Multidisciplinary research projects feeding master curricula

*Dan Cristea, Dorel Lucanu, Adrian Iftene*
{dcristea, dlucanu, adiftene}@info.uaic.ro
Faculty of Computer Science
“Alexandru Ioan Cuza” University of Iasi, Romania
The context

• The Alexandru Ioan Cuza University
• The Computer Science Faculty
  – The Didactic Department:
    • 3 years background in CS
    • 2 yours master studies: 5 master programmes
    • 3 years doctoral studies
  – The Research Department:
    • national and international projects
The 2009 experiment

- Use on-going research projects as sources of inspiration for student projects at master level.
- Involve students of different CS master specializations in common research projects
Motivation and aims

• Students get acquainted with hottest research topics
• Students learn to interact in teams
• Ideas developed by students could be used in the projects
• Sources of ideas for new projects
• Improvement of the curricula
• Orientate skilful students towards PhD
Context: master programs in FII

• Computational Linguistics (CL)
  – study languages from a computational perspective; develop applications in the new and spectacular field of NLP
  – taught to use resources (grammars, lexicons, corpora etc.) and technologies (machine translation, information retrieval from texts, document classification, text understanding, dialog systems, parsing, textual entailment, question-answering)
Context: master programs in FII

- **Software Engineering (SE)**
  - form specialists capable of performing engineering activities in the field of software systems
  - elements of analysis and design, advanced programming techniques, quality of software systems, project team work, economic engineering and project management
Context: master programs in FII

- Computational Optimization (CO)
  - advanced techniques for modelling and solving a wide range of optimization problems in industry, engineering, economy, optimal control, transport, communication, medicine and biology
  - elements of linear programming, integer programming, combinatorial optimization, optimization methods inspired from nature
Context: master programs in FII

• Distributed Systems (DS)
  – design, implementation and verification techniques for distributed systems
  – parallel algorithms, distributed processing on clusters and grids, distributed operating systems, architectures for parallel computing etc.
Context: master programs in FII

• Information Security (IS)
  – playing a very important role in domains like: government, military security, banking and financial activities, education, transport, medicine, agriculture, legislation, recreational activities
  – fundamentals of information security, the main research topics in this field
A case study: the project CLARIN

What:
• Create a European infrastructure that makes language resources and technology (LRT), available to scholars of all disciplines, especially social sciences and humanities

How:
• Putting together existing digital archives into a federation of archives with unified web access
• Provide language and speech technology tools as web services operating on language data in archives

http://www.clarin.eu

From Steven Krauwer
According to the CLARIN philosophy

• Help HSS (CS or CL novices): access and integrate language processing tools in meaningful processing chains

• Tools?
  – elementary language processing modules
  – accessible as web-services
  – recognizable by unique identifiers
  – process language resources in multiple languages
  – can be interconnected in complex processing chains
Usage scenarios in CLARIN

• To show the usefulness of the CLARIN technology: a call for user scenarios addressed to HSS researchers
• 29 received
• 4 selected by a CLARIN committee to serve as best examples

– see *Usage scenarios and basic workflows* by Valeria Quochi, in CLARIN Newsletter no. 6, online at [http://www.clarin.eu/newsletter](http://www.clarin.eu/newsletter)
Inspired from a winning CLARIN usage scenario

• *What we are is what we eat* ([http://www.clarin.eu/scenarios](http://www.clarin.eu/scenarios))
  – interpret NL in recipes books and compare with basic data stored in databases or ontologies in order to:
    • compute caloric intake of prepared food
    • extract nutritional habits of certain historical periods
    • compare national food habits based on caloric value
    • compute diets in doctor’s cabinets
    • display caloric intake of food in restaurants
Challenges addressed by this project

• **NLP**
  – some tools may exist already ➔ learn to reuse them
  – some have to be written ➔ understanding the principles

• Need additional data: an ontology (of food) ➔ search the Web for resources

• Language independent formulation
Project methodology

• Problem context identification
• Designing the solution
• Implementing the solution
• Testing, validation, and evaluation
Problem context identification

– students should discuss and evaluate the problem
– appreciate the overall feasibility
– foresee possible applications
– appreciate to what degree is the problem bound to a certain input language. Could it be formulated as language independent?
– organize the teams
– propose and approve evaluation criteria

• participants: all
• time: beginning of the term
• method: brainstorming group sessions
Designing the solution

- design the architecture of a system (CL, SE, DS)
- identify the type of the component parts (CL)
- identify the type of the component resources (CL)
- discuss inter-module communication standard (CL, SE, DS)
- appreciate optimization and security issues (CO, IS)

• Period: the next 2 weeks
• Method: group work
Implementing the solution

– target or implement the component parts
  • if they exist already, use them (CL)
  • if they have to be created, write code (SE, CL, CO)
– if needed, build the necessary wrappers (SE, DS, CO, IS)
– target the necessary component resources (CL)
  • build some, if needed (CL)
– test the component parts (all)

• Period: until 3/4\textsuperscript{th} of the term
• Method: individual work with weekly team meetings
Testing, valuation and evaluation

– test the whole system (all)
  * repair if needed (mainly SE)
– propose state-of-the-art baselines and valuation criteria (CL)
– formulate conclusions, appreciate the solution (CO)
  * suggest redesign if the valuation places the solution under expectations (CO, SE, DS, CL)

• Period: last 1/4\(^{rd}\) of the term
• Method: individual work and weekly team meetings
• Final assembly presentation of solutions
Conclusions

• Targeted skills:
  – team work: communicate, have a common goal, respect deadlines
  – Integration: learn to interconnect different parts in functional workflows, search for optimum, redesign if needed
  – valuation: learn to valuate a solution, to compare against state-of-the-art
Conclusions

• Distillate skilful researchers
  – exploit the “catching” world of the research
  – 3 years for a PhD program is too short... ➔ start earlier during the master
Thank you!