pointed out in the first section of the students increasingly expect that things will work the first time.



Connect your remote computer to the host computer

To connect your home computer, which is the client (or remote) computer to your work (or host) computer, follow these steps:

1. On your home computer, click **Start**, point to **All Programs**, and then point to **Accessories**.
2. In the **Accessories** menu, point to **Communications**, and then click **Remote Desktop Connection**.
3. In the **Computer** box, type the computer name of your host computer, which you wrote down earlier.

Computer: mediawork

4. Click **Connect**.

Figure 12 Here it is shown how this would work with Microsoft Remote Desktop.
A side benefit is that more teachers can coordinate this and possibly. use each other's programs.

VIII. CONCLUSION

We have described how our experience has been with teaching a difficult technical class at an interdisciplinary IS program. We addressed several teaching methods that we have already been using. Then we went on to discuss several new methods which can enhance both the teaching experience and the programming experience. Our contribution is that we using standard and existing tools have identified new processes that can be used by other teachers facing the same challenges.