


Teaching and research activities

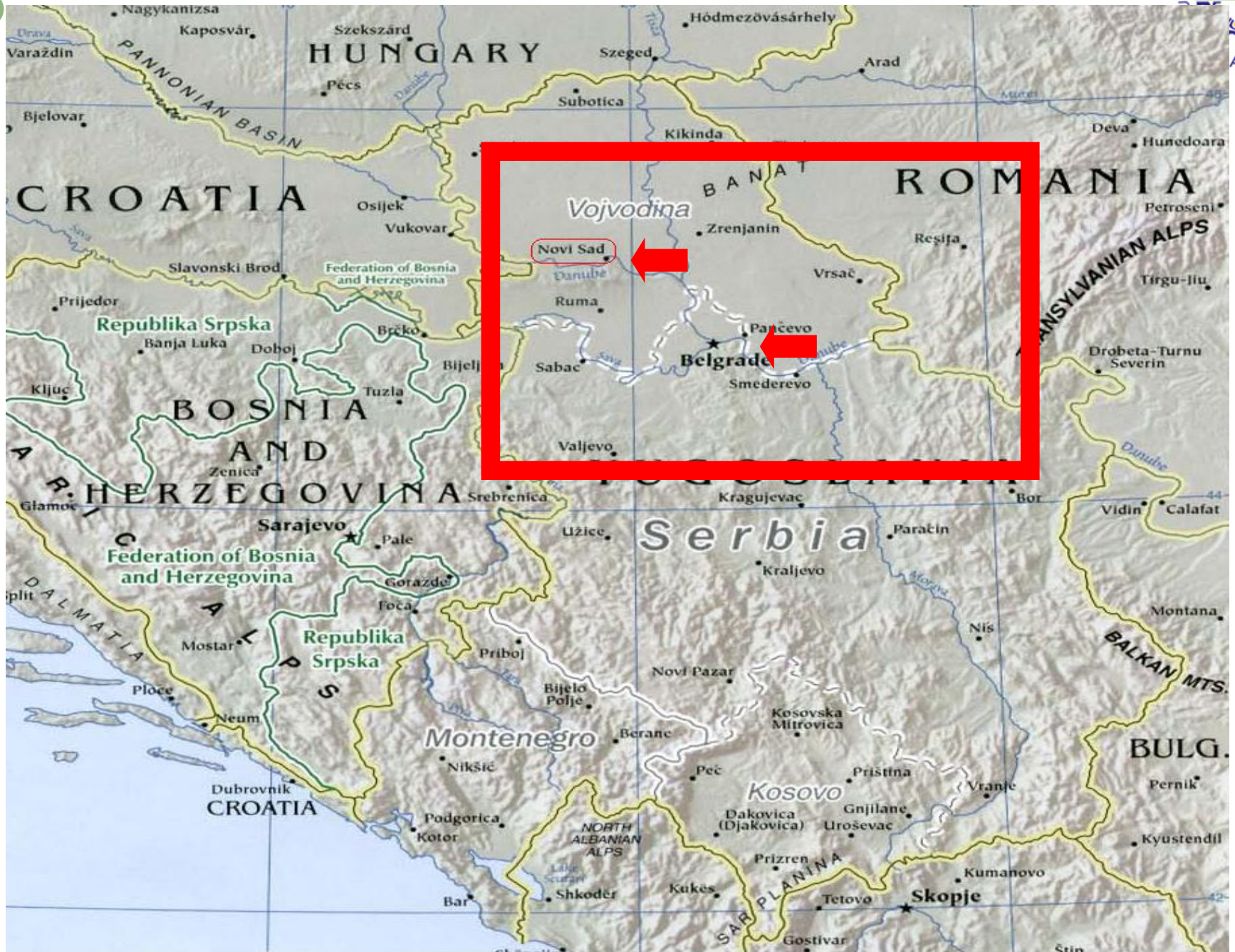


Mirjana Ivanović, University of Novi Sad, Serbia

AGENDA

- 
- Part I
 - General information
 - University, Faculty
 - Chair of Computer Science
 - Teaching
 - Research
 - Part II – Contemporary Research Topics





VOJVODINA



- Area:
21.506 km²
(comparable to
size of Netherlands)
- Population:
2 million
- Capital city:
Novi Sad
(Neoplanta,
Ujvidek,
Neusatz)

Vojvodina

- By its geographic location, cultural, historical and other characteristics belongs to Central-European region.
- Agricultural area.
- **Multinational:** 16 different nationalities
- **6 different languages in official use:** Serbian, Croatian, Hungarian, Romanian, Slovak, Ruthenian



The highest peak is 641m .



NOVI SAD

Neoplanta, Ujvidek, Neusatz





NOVI SAD

Neoplanta, Ujvidek, Neusatz



Novi Sad

- Established 300 years ago in 1694
- First settlements earlier than 3000 years BC
- **Today:**
 - **Capital** of the Autonomous Province of Vojvodina
 - **Population:** 350.000
 - **Cultural center of Serbia:**
 - SERBIAN NATIONAL THEATRE (The oldest theatre of the Serbs)
 - MATICA SRPSKA (The oldest scientific and cultural institution in Serbia)
 - EXIT (The biggest rock (music/multimedia) festival in South-East Europe)**
 - UNIVERSITY OF NOVI SAD . . .



AGENDA

- Part I
 - General information
 - University, Faculty
 - Chair of Computer Science
 - Teaching
 - Research
- Part II – Contemporary Research Topics

About University

- The University of Novi Sad was founded on 28 June 1960.
- Today it comprises 14 faculties located in the four major towns: Novi Sad, Subotica, Zrenjanin, and Sombor.
- Invested considerable efforts in intensifying **international cooperation**
- Participating in the process of university reforms in Europe.
- University of Novi Sad has come to be recognized as a **reform-oriented university in the region** and on the map of universities in Europe.

We within the University

- Faculty of Sciences (<http://www.pmf.uns.ac.rs/>)

Department of **Biology and Ecology**

Department of **Chemistry**

Department of **Physics**

Department of **Geography and Tourism**

Department of Mathematics and Informatics (<http://www.dmi.rs/>)

Chairs:

7 – In Different Mathematical Fields

Chair of Computer Science (<http://perun.pmf.uns.ac.rs/>)

Chair of Information Systems

Chair of Theoretical Computer Science

AGENDA

- Part I
 - General information
 - University, Faculty
 - Chair of Computer Science
 - Teaching
 - Research
- Part II – Contemporary Research Topics

Chair of Computer Science

- Staff
 - Zoran Budimac, full professor, Head
 - 1 full professors (Mirjana Ivanovic),
 - 6 assistant professor (Vladimir Kurbalija, Milos Radovanovic, Milos Savic, Zoran Putnik, Aleksandra Klasnja Milicevic, Gordana Rakic)
 - 7 assistants (several extra PhD students)
 - (<http://perun.pmf.uns.ac.rs/>)

- Teaching

- Projects, Research

Chair of Computer Science

■ Teaching

- Introduction to programming
- Object-oriented programming, Visual Programming, WEB Design
- Data structures and algorithms
- Programming languages
- Multimedia systems
- E-business
- Operating systems
- Compiler construction
- Software engineering
- Most of courses for software engineering MSc study
- Educational software, Methodics of Informatics and Programming
- ...

Teaching methodology

- Programming courses
- Software engineering course
- Privacy, ethics and social responsibility; Introduction to eBusiness
- Educational tools

Teaching methodology

Typical for programming courses

Duration (weeks)	13
Lectures (hours/week)	2 and 2 for theoretical exercises
Labs (hours/week)	2 hours/week, groups of 10-15 students
Homework	None
Evaluation (Grading)	Practical assignments (30%), three interim theoretical tests (30%), oral exam (60%)
Teaching approach	Blended learning. Several practical assignments, from simple to more complicated. Objects-first (within the course). Learner-centered.
Programming env./Tools	Web-based tutoring system. BlueJ. Jeliot.
Utilization of TEL tools	Moodle, Svetovid in-house submission system.

Teaching methodology

Typical for programming courses

LMSs and Tutoring Systems	LMS Moodle, Web-based tutoring system
Programming Environments and Tools	BlueJ. Jeliot.
Assessment tools	<p>LMS Moodle as a tool for testing students' theoretical knowledge (using Quiz module) and small tests for self-evaluation.</p> <p>In-house Svetovid system for assessing students programs.</p> <p>Use Moodle for administration of all points and final grades.</p>
Communication and cooperation	<p>Usually e-mails.</p> <p>Students intensively use LMS Moodle: discussion forums, instant messages, chat sessions, e-mail.</p>

Teaching methodology

Software engineering course

- **Assignment 1:** Reading and reviewing of the preliminary requirements specification and requirements specification for a case-study “Seminar Organization” (or alternative).
- **Assignment 2:** Application of the “Function point” method on the requirements specification, in order to calculate the price and the human resources needed for a chosen case-study.
- **Assignment 3:** Analysis of a product model, resulting from the application of structure analysis.
- **Assignment 4:** Development of a part of a static model by creation of class diagram and use-case diagram.
- **Assignment 5:** Development of a formal specification for several new operations, based on formal specifications presented during lectures.
- **Assignment 6:** Analysis and review of another teams’ solution of the fourth assignment “Development of the part of a static model”.
- **Assignment 7:** Application of software metrics methods, through usage of a tool.
- **Assignment 8:** Specification of a regression test.
- **Assignment 9:** Creation of “classification tree” for software testing.

Teaching methodology

Software engineering course

- **Team work**
- **Peer review approach in assessing assignments, discussions, critics,...** (solutions *are different*, so each one will have their opponents, students who would challenge and confront it, so the fruitful discussion would happen)
- **Questionnaires** for team members assessment
- Practice of application of teamwork at several other courses was spoiled by students who didn't do their part of the task
- **Usage of Wikis** for assignment solving combined with the ability of LMS Moodle to reveal all of the activities and history of changes, enabled us to separate actual doers and non-doers for each of the assignments, to the satisfaction of both teachers and students.

Teaching methodology

	+	Ok	-
<i>What do you think about the difficulty of the assignments?</i>	43%	36%	7%
<i>Did you find the assignments encouraging (motivating)?</i>	43%	29%	14%
<i>Do you think that working in a team was valuable for gaining realistic experience?</i>	36%	43%	
<i>How often did you wish to change a member of a team with some other member, or just "fire" her/him?</i>	14%	14%	50%
<i>Do you think that assignments solving would be easier, better, and more successful if you have done it alone?</i>	36%	14%	29%
<i>Most of the assignments did not have "exact correct answer". How much does that bother you?</i>	64%		14%
<i>Marks for the assignments were given based on "how much effort your team invested" more than anything else. Is that fair?</i>	60%		14%

Teaching methodology

Privacy, ethics and social responsibility; Introduction to eBusiness

- After successful usage of Moodle, elements of Web 2.0 grabbed our attention: start using **forums and Wikis** to simulate classic classroom activities.
- Forums were intended for discussions, role-playing games, and similar, using asynchronous method of communication.
- Assignments were given to students, with the appropriate time-frame to participate in discussion on given topic(s) and each of their posts was graded.
- Combination of several such graded forums and individual assignments of "file upload" type produced the final grade.

Teaching methodology

- **Educational tools – part of research**
- PROTUS
- PROgramming Tutoring System - provide learners with personalised courses from various domains.
- General tutoring system.
- It is an interactive system
 - allows learners to use teaching material prepared for appropriate courses,
 - includes parts for testing acquired knowledge.
- Components that support different recommendation techniques were integrated in Protus in order to allow:
 - standardization and formalization of content and

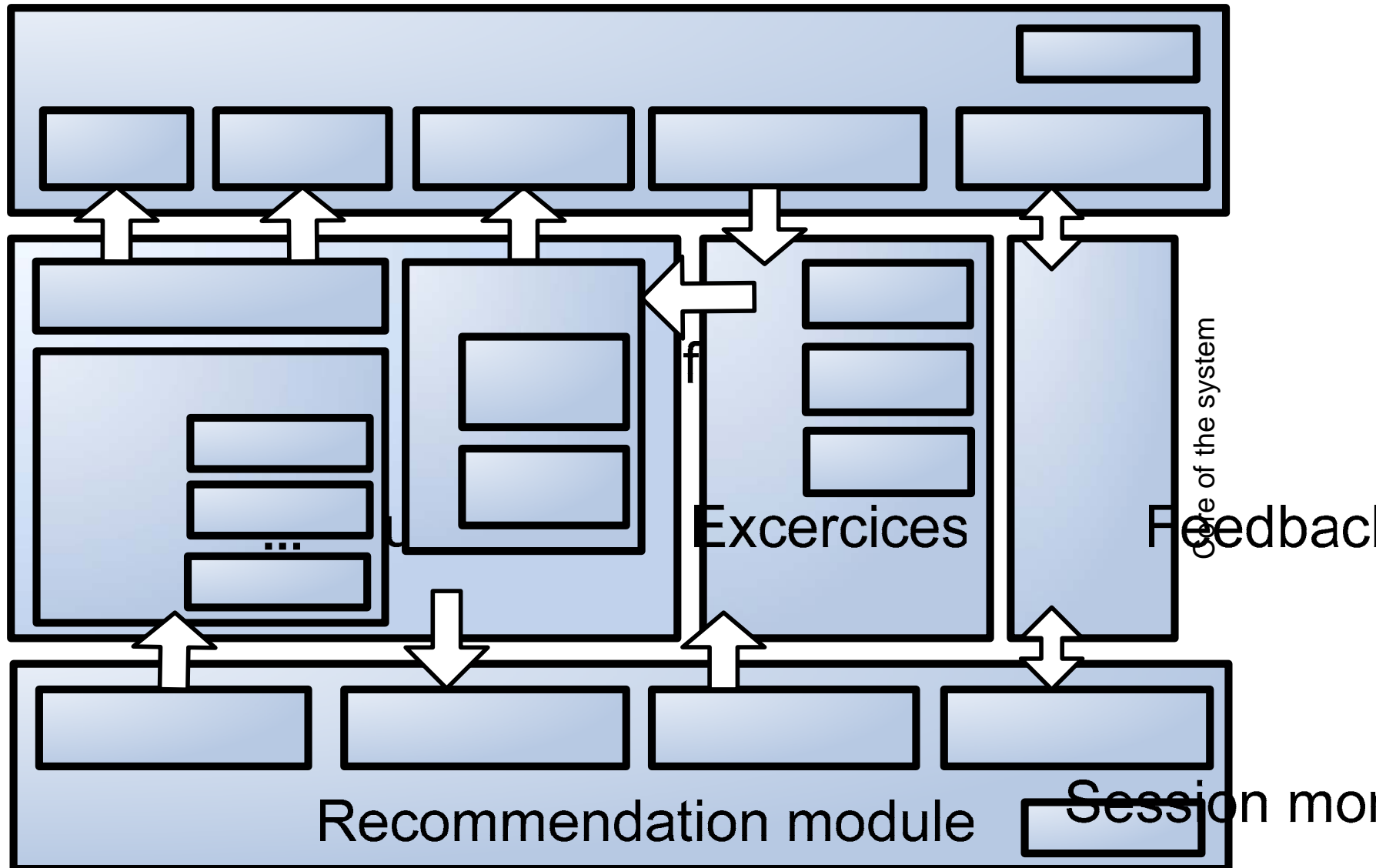
PROTUS

- The first completely implemented and tested version of the system was for introductory Java programming course.
- The course is designed for learning programming basics for learners with no previous object-oriented programming experience.
 - Beneficial for providing learners with personalized learning experience.
 - Useful for generating feedback for other key participants in the learning process – content authors and teachers.

PROTUS

- System architecture of Protus enables the development of courses in different domains in three phases:
 - creation of skeleton application with use of Vaadin Java framework,
 - creation of individual courses, appropriate teaching materials for each course as well as a set of appropriate tests for assessment of acquired learners' knowledge,
 - presentation of personalized learning materials to each individual learner.

PROTUS



PROTUS

- A repository of ontologies was built to achieve easier knowledge sharing and reuse, more effective learner modeling and easier extension.
- Educational ontologies for different purposes can be included in architecture design, such as:
 - **presenting a domain** - domain ontology,
 - **building learner model** - learner model ontology,
 - **presenting of activities in the system** - task ontology,
 - **specifying pedagogical actions and behaviours** - teaching strategy ontology,
 - **specifying behaviours and techniques at the learner interface level** - interface ontology.
- Protégé tool was used for development of ontologies and their translation into OWL.

ASQ?

- **Under: Institutional partnerships (SCOPES), Project no. 160480; Innovative teaching curricula, methods and infrastructures for computer science and software engineering, Switzerland, Slovakia, 2015-2018**
- **Plan to use ASQ? <http://asq.inf.usi.ch/> in some of our courses**
- ASQ is a Web application for creating and delivering interactive HTML5 presentations.
- It is designed to support teachers who need to gather real-time feedback from the students while delivering their lectures.
- Presentation slides are delivered to viewers who can answer the questions embedded in the slides.
- The objective is to maximize the efficiency of bi-directional communication between the lecturer and a large audience.

Chair of Computer Science Projects

- *Intelligent techniques and their integration into wide-spectrum decision support*
2011 – 2015, Supported by Republic of Serbia.
- *Infrastructure for technology enhanced learning in Serbia*
2011 – 2015, Supported by Republic of Serbia.
- **Institutional partnerships (SCOPES), Project no. 160480; Innovative teaching curricula, methods and infrastructures for computer science and software engineering, Switzerland, Slovakia, 2015-2018**
- **Thematic network project: "FETCH - Future Education and Training in Computing: How to support learning at anytime anywhere", 65 institutions, 35 countries, 2014-2016**
- **Emotional Intelligence and Applications Based on Multi-Agent Systems, 2013-2015.**
Bilateral project Fudna University, Shanghai, China

Chair of Computer Science Projects

- *Software Engineering: Computer Science Education and Research Cooperation*
2001-2016. Supported by: DAAD and Stability Pact for South Eastern Europe, Most of Balkan Countries, Germany.
- *Intelligent Techniques for Data Integration and Decision Support in the Medical Domain*, 2011-2012, DAAD Bilateral Project with Germany
- *Towards a better software metrics tool*
2010-2011, Bilateral project with University of Maribor, Faculty of Electrical Engineering and Computer Science, Slovenia
- *Correlating Images and Words: Enhancing Image Analysis through Machine Learning and Semantic Technologies*
2010-2011, Bilateral project with Institute "Jožef Štefan" (Ljubljana, Slovenia)

AGENDA

- Part I
 - General information
 - University, Faculty
 - Chair of Computer Science
 - Teaching
 - Research
- Part II – Contemporary Research Topics



Chair of Computer Science

■ Research:

- Educational Tools and different E-learning and learning aspects
- Agents, SOA, Applications
- Some Software Engineering Topics
- CBR – Case-Based Reasoning, Case Based Generator, Curve Based Generator
- Web Mining, Text Categorization, Time Serises and Hubness

Educational Tools and Different E-learning and Learning Aspects

■ Tutoring system

- Learning styles, Tagging and Recommendations in order to support personalization, Ontologies
- Case study: System for teaching Java

■ **Submission system:** submission students' programs/code

- Uses in everyday practice
- Applied for different programming courses

■ Development of different **components for Moodle system** – Adaptive courses, Forming teams for teamwork projects

Agents, SOA, Applications

- **Previous research (more than 14 years): Decentralized Mobile Agent Architecture, applied in WFMS**
- **Current research**
 - **SOA-based MAS (SOM):** an implementation of a runtime environment for agents based on web services, with the idea of **building an interoperable, standards-compliant architecture**
 - **Agent Language for SOM (ALAS):** provides developers with language constructs that **hide the complexity of agent development**
 - More importantly, **the compiler can regenerate executable code of an ALAS-based agent to match the underlying platform (e.g. Java or Python)**
 - The main goal is to support **transparent agent migration in heterogeneous networks**

Agents, SOA, Applications

- Some additional research directions
 - Using **mobile agents** for building and maintaining **fault-tolerant networks**, featuring automated failure detection
 - **Metadata harvesting learning objects**: collecting new learning materials for students from remote (heterogeneous) repositories
 - **Pedagogical agents** that motivate and guide students in e-learning systems (teaching programming, exercises)

Some Software Engineering Topics

Software Metrics, Software Quality - **SSQSA Architecture**

- **eCST generator** - enriched Concrete Syntax Tree
 - constructs an internal eCST equivalent for input program
 - Based on parsers for different languages
 - Applied:
 - **Languages:** Java, C#, Modula-2, Delphi, Pascal, WSL, OWL i ITL, Haskel, Erlang
 - **Metrics:** Halstead, CC, LOC, OO Metrics, Design Metrics

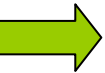
- **BACK-END tools**
 - SNEIPL, an extractor and analyzer of complex software networks,
 - SSCA, an analyzer of software structure and its changes with an access to a software repository
 - SMIILE, a calculator of software metrics with its own repository of metrics history

Analysis of software networks

- **Software network - dependencies among software entities** (functions, classes, etc.) in a software system
- Investigated to understand **structural complexity and evolution of large-scale software systems**
- Analyzed under the framework of **complex network theory (theory of scale-free and small-world networks)**

AGENDA

■ Part II – Contemporary Research Topics

- 
- CBR – Case-Based Reasoning, Case Based Generator, Curve Based Generator (dr Vladimir Kurbalija)
 - Time Serises and FAP framework
 - Web Mining, Text Categorization, Hubness (dr Miloš Radovanović)

Case-Based Reasoning, CaBaGe, CuBaGe

CaBaGe – Case Base Generator

- **CBR shell** - Realized for easy use of non-programmers, text and numerical form of cases
- **Usage of - Case Retrieval Nets (fast search for big databases)**
- **Basic, core framework that can produce decision support systems**
- **Application in Multiple Sclerosis Diagnosis**

Case-Based Reasoning, CaBaGe, CuBaGe

CuBaGe – Curve Base Generator

- Analyses of curves (graphical cases), **compares them to similar curves from the past and possibly predicts the future behaviour** of the current curve (spline based)
- Indicated by different practical needs: **Classification, Indexing, Clustering, Prediction (Forecasting), ...**
- **Time Series Similarity Measures:** *Minkovski Metrics, Dynamic Time Warping (DTW), Distance based on Longest Common Subsequence (LCSS), Edit Distance with Real Penalty (ERP), Edit Distance on Real sequence (EDR), etc*
- Cooperation with the company “Novi Sad Fair”, the combination of **CBR and data warehousing**

AGENDA

- Part II – Contemporary Research Topics
 - CBR – Case-Based Reasoning, Case Based Generator, Curve Based Generator
 - - Time Serises and FAP framework
 - DMGUI
 - SCVGUI
 - Web Mining, Text Categorization, Hubness

Time Serises and FAP framework

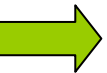
- FAP - Framework for Analysis and Prediction** - a multipurpose, multifunctional system, open source, easily extendable
- FAP system will **contain all main features** needed for time series analysis: **pre-processing tasks, similarity measures, time series representations**
 - Specialised for different **data mining tasks**: indexing, classification, prediction etc.
 - FAP supports all mentioned concepts with the possibility to **easily change some existing or to add new concrete** (more efficient) implementation of any **concept**

Time Serises and FAP - Conclusions

- FAP already tested in several bigger domains
- On behavioural data, gained from experimental lab system, **Socially Augmented Microworld** with three human participants
- **Medicine/Nephrology domain (Kidney transplantation)**

AGENDA

- Part II – Contemporary Research Topics
 - CBR – Case-Based Reasoning, Case Based Generator, Curve Based Generator
 - Time Serises and FAP framework
 - Web Mining, Text Categorization, Hubness



Web Mining, Text Categorization, Time Series and Hubness

- A new meta-search engine **CatS** – utilizes text categorization to **improve the presentation of search results** obtained from a major Web search engine.
- **Classifiers** trained on **Web-page titles and descriptions from dmoz** Open Directory using the **WEKA** workbench
- **Text Classification Experiments**
 - **Document Representations** - Bag-of-words (BOW)
 - **Classifiers:** Complement Naïve Bayes, Support Vector Machines (Sequential Minimal Optimization (SMO)), The Perceptron (Voted Perceptron (VP)), k-Nearest Neighbor (Instance-Based Learner (IBk))

Multidimensional Data and Hubness

Motivation

- Data is often represented in form of a **table where rows are data items and columns are the attributes** that describe them
- If all values are numeric, data items can be viewed as **points (vectors) in a multi-dimensional vector space**
- A large number of columns (**high dimensionality**) can cause **problems** in many application areas, commonly referred to as

“The curse of dimensionality”

University of Novi Sad, Serbia
Faculty of Science
Department of Mathematics and Informatics
Mirjana Ivanović

