Group for Programming & Software Engineering (PSE), master thesis
22th September 2015
PSE, at the core of computer science

- **OO Programming and programming languages**
  - Making new programming language mechanisms (packages of classes)
  - Let the compiler catch more programming errors when using patterns.
  - Make a small/medium sized system

- **Modelling and design patterns**
  - Can modelling (like UML) be made executable, and programs models?
  - Design patterns for distributed and parallel programs
  - Design Patterns for Secure Programming

- **Software Engineering**
  - Evaluating the various System Development methods.
  - Investigating and finding the ‘best’ agile method.
  - Making tools (eclipse) for embedded systems

- **Parallel tools & algorithms**
  - New parallel algorithms, triangulation
  - Comparing GPU/Multicore/Grid computing; tools and efficiency

- **Emerald/cloud computing**

- **Security**
How to contact us at PSE for a Master thesis:

- Parallel algorithms: Arne Maus, arnem@ifi.uio.no
- Language: Stein Krogdahl, steinkr@ifi.uio.no
- Eric B. Jul, ericbj@ifi.uio.no
- Modelling: Birger Møller-Pederesen: birger@ifi.uio.no
- ; Ragnhild Kobro Runde: ragnhilk@ifi.uio.no
- Software Engineering: Dag Sjøberg: dagsj@ifi.uio.no
- Yngve Lindsjørn, yngelin@ifi.uio.no
- Knowledge Federation: Dino Karabeg: dino@ifi.uio.no
- SINTEF (Arne Jørgen Berre, Ketil Stølen) -
The effect of level of parallelism (threads) on performance of algorithms.

- Which effect do we get from increasing the number of threads (way) beyond the number of cores we have.
  - Sorting algorithms (full parallel Quicksort, left and right Radix, improved Shellsorting)
  - Delaunay triangulation, prime number factorising

- Two different machine configuration
  - Shared memory multicore (4-16, 64 cores)
  - Abel-cluster at Ifi (10 000 cores + XeonPhi)

- Practical testing, measured execution times

- Kontakt: Arne Maus arnem@ifi.uio.no
Master topics – Teamwork and Large Scale Agile Software Development

1) Conduct a literature review of research on teamwork and the relation to team performance and personal success in software development

2) Conduct an empirical study (qualitative and/or quantitative) on how teamwork factors such as team leadership, team cohesion, communication and self management effect team performance and personal success in software development teams.

3) Large-Scale Agile Software Development. Agile methods were first used in small projects with little criticality. How can agile practices be adapted and combined with traditional practices to function effectively in large-scale development and multi-team environment?
   Challenges:
   • System architecture across teams
   • Consistency across teams for the development practices
   • Inter team dependences
   • Team leader role and product owner role
Contact

Yngve Lindsjørn
ynglin@ifi.uio.no
91549139
Assessing programming tasks/exams

- Do teachers, examiners and teaching assistants agree on how programming tasks and exams should be evaluated?
- What criteria do the students believe are being used?
- Will performance improve if the criteria are made more explicit?

Contact: Ragnhild Kobro Runde, ragnhilk@ifi.uio.no
Eric B Jul -

- Thesis topic(s) on:
  - Cloud Computing and Design Patterns
Master thesis proposals

Birger Møller-Pedersen/Ragnhild Kobro Runde
[ birger@ifi.uio.no ]/[ ragnhilk@ifi.uio.no ]

- Domain Specific Language (DSL), UML-profile or framework?
- Combined modelling and programming?
- Modules as objects
Domain Specific Language (DSL), UML-profile or framework?

- In various EU-projects we have been involved in making a number of Domain Specific Languages.
- Often there are good reason for doing so, but in general one may have the choice between:
  - A Domain Specific Language,
  - A UML-profile (or annotated Java),
  - A framework of classes (in UML or Java).

- The thesis shall compare these alternatives, preferably on a case that you come with yourself, otherwise we define one as part of the thesis. Existing DSL are candidates as cases.
Combined modelling and programming

- Are modelling languages to become executable? Are programming languages to become useful for modelling? Or should we rather opt for a combined language?

- Such a language must be designed so that it supports mechanisms in modelling just as well as it supports programming.

- The thesis shall investigate which modelling mechanisms may become part of a combined language, and which do not belong in an executable language.

- Instead of making a complete new language, the thesis extends e.g. Java, Scala or C# (or whatever language you prefer) with the identified modelling mechanisms, with respect design and in implementation. In addition the extension have to be validated on examples.

- May be based upon theses that have done this for associations and state machines.
Modules as objects

- Some languages designers advocate something like modules in addition to objects and classes, and these should be different from objects.
- Most languages have a package-like concept: a grouping of types, interfaces and classes, but it does not qualify.
- Package Templates (developed at Ifi) generalize packages: Templates may be tailored to specific needs and templates may be combined by means of instantiations:
  - Renaming classes and renaming properties of classes
  - Combining classes from several templates into one class
  - Adding properties to classes as part of instantiations, and override methods
- All of this is done on the program text/AST, but still type safe: type properties of templates are maintained in the instantiated package.
- How many of the PT-mechanisms may also be supported by a module concept where modules are objects??
Knowledge Federation MS Themes
Autumn, 2015

---

**SAŠA RUDAN**

sasharu@ifi.uio.no

---

**SINIŠA RUDAN**

sinisa.rudan@gmail.com

---

**DINO KARABEG**

dino@ifi.uio.no
CollaboScience (CS) is a cyberinfrastructure for supporting transdisciplinary scientific research
  built as a set of pluggable components (following SoA)
  increasing scientific collaboration
  supporting collective intelligence
  promoting dialogical science
  optimizing socio-business processes
Video demo (https://vimeo.com/140002373) (TNC)
List of Theses

1. RIMA
   Mapping communities, interests, and resources, visualization, SNA

2. DataTalks
   Advanced knowledge construction, data and processes integration

3. TopiChat
   Transformative Dialogue: actions, decisions and knowledge creation

4. Additional theses available on demand:
   KnAllEdge Visualization (graphs, knowledge mapping, D3)
   CS infrastructure (knowledge, plugins, processes mapping & mining)
   KnAllEdge Browser Extension (integrating with browser)
   Creativity, Art and Collaboration (Inclusion of mechanisms of Creativity in CS & sister project CollaboArte)
RIMA (Resource and Interest Mapping)

- implements **metrics** for detecting **user-to-user** or **user-to-item** similarities/closeness
- helps individuals to **map** their **interests**, build **reputation** and **match** common interests
- helps with **serendipity** discovery, team and **collaboration** structuring
- reduces information overload through **semantic filtering**
- built on **WhoAmI** concept

**Task**
Support interactive **WhoAmI** mapping and RIMA matching
Statistical and graph visualization of RIMA relationships

**Technologies**
**MEAN** (MongoDB, Express, Angular, node.js)
DataTalks

- CS data-model for simplified and integrated representation of complex data and processes
- governs data transformation
- high-level data changes and states are presented with state diagrams
- transformation processes utilizes BPMN-like (UML Activity-like) diagrams

Task
Creating interactive in-browser state and BPMN diagrams as KnAllEdge data representation
Mapping between Mongo DB (NoSQL DB) and data states and processes

Technologies
MEAN (MongoDB, Express, Angular, node.js)
D3 visualization & responsive HTML 5
TopiChat

dialogic companion to collaborative building and transforming knowledge inside the CS
implemented as semantically augmented chat component
supports generic grammars for transforming dialogue into knowledge,
decissions, sense-making (i.e. IBIS, …)

Task
Join us in working on TopiChat
Extend semantic and NLP mappings

Technologies
MEAN (MongoDB, Express, Angular, node.js)
responsive HTML 5
stream.io & stream.io plugins
SoundCloud API integration
basic NLP (Natural Language Processing)
Sintef – two adjunct prof. at Sintef

- Arne Jørgen Berre
- Ketil Stølen

Will present Master theses proposals
Sintef 30. Sept. at 14.15