PREPRINT - Proceedings of the IEEE International Education Engineering Conference (EDUCON 2013), p. 190-199, Berlin (Germany), March 13-15 2013. ISBN: 978-1-4673-6109-5. https://doi.org/10.1109/EduCon.2013.6530105

Training Sessions in a Master Degree “Informatics as a Second Competence”

Agathe Merceron
Media Informatics department
Beuth University of Applied Sciences
Berlin, Germany

Sergio Luján-Mora
Department of Software and Computing Systems
University of Alicante
Alicante, Spain

Jean-Michel Adam, Daniel Bardou, Jerome David
Laboratory of Informatics of Grenoble
Pierre Mendes France University
Grenoble, France

Marek Milosz
Institute of Computer Science
Lublin University of Technology
Lublin, Poland

***Abstract*—The ERAMIS acronym stands for *European-Russian-Central Asian Network of Master's degrees "Informatics as a Second Competence"*. The aim of this project is to create a Master degree “Computer Science as a Second Competence” in 9 beneficiary universities located in Kazakhstan, Kirghizstan and Russia. One crucial aspect of the project is training. In this contribution we describe how training sessions have been organized, the pedagogical issues that are involved and ways to address them.**

Master degree; Computer as a Second Competence; teaching experience;

I. Introduction

Our world does not work anymore without Information Technologies. Almost every profession needs computers with adequate software. The production of this adequate software requires that the software developers understand the needs of the software users. Most of the time this understanding necessitates a person that understands both worlds: the world of the users and the world of the developers. This need has been recognized as early as 1987 at the University Pierre Mendès France in Grenoble, France where a Master “Informatics as a Second Competence” has been created. This degree is intended for students having a Bachelor in social sciences and therefore who are non Information Technology (IT) specialists. The Master's program objective is strictly professional, as it aims to train versatile people who will combine good skills (acquired in a Bachelor’s degree) in the first discipline with theoretical and technical skills in computer science, enabling them to create, develop and implement tomorrow's software tools in a better way.

Following the model of the Master of the University Pierre Mendès France, a similar Master has been created in the Kyrgyz National University, Bishkek, Kirghistan [1]. Presently this model is further extended in the EU Tempus project ERAMIS [2]. The ERAMIS acronym stands for ***European-Russian-Central Asian Network of Master's degrees "Informatics as a Second Competence"***. This 3,5-year project is supported by the European Union (EU) in the Tempus scheme and involves five universities from five partner countries: Finland, France, Germany, Poland and Spain. It started in January 2010. Its aim is to create a Master degree “Computer Science as a Second Competence” in 9 beneficiary universities located in Kazakhstan, Kirghizstan and Russia. This degree is intended for students having a Bachelor in social sciences, exact sciences or engineering but who did not graduate in computer science. This Master is thus a geographical and disciplinary extension of the two already existing Masters.

The working-methodology adopted in the ERAMIS project is described in [2]. One crucial aspect of the project is training. [3] describes the faculty training put in place inside the project. The faculty training is completed by training sessions taking place in the universities of the beneficiary countries. In this contribution we describe how these training sessions in the beneficiary universities are organized and the pedagogical issues that are involved. Further we share ways to address some of the pedagogical issues and show how they have been put into practice with concrete examples of training sessions.

II. Organisation and Pedagogical Issues

A. Format of a Training Session

A training session is a course of 10 to 20 hours that takes place over 4 to 5 days. It is intended in the first place for students of the Master. The number of participants should be from 10 to 20. If place allows academic staff or other interested students form the beneficiary university can attend. A training session is taught by a member of a European university member of the project. The teaching language is mainly English, as very few persons form the European universities can speak Russian. Furthermore the ERAMIS project foresees that 20% of the teaching in the Master has to be done in English. Holding training sessions in English helps to meet this goal.

B. Putting up the Offer

Partly supported by the EU Tempus Project ERAMIS 159025-TEMPUS-1-2009-1-FR-TEMPUS-JPCR

The European Universities gave a list of training sessions they could offer according to the curriculum of the Master and to their respective strengths. Each beneficiary University expressed its wishes for the suggested sessions. From those wishes we could determine the sessions which would be organized and the universities which would host them. Sessions were distributed in a fair way among universities, depending on the competences of each of them and are shown in Table I.

C. *Pedagogical Issues*

They are several pedagogical issues that make teaching a training session different from the usual teaching in one’s home university.

- The format of the training session means that one has to teach to the same students several hours a day and 4-5 days in a row. Students do not have one week to absorb the content of a day.
- Students have already a Bachelor degree, which means they already have a good level of maturity and motivation. However the topic of the training session is a topic in computer science at a Bachelor level.
- The exact level of the students (and also of the academic staff) in the topic taught is somewhat unknown before the training session begins.
- The level of English of the students may vary greatly and may not be very high. English is also a second language for all trainers.

TABLE I. PROGRAM OF THE TRAINING SESSIONS

University	**Topic**	**Nr. hours**
Astrakhan State University	Design patterns in Java	10
	Frameworks in Java	10
	Spatio-Temporal Representation and Reasoning	10
	Practice of the Semantic Web	10
University Al -Farabi Almaty	Design patterns in Java	10
	Frameworks in Java	10
Voronezh State University	Web Programming with PHP	15
	Algorithms and data structures	10
	Wireless Sensor Networks	10
Kazan state University	Database Design	10
	Programming in T-SQL on MS SQL Server 2008	20
	Data Mining and Data Warehousing	20
Kyrgyz Natonal University Bishkek	Model Driven Engineering and WebML	10
	WebML in practice using the WebRatio	10
Kyrgyz State Technical University Bishkek	Web Application Programming with Google Maps	15
	Introduction to XML information system architecture	16
Karaganda State University	Model Driven Engineering and WebML	10
	WebML in practice using the WebRatio	10
	Introduction Router networks	10
Pavlodar Innovate University	Unified Modeling Language and C++	20
	Wireless Sensor Networks	10
	IPv6 Low power Wireless Personal Area Networks	10
	Introduction to Web Accessability	15
Saratov University of Economy	Data Mining and Data Warehousing	20
	Introduction to MATLAB for Computer Science	12

III. WAYS OF ADRESSING THE PEDAGOGICAL ISSUES

An efficient way of addressing the pedagogical issues mentioned above is to practice *content reduction* as well as to *encourage and require both interactions and motivation* from students [4,5]. Practicing content reduction means selecting carefully not only the topics to be addressed, but also how deeply to address them and preparing a good outlook so that students get an idea of what they know and what they need to discover later in the field. Requiring and encouraging interactions and motivation means planning tasks where students have to be active and engaged, supporting them while solving the tasks and appreciating positively their activity. Content reduction makes it easier to adjust the teaching to the level and the absorption’s capacity of students: no need to rush through the material. Requesting and encouraging interactions and motivation allows the teacher to be better aware of how well students follow, as she can check how students solve exercises, and whether they have difficulties in doing so.

In the following we give concrete examples how these principles have been put into practice in several training sessions.

A. *Training Session Data Warehousing & Data Mining*

This 20-hours training session was delivered over 4 days to about 12 students. Its description is given in Table 2.

TABLE II. DESCRIPTION OF THE SESSION DATA MINING & WAREHOUSE

Description	Data Mining and Data Warehouse belong to the fact-based systems supporting Business Intelligence. This course introduces both fields with an emphasis on Data Mining.
Objectives	- To introduce the field Data Mining and Data Warehouse - To show the Data Mining Process Cycle - To introduce the main areas of Data Mining and a technique within each area - To introduce the tool RapidMiner
Content	- The Data Mining Cycle - Data, data preparation and Data Warehouses - Clustering, K-Means clustering and clustering

PREPRINT - Proceedings of the IEEE International Education Engineering Conference (EDUCON 2013), p. 190-199, Berlin (Germany), March 13-15 2013. ISBN: 978-1-4673-6109-5. https://doi.org/10.1109/EduCon.2013.6530105

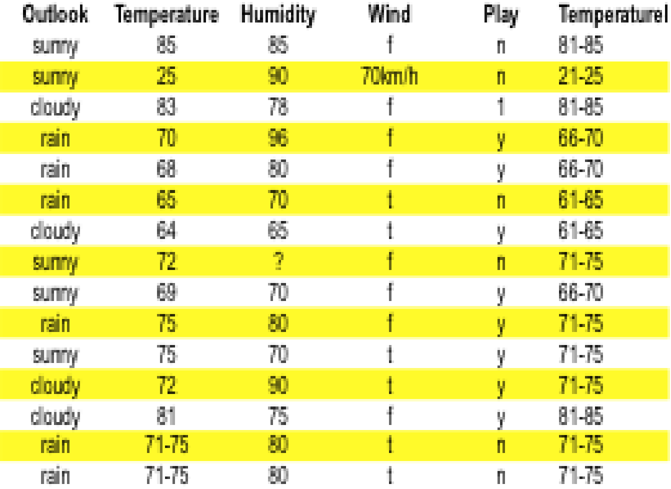
	evaluation - Classification, decision trees and evaluation - Association rules and interestingness measures - Case study with RapidMiner
Students' background	Statistics and Mathematics first year University
Software	RapidMiner
Teaching language	English, completed by translations of the present teacher because the poor level of English of some of the students
Students' material	Course slides, examples, exercises all in English

Practicing content reduction meant in particular to cover only one technique per data mining area and one way to evaluate this technique, so as to still practice the data mining cycle entirely. For clustering the k-means algorithm and the evaluation "average within centroid distance" were selected; for classification the technique decision tree and the evaluation accuracy / confusion matrix were selected; finally for association patterns, the technique association rules and the interestingness measures lift and cosine were selected for evaluation.

A number of notions have not been lectured classically, the teacher explaining and the students listening. Instead they have been introduced through exercises that requested students to be active and engaged, that made them ask questions and explain what they discover, understand or don't understand. In the sequel several examples of such activities are given.

On the first day in the first half-hour students were given the data set shown in Table III, a modification of an original data set used in textbooks like [6,7], and had to find the missing, erroneous, duplicate and obsolete data. After this exercise it was easy for them to understand data cleaning and data preprocessing. A secondary effect of that exercise, as solving it requires no deep knowledge but common sense and observation skills, is that students gain confidence and become willing to speak in English, even if they can't express themselves completely fluently and have to ask their neighbors for words.

TABLE III. DATA SET TO INTRODUCE DATA CLEANING



Outlook	Temperature	Humidity	Wind	Play	Temperaturel
sunny	85	85	f	n	81-85
sunny	25	90	70km/h	n	21-25
cloudy	83	78	f	1	81-85
rain	70	96	f	y	66-70
rain	68	80	f	y	66-70
rain	65	70	t	n	61-65
cloudy	64	65	t	y	61-65
sunny	72	?	f	n	71-75
sunny	69	70	f	y	66-70
rain	75	80	f	y	71-75
sunny	75	70	t	y	71-75
cloudy	72	90	t	y	71-75
cloudy	81	75	f	y	81-85
rain	71-75	80	t	n	71-75
rain	71-75	80	t	n	71-75

Many data mining techniques, for instance k-means clustering, require a distance to be calculated between two objects. When objects are described by attributes that are all numbers a common distance is the Euclidean distance. A characteristic of the Euclidean distance is that numbers with a great order of magnitude may impact the final calculation much more than small numbers. Students were not taught this characteristic. Instead, after making sure that they understand the formula, they were given the data shown in Table IV and had to answer the questions below it. The data in Table IV describe three persons with identity 1, 2 and 3 that are represented by two attributes: their weight in kilograms and their height in meters.

TABLE IV. DATA SET TO INVESTIGATE PROPERTIES OF EUCLIDEAN DISTANCE

Id	Weight	HeightInMeter
1	51	1.2
2	64	1.5
3	58	1.9

a. Use the Euclidean distance to calculate who is nearer to person 1: person2 or person 3. Show your calculations.

b. Create a new attribute called HeightInCm that gives the height of a person in centimetres. Use again the Euclidean distance and the attributes Weight and HeightInCm to determine who is nearer to person 1: person 2 or person 3. Show your calculations.

c. Explain the difference.

d. Give a way to calculate the distance using attributes that all have the same order of magnitude.

With this exercise students are at first puzzled because questions a and b give contradictory results though the same persons are handled, but they grasp quickly how attributes with big numbers may impact Euclidean distance. This exercise made also during lecture on the first day is convenient to introduce transformations that put all data within the same range (last question). It makes also students more talkative and curious.

K-means clustering has a number of properties and pitfalls. One property is that outliers or extreme values tend to form they own small cluster. One pitfall is that this technique always returns a set of clusters, even if the data are random and do not cluster naturally. To boost curiosity and motivation of students and to be aware of how much they follow and understand, these two aspects were not lectured classically, instead students had to discover them with exercises using the tool RapidMiner [8]. Students had to explore three data sets, look visually for some natural clustering in the data using plots, run the k-means algorithm with different values for k, the number of clusters, and evaluate each clustering with average within centroid distance, a facility provided by the tool RapidMiner. Two data sets are given in Table V and represent points with their coordinates. The third dataset has to be constructed by students taking the original data of the set shown in Table I and selecting only the two quantitative attributes Temperature and Humidity.

TABLE V. TWO DATA SET TO INVESTIGATE PROPERTIES OF K-MEANS

x	y
2	10
2	5
8	4
5	8
7	5
6	4
1	2
4	9

x	y
2	10
2	5
8	4
5	8
7	5
6	4
1	2
20	-3

The first data set has 3 natural clusters. The second data set has one outlier, the last element. The data of the last data set do not cluster naturally. A scatter plot shows them more or less randomly distributed in the graph, see Figure 1.

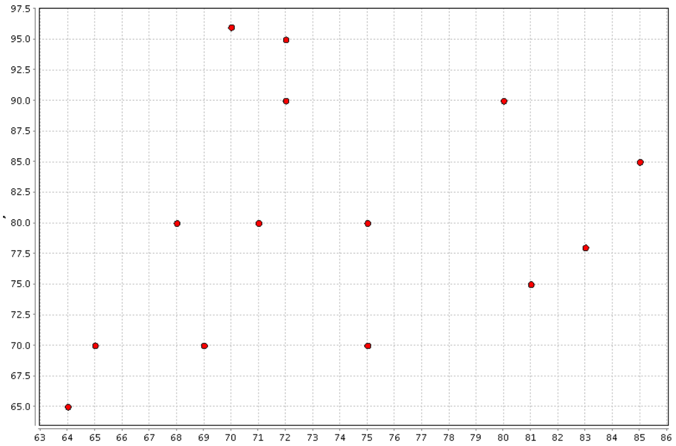


Figure 1. Scatter Plot of the dataset showing no obvious natural clustering.

Students observe that the outlier of the second data set forms a cluster by itself, whatever the value of k is, as illustrated in Figure 2 for k=2. They also observe the sharp drop of the average within centroid distance when k=3 in the first data set, which indicates that 3 is a good number of clusters, with k=4 for the second data set. However no such sharp drop can be observed for the last data set pictured in Figure 1, which indicates that a clustering pattern might not be discovered. This exercise makes students understand the importance of evaluating a pattern much more efficiently than any discourse and motivates them further.

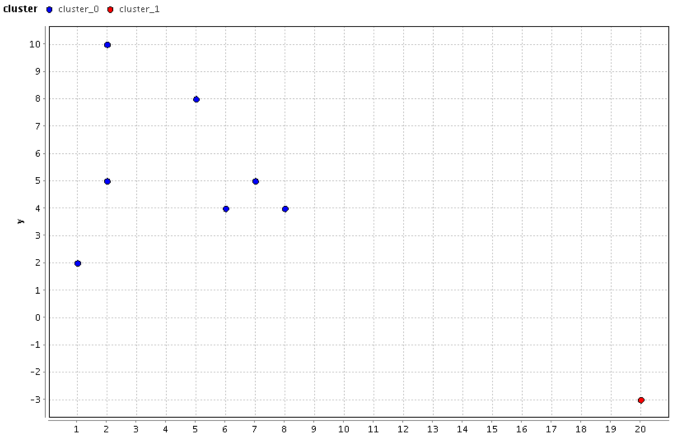


Figure 2. Scatter Plot of the dataset Table V right showing the outlier in a cluster by itself.

A final example of requesting students' interaction and encouraging motivation is taken from the association rules area. The association rules technique aims at discovering rules like "if costumers buy beer and coke, then they also buy diapers". Such a rule gives a dependency between consequent, here "buy diapers" and the antecedent, here "buy beer and coke" and is often denoted by the short form "beer, coke → diapers" [6,7]. In the same way as average within centroid distance helps evaluate the goodness of a clustering, lift and cosine are two measures that help evaluating the interest of an association rule found by the algorithm. As these two measures might give contradictory results, it is quite important to understand both their intuitive meaning and their different properties. One property of cosine that lift does not have is the so called null-invariant property [6,7], that will be explained shortly. Students had to discover it by themselves calculating cosine and lift of the rule X → Y for the three data sets shown in Table IV. Tid is the transaction identity of some customer who has bought items X, Y or Z. The first data sets contains only 5 transactions, the second data set 7 and the last one 10. The five first customers in the 3 data sets have all bought the same items. Customers 6 to 10 buy at most item Z. Calculations show that cosine (X → Y) has the same value for the 3 data sets while lift (X → Y) increases: the smallest value is for dataset 1 and the highest value for dataset 3. Lift is sensitive to transactions containing neither X nor Y, while cosine is not. Cosine stays invariant when the data set is augmented with null-transactions with respect to the rule. This result surprises students and makes them reflect deeper on the intuitive meaning of these two measures.

TABLE VI. DATA SETS TO INVESTIGATE THE INTERESTING MEASURES LIFT AND COSINE

TId	Set 1	Set 2	Set 3
1	X, Y, Z	X, Y, Z	X, Y, Z
2	X, Y	X, Y	X, Y
3	X, Y, Z	X, Y, Z	X, Y, Z
4	X, Y	X, Y	X, Y
5	X, Z	X, Z	X, Z
6		Z	Z
7		Z	Z
8			Z
9			Z
10			Z
11			Z

B. *Training Sessions Database Design and Programming in T-SQL*

The 10-hours training session Database Design lasted for 1.5 days, followed by 20-hours training session Programming in T-SQL on MS SQL Server 2008 – next 3 days. Both sessions were bound together and included the design and development of database queries in a particular Relational Database Management System (RDMS) – Microsoft SQL Server 2008. Their description is given in Table VII and Table VIII. The Database Design training session has been planned as the realization of a typical approach to design a Database (DB): from requirements to physical data model (in Relational Schema) and SQL Data Definition Language (DDL) commands to run on the DB server.

TABLE VII. DESCRIPTION OF THE SESSION DATABASE DESIGN

Description	Designing of the database structure is one of phases of application developing process. The way from user requirements, formalized in different methods, to database logical and physical structure will be presented. Entity Relationship technique and transformation Logical Data Model to the Relational Scheme (with the SQL code generation) will be described and practical trained using a CASE tool.
Objectives	- To show students the fundamentals of two phases of Data Modelling using application requirements. - To show students concepts of Logical and Physical Data Models. - To show students the methodology of Entity Relationships modelling and transformation model to Relational Schema. - To show students the methodology of database developing using special CASE tool.
Content	- Application design – general idea From requirements to database structure - Entity Relationships technique – Logical Data Model - Relational Schema – Physical Data Model Model transformations - SQL Code generation - Some challenges and practical solutions of Database Design - Practice and Labs (CASE Tool)
Students' background	- Knowledge of software engineering processes - Basic knowledge of relational database techniques
Software	Toad Data Modeler
Teaching language	Russian
Students' materials	Documentation (slides), examples, and exercises in English

Classes in Database Design training was conducted as activating students' seminars, which consisted of the following activities:

- lectures (about 20% of the time),
- joint group exercises presenting techniques and CASE tool (30%),
- independent solving of case studies by students (40%),
- discussion and evaluation of the results of independent practice (10%).

Lectures were a theoretical introduction to DB design. They presented a position of design in an application life-cycle. In particular they presented typical techniques of defining requirements, such as user goals analyzing (Figure 3), events analyzing, CRUD (Create, Read, Update, Delete) – Figure 4, documents analyzing, things definition.

User	User goal
Student	Look up the lessons schedule Look up the exam results Check the news
Professor	Look up the lessons schedule Put the exam results Put the news
Dean	Put the lessons schedule Put the news

Figure 3. An example of user goal analysis.

Data (things/object)	C	R	U	D
Lessons schedule	Dean	Student Professor	Dean	(Dean or Admin?)
Exam results	Professor	Student	Professor	(Admin or Professor?)
News	Professor Dean	Student	Professor Dean	Time

Figure 4. An example of CRUD technique

During the lectures a typical approach to the modeling of DB has been introduced, see Figure X, including the division into two types of models: Logical Data Model (LDM) and Physical Data Model (PDM). Principles, notation, and construction rules of Entity Relationships Diagram (ERD) as a model of LDM have also been described. Also the modeling problem of the inheritance in LDM was featured.

General ideas of relational schema (tables, keys, data integrity rules, indexes, views etc.) and problems of Entity Relationship Diagram transformation to Physical Model were presented during the lessons in details.

During the practical part of the training, students became familiar with one of the CASE tool used for DB modeling (Toad Data Modeler) and process of DB designing. Students designed the LDM for a case study of CAFE support software and mapped it into PDM. The case study concerned the DB design for a café shop support system called CAFÉ, chosen to motivate students. Students received one-page description of processes in a cafe shop. Based on this description students defined system users, created the goal analysis, found things and, using the CRUD technique, checked the completeness and correctness of their results.

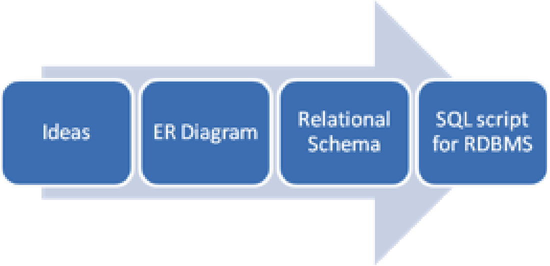


Figure 5. Database modeling phases.

When the DB logical structure was ready, students prepared and performed presentations of the design results (models). After the presentations a brainstorming session was held. This session purposes were the formal verification of the models and the checking of the end-user fulfillment of software requirements. During the brainstorming session students developed the final LDM (Figure 6). Some student's teams rebuilt their solutions. Modifications were made using the CASE tool. Efficiency of using the CASE tools for designing the DB was highlighted thanks to this exercise.

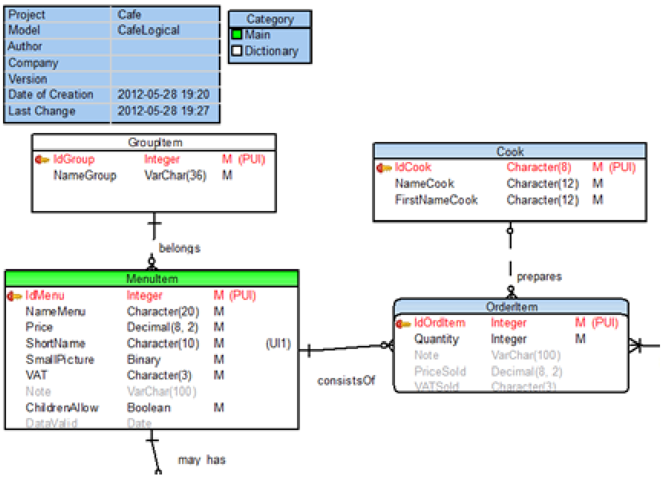


Figure 6. Logical Data Model for CAFE database (a part)

After the definition of the target environment and some minor modification of mapping tables, students generated the PDM for MS SQL Server 2008 using the CASE tool functionality.

As a final results of DB process designing, students generated the code in DDL using the CASE tool as well and compared it with the PDM diagram. The code was used to generate the database during the second phase of training.

The training *Programming in T-SQL on MS SQL Server 2008* was designed as a programming course in Transact-SQL language and covered selected groups of commands from DDL, DML (Data Manipulation Language) and QL (Query Language) – the SELECT command. The detailed description of training is presented in the table VIII.

TABLE VIII. DESCRIPTION OF THE SESSION PROGRAMMING WITH T-SQL

Description	The presentation of fundamentals of programming in Transact-SQL in the MS SQL Server 2008 environment. There will be presented basic and advanced commands and programming structures that let to explore data in the database and process them. All topics will be described in detail using the examples.
Objectives	- Getting knowledge and skills how to create and modify data structures in database - Getting knowledge and skills how to improve data exploration - Getting knowledge and skills how to manipulate data i.e. how to insert data into the database, how to update and delete data - Getting knowledge and skills how to explore data in the database - Getting knowledge and skills how to process data using cursors
Content	- Creating data structures i.e. tables and relationships between them on the base of exemplary Entity Relationships Diagram (ERD) - Creating indexes to improve data exploration

	- Data import into the tables - Data exploration using basic form of SELECT command (ordering data during displaying, removing duplicates from the result set, data filtering, joining the tables) - Data exploration using advanced option of SELECT command (data grouping, aggregate functions, subqueries) - Data processing with the use of cursors (declaration of the cursor, opening the cursor, fetching data from the cursor, closing the cursor)
Students' background	Basic knowledge about relational databases and its design techniques (ERD)
Software	MS SQL Server 2008
Teaching language	Russian
Students' materials	Documentation (slides), examples, and exercises in English

C. Training Session Web Programming with PHP

This 15-hour training session was held during 5 days in sessions of 3 hours in the Faculty of Computer Science of the Voronezh State University for about 12 students. Its description is given in Table IX.

TABLE IX. DESCRIPTION OF THE SESSION WEB PROGRAMMING WITH PHP

Description	PHP is a general-purpose server-side scripting language originally designed for web development to produce dynamic web pages. For this purpose, PHP code is embedded into the HTML source document and interpreted by a web server with a PHP processor module, which generates the web page document. . This is an introductory course for students without previous knowledge on PHP, but students must have previous knowledge on programming in general and specifically on C, C++, or Java.
Objectives	- To show students the fundamentals of web applications - To show students how to develop web applications based on PHP - To show students some tools that can help them to work with PHP
Content	- Basics of web applications - Installation, configuration, and use of the programming platform - Analysis and design of the web project - PHP's Syntax: variables, arrays, statements, etc. - Object-oriented programming with PHP - Managing web forms with PHP - Accessing a database with PHP - Implementation of the web project
Students' background	- Programming in general - Object oriented programming - Some knowledge about HTML and CSS
Software	- Text editor (e.g., Notepad++) - Web development platform (e.g., XAMPP) - Web browsers (e.g., Internet Explorer 9, Mozilla Firefox 3) and Internet connection
Teaching language	Mainly English, but sometimes one of the students had to help the teacher's explanations, because the level of English of some of the students was not very good
Students' material	Documentation (slides), examples, exercises, and website in English

This training session was designed to serve as a first course in undergraduate web application programming in computer science curriculum. The prevailing atmosphere throughout the course was very nice and all students participated actively, asking questions and completing the exercises proposed. In addition, all students made the seminar project that was proposed.

Figure 7 shows a few students doing the exercises. In the background of this picture, the image of the teacher's presentation can be seen, since a video projector was used to display the teacher's presentation during the seminar. As it can be seen in this figure, each student had their own computer, because the seminar was mainly practical and students had to do many exercises and the seminar project.

Besides the slides, a public website was prepared to support the training seminar, see Figure 8. This website was updated in real time during the teacher's explanations to reflect the questions and doubts that students expressed during lectures, and thus encouraged students interactions. In this sense, the website was and is a true reflection of the training seminar that was given. The website was structured in four main parts:

- Course: the whole seminar, with lessons, exercises, and project explanations.
- Lessons: the lectures of the seminar.
- Exercises: the exercises proposed to do during the seminar.
- Project: the project explanations and documents needed to do the project.

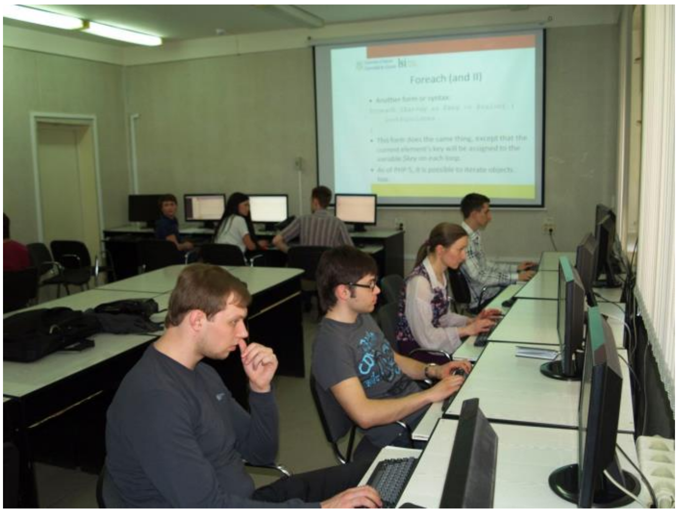


Figure 7. Students doing exercises during the training.

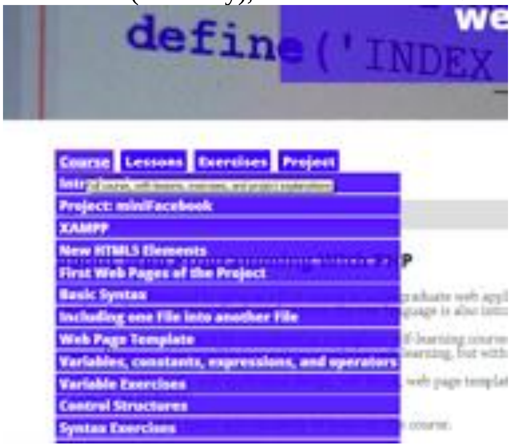


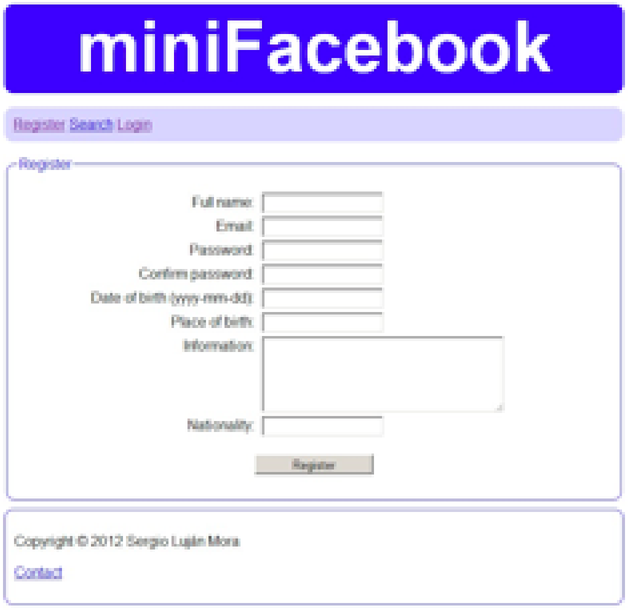
Figure 8. Website used during the training.

During the training seminar, besides the proposed exercises, students had also to work on a personal project. The project was a web application inspired by Facebook: students had to develop miniFacebook, a very simple version of this well-known social network and a topic particularly motivating for students. The aim of this project wasn't to create a perfect and complete social network, but a web application with the basic elements of a real social network.

The main features of the project were:

- To create a miniFacebook application with registered users through email and password.
- The application is divided into two connected but separated parts: the public area and the private area.
- In the public area, the list of users can be seen, but users' profiles or their posts can't be seen.
- To view the users' profiles and posts, a visitor must have an account and must be logged into the application.
- In the private area, a user can write and post messages, can modify his profile, and can delete his account.

Finally, in Figure 9, the web page to register as a new user of the application is shown. When registering into the application, the following information should be entered: full name, email, password, confirm password, date of birth, place of birth, information (e.g, description of the user), and nationality. This information is stored in a MySQL database that students had to create and manage. The web application phpMyAdmin was used as MySQL client.



Copyright © 2012 Sergio Luján Mora

Figure 9. Webpage to register a new user.

D. Training Sessions Design Patterns in Java and Frameworks in Java

Both these two 10-hours training sessions were delivered over 5 days in Astrakhan, and then over 4 days again in Almaty. In Astrakhan the training had to take place 4 hours a day late in the evening because the attendees were busy during the rest of the time, whereas in Almaty more standard sessions were organized. The description of these sessions is presented in table X.

TABLE X. DESCRIPTION OF THE DESIGN PATTERNS IN JAVA AND FRAMEWROKS IN JAVA SESSIONS

Description	Design patterns and frameworks are nowadays intensively used in object-oriented software engineering. The former of these two courses introduces design patterns and how they can be used in Java while the latter introduces the use of patterns in common Java frameworks.
Objectives	- To introduce the concept of pattern - To introduce some well known design patterns - To introduce the concept of framework - To introduce MVC (- To introduce Java DAO persistence API, initially planned, not retained)
Content	- Presentation of the pattern concept - Overview of the 23 patterns of the "Gang of Four" - Case study: implementing support for a lexicographic tree in Java and improve it while applying 3 design patterns. - Presentation of the framework concept - Presentation of MVC and its underlying patterns - Introduction to Swing widgets - Case study: implementing a graphical user interface (GUI) for a bounded integer in Java/Swing
Students'	- Object-oriented programming (Java prefered)

background	- Some knowledge of UML
Software	Java Development Kit, Eclipse IDE
Teaching language	English, completed by translations of the present teacher because the poor level of English of some of the students
Students' material	Course slides and exercises all in English

A full lecture on (Object-Oriented) Design Patterns may encompass software design, object-oriented basic concepts (encapsulation, state, behavior, inheritance, …), design and programming techniques such as object delegation, classes and packages decoupling, building inheritance hierarchies, but also a full explanation of the concept of pattern, the way it can be described and the relations that one can put on them, pattern systems and languages.

Content reduction on this training session was handled this way: a brief history on patterns as an introduction, the presentation of the 23 famous design patterns of the "Gang of Four" (GoF) [9], and practical advices on how and when to use design patterns and a quick glance at the anti-pattern concept as a conclusion. In the main part of the lectures presenting the GoF, a very simple describing format was used pointing out the name of the pattern, the problem it is addressing and the principles of the proposed solution, its structure depicted in Unified Modeling Language and a brief discussion of the pros and cons of applying the pattern. Only 10 of the GoF patterns were presented with this low level of details (3 creational patterns, 3 structural and 4 behavioral ones), the 13 remaining patterns have only be cited for further reading while their principles were briefly compared to the 10 former patterns. After each pattern presentation, known uses of the pattern in Java predefined classes was also discussed to illustrate the importance of patterns in modern programming. Considering content reduction the lecture could have focused on a smaller number of design patterns but it was wanted to give the students an overview of the diversity and coverage of a whole patterns family.

For encouraging student interactions and motivation, the lecture was highly backed by a single case study considered from the beginning of the session until the end of it: programming a lexicographic tree (see Figure 10). Instead of simply designing the classes, the students were asked to program so that a very practical approach such as the one of [10] is adopted.

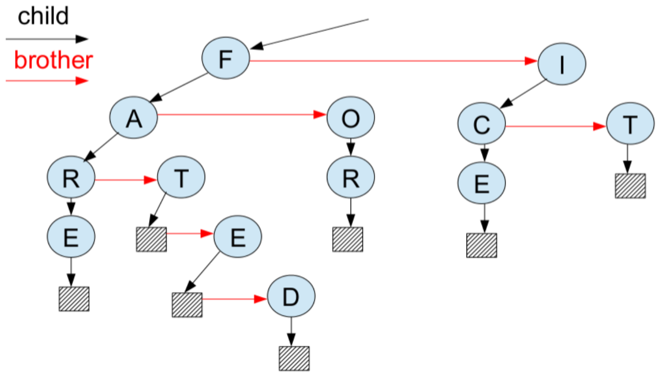


Figure 10. Example of a lexicographic tree containing 7 words.

The exercise started with the programming of 3 classes in a small inheritance hierarchy (AbstractNode, Node and Mark, see Figure 11), initially considering that only nodes are sufficient to implement a tree.

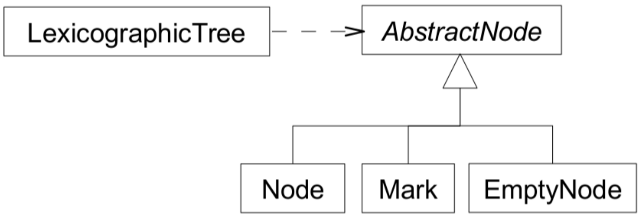
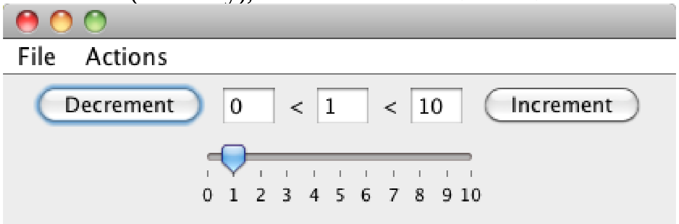


Figure 11. Java classes involved in the lexicographic tree project.

The difficulty to deal with an empty tree (*i.e.* a null reference in an AbstractNode variable) on the one hand, and the difficulty to encompass a tree as a whole on the other hand are then discussed with the students. This discussion leads to apply the Adapter pattern and create a new class LexicographicTree to improve the project. The Null-Object pattern (as described in [11] for example) is then presented and applied on the project to get rid of lots of null-reference checking in the programs, resulting in the definition of a new class EmptyNode in the node classes hierarchy. Finally the Singleton pattern is applied as an empty node being a factice node, there is no need to have several instances of EmptyNode. After applying each of these three patterns to the project, some time is taken with the students to discuss the improvements of the code and recall the role of each class in each applied patterns.

The goal of the Frameworks training session was to show students how patterns are used in practice in common application development frameworks. The initial objective was to teach two different frameworks: MVC (Model View Controller) [12] illustrated on Java Swing and DAO (Data Access Object) illustrated on Java Persistence API. Since students were not familiar enough with Java technology and development, content reduction resulted on keeping only the MVC part.

The lecture was built along a single running example consisting of a model for a bounded integer. Students have been asked to refine and develop this example as concepts were introduced. After an overview of the MVC paradigm, the focus was step by step put on each part of MVC in order to build a view for the bounded integer model. At first, the model has been extended in order to be "Observable" (use of the Observer/Observable pattern), some extension of this pattern using delegation pattern has also been shown. After a brief presentation of Java Swing widgets, students were asked to build their own Graphical User Interface (GUI) for the given model. Finally the model was connected to the view by allowing the model to notify its changes to the view and by developing controllers in charge of updating the model following notifications of GUI widgets. At the end of the training, students obtained a fully functional bounded integer application built on their own (see Figure 12).



0 < 1 < 10

Figure 12. Example of GUI for the bounded integer application.

The last two hours of the Frameworks training have been dedicated to establish a link between the two training sessions. Students have been asked to build a MVC application for the lexicographic tree they developed in the Design Patterns training session (see Figure 13 for an example).

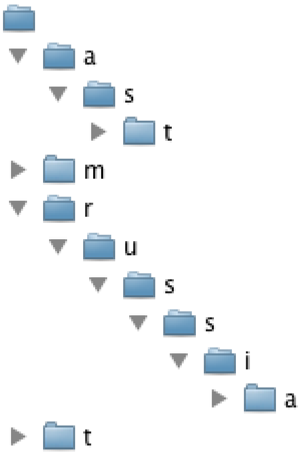


Figure 13. Part of the GUI for the lexicographic tree achieved by the students.

It was firstly planned to have the students attend the Design Patterns session before the Frameworks one as Frameworks can be thought as Patterns of higher level. However the two training sessions were mixed together afterwards as it was more comfortable for the students to vary their activities but it also helped the weakest students to gain more confidence in their Java skills prior to encountering the real difficulties of each training session. The heterogeneous level of students attending the sessions and the language issue was a problem for ensuring a smooth progression of the lecture and the exercises. The small number of students (10-15) attending the sessions however helped on these issues in some way as the most skilled students (either in English or in Java) spent some time to re-explain the weakest ones some points of the lecture in Russian. Using a single running example for each session was a good choice as students did not have to get familiar with several examples to progress in such a short period of time. This short period nevertheless did not benefit to the weakest students that did not end with fully working programs despite the efforts of the teachers to control the progression of the whole group.

IV. Conclusion

Training sessions for students in a Master “Informatics as a second competence” are interesting but also challenging for teachers, especially in the context of the ERAMIS project, because students and teachers come from different countries, and teaching is conducted in English, a second language for all. In this contribution we have exposed ways that have been used in the ERAMIS project to make teaching these sessions as effective as possible. Content reduction as well as encouraging and requiring both interactions and motivation are effective ways for a better teaching in any setting. In the context of the training session inside the ERAMIS project, they are even more valuable.

In general, participation of students during seminars was high. However, it was noticed a decline in participation when students and teachers attended the seminars together. In one of the seminars, one of the students was asked why they were not actively involved in the seminar. The answer was that, because of the respect they had for their teachers, they did not dare to ask questions in front of them. This fact can be taken as evidence that in the countries where seminars were conducted, there is still a large separation or gap between teachers and students. This comment should not be taken as a criticism to this situation, but as a finding of this situation, and as in some European countries this separation between teacher and student may have been lost, and eventually this lost has driven to a lost of respect.

Acknowledgment

We thank all colleagues and students in partner and beneficiary universities who are not necessarily formally involved in the ERAMIS Tempus project but who participated in organizing or conducting training sessions.

References

[1] M. Miłosz, “Computer Science Study as a Second Education for Social Science Graduated - The Tempus Project JEP-26235-2005.” in Proceedings of Science Practice Conference, pp. 9-13. Bishkek, Kyrgyzstan, 2009.06.15-18

[2] J-M Adam, S. Lujan-Mora, A. Merceron, M. Milosz and A. Toppinen, “European-Russian-Central Asian Network of Master's degree Informatics as a Second Competence”, in Proceedings of EDUCON 2012: The IEEE International Education Engineering Conference, Marrakesh, Morocco, April 2012, IEEE, pp. 722-729

[3] A. Merceron, J-M. Adam, S. Luján-Mora, M. Milosz, A. Toppinen “Faculty Development and Quality Assurance in the EU ERAMIS Project“, in International Journal of Engineering Pedagogy (iJEP), Volume 2, Issue 3, 2012, pp. 52-57.

[4] M. Lehner, “Viel Stoff – wenig Zeit”. Haupt, Wien, 2009.

[5] B. Ritter-Mamczek, Stoff reduzieren Methoden für die Lehrpraxis. Verlag Barabara Budrich UTB, 2011

[6] J.W. Han, M. Kamber, J. Pei, “Data Mining Concepts and Techniques”. Morgan Kaufmann, 2006.

[7] P.N. Tan, M. Steinbach, V. Kumar, “Introduction to Data Mining”, Addison Wesley, 2006

[8] RapidMiner rapid-i.com retrieved 29.11.2012

[9] E. Gamma, R. Helm, R. Johnson, J. Vlissides, “Design Patterns: Elements of Reusable Object-Oriented Software”, Addison-Wesley, 1994

[10] E. Freeman, E. Robson, B. Bates, K. Sierra, “Head First Design Patterns”, O'Reilly Media, 2004

[11] “The Portland Pattern Repository”, http://c2.com/ppr

PREPRINT - Proceedings of the IEEE International Education Engineering Conference (EDUCON 2013), p. 190-199, Berlin (Germany), March 13-15 2013. ISBN: 978-1-4673-6109-5. https://doi.org/10.1109/EduCon.2013.6530105

[12] T. Reenskaug, “Thing-model-view-editor – an example from a planningsystem”, Xerox PARC technical note, 1979