Group for Object-orientation, modeling, and languages (OMS), master thesis
18th September 2013
OOM, 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.
  - 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
  - Working extensions to the Joly system

- **Cloud computing**

- **Security**
Courses to take fall 2013/spring 2014:

- Take ≥ 30 credits from this list

Fall 2013:
- Programming languages (INF 3110) – 10 credits
- Algorithms and efficiency (INF 4130) – 10 credits
- Process Improvement and Agile Methods in Systems Development (INF 5181) – 10 credits
- Unassailable IT-systems (INF 5150) – 10 credits

Spring 2013:
- Models of concurrency (INF 4140) – 10 credits
- Distributed Objects (INF 5510) – 10 credits
- Compiler technique (INF 5110) – 10 credits
- Operating systems (INF 4151) – 20 credits
How to contact us at OMS for a Master thesis:

- **Parallel algorithms**: Arne Maus, arnem@ifi.uio.no
- **Language**: Stein Krogdahl, steinkr@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
- **Knowledge Federation**: Dino Karabeg: dino@ifi.uio.no
- **+ Eric, Sasha,**
1. Comparing two parallel programming environments:

1) C++ med Intel Clik eller Intel Threading Building Blocks, TBB,
2) Java on a multi core CPU.

Both a more theoretical comparison and practical programming a small number of (sorting) algorithms comparing execution times, ease of programming, debugging times etc.

2. Implementing ‘ParallelRecursiveProcedures’ in Java on a set of multicore CPUs interconnected by ordinary broadband.

recursion tree

Threads on a multicore CPU

single machine, stack
Master thesis proposals

Birger Møller-Pedersen
[ birger@ifi.uio.no ]

- Domain Specific Language (DSL), UML-profile or framework?
- Combined modelling and programming?
In various EU-projects we have been involved in making a number of Domain Specific Languages.

Often there are good reasons 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. The 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 parts 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.
Proposal 1: Make compilers more helpful

It would sometimes be convenient if we could ask the compiler to check that the program follows some requirements that we can describe for the compiler. It could e.g. be:

- That the declared names in a certain class all start with a given prefix.
- That there are no calls in class B to methods in class A.
- Your company may have certain guidelines for how programs should be written, and we want the compiler to check that they really are over held.

The master work: Would be to design a language in which we could describe such requirements.

The compiler would then read this description before compiling a program, and it should then do the described tests during compilation.

This should be implemented in a suitable compiler as far as possible.

We have compilers and tools that could be suitable for such an implementation.
Proposal 2: "Finish" the implementation of Java-PT

Package Templates (PT-mekanismen)

- A mechanism for writing separate "modules" (supports separation of concerns)
- The modules are called "package templates" or just "templates"
- Good features:
  - A template can contain any number of classes, that can e.g. be parameterized together
  - When used in a program, templates can be "taylored" to their new environment (with additions to the classes, merging of classes, "deep" renaming, etc.)
- Templates are taken into the program at compile-time, giving a simple type system

The master work: We have compiler for Java-PT (= JPT) that does not implement all aspects of PT, and that still has (probably many) errors. The work will be to bring the compiler as far as possible towards general usability.
Proposal 3: How to give good error messages at runtime

I'm not quite sure that this is enough for a masters thesis, so the first step of the work will be to look into that!

- A compiler for a high level language may produce different types of code:
  - Detailed machine code
  - An intermediate code, like byte-code for Java or CIL-code for .Net.
  - Code in another high level language (like we do in our Java-PT compiler).

- Especially in the last case (but to some extent also in the second) the program will be compiled with a compiler that only has an understanding of the language it compiles. E.g. from our own Java-PT compiler:
  - Assume we compile a program in Java-PT (including a number of separate templates) into a Java program, that is afterwards compiled with a traditional Java compiler.
  - Thus, if we afterwards get a run-time error during execution of our program, the error message will refer to the line number in the ”intermediate” Java program.
  - However, this is not understandable for the PT programmer. A reference to a certain template in his/her PT program and the line number in this template would be much better

Master work: Look at how such situations are handled in different systems, propose mechanisms/tools for tackling this better, and implement them as far as possible.
Two Thesis Topics in Knowledge Federation

Knowledge federation

- ‘Programming’ on the scale of the Web.
- Systemic innovation.

Brief historical introduction

Many years ago I dreamed that people were talking seriously about the potential of harnessing our technological and social nervous system to improve the IQ of our various organizations.

- 1951 Engelbart’s dream
- 1968 The Demo
- 2008 Program for the Future, First KF workshop
- 2010 Second KF Workshop, IUC Dubrovnik
- 2011 Third KF Workshop, Stanford University
- 2012 International KF Course, IUC Dubrovnik
Service-Oriented Architecture for Open Innovation

What’s it about
Developing a model service-oriented architecture in a concrete situation

What you’ll do
Draft a service-oriented architecture model for Induct Software and create a minimal prototype.

What you’ll learn
About service-oriented architectures and open innovation
WS-BPEL language and possibly other business process modeling tools and techniques.
<table>
<thead>
<tr>
<th>Topic</th>
<th>Hacking Bullipedia</th>
</tr>
</thead>
<tbody>
<tr>
<td>What’s it about</td>
<td>Recreating the conventional cookbook—by participating in a small project team on Hacking Bullipedia project</td>
</tr>
<tr>
<td>What you’ll do</td>
<td>Take part in developing a DebateGraph-based procedure and toolkit for Wikipedia-like interactive cookbook</td>
</tr>
<tr>
<td>What you’ll learn</td>
<td>The practice of systemic innovation in a team. DebateGraph and other selected collective knowledge-work tools and techniques.</td>
</tr>
</tbody>
</table>

Dino Karabeg, OMS Group, Ifi, UiO
dino@ifi.uio.no
Phone 22 85 27 02
Mobile 412 92 049
Office 9466
1) **DebateGraph Web Browser Integration**:

- [www.debategraph.org](http://www.debategraph.org)
- DebateGraph purpose is helping big communities on agreeing on arguments or knowledge.
- It organizes arguments/knowledge in a map
- It has been used in the White House, CNN, conferences, etc
- We are strategic partner with DebateGraph team
- We believe that DebateGraph lack natural way of integration into common process of knowledge capturing
1) DebateGraph Web Browser Integration

**Task**
- Create a support for integration of DebateGraph into WebBrowser (Chrome, …) as a **Chrome Extension**
  - We have a **similar project** implemented as a Chrome Extension
  - You will have a good starting point and original developer to help

**You will learn**
- Knowledge management concepts, Chrome Integration, advanced JavaScript, HTML5 techniques, Client-Server, Visualizing libs

**Potentially**
- Future **research** and **conference paper** in a domain of Knowledge Management and Arguments Negotiation
2) WhoAmI HTML5 Integration

- **WhoAmI** is a service helping individuals to **explain their interests**, build **reputation** and **match** on the base of common interests.

- Each actors’ interest is represented as a **WhoAmI object**, which is defined by:
  - **WHAT** interest user is relating himself to
    - (e.g. volleyball, dance, writing, etc)
  - **HOW** he is related to it
    - (e.g. like, practice, teach, etc)
  - **WHERE** defines privacy/visibility rules
WhoAmI implements **metrics** for detecting **user-to-user** or **user-to-item** similarities/closeness.

**Task**
- It is developed as a **RESTful** (JSON) Web Service
- We want to create **HTML5 AJAX** responsive interface

**You Will Learn**
- HTML5, responsive design, AJAX, ranking theories

**Potentially**
- **Research** and **conference paper** on crowd-source rankings and SNA actors interests-matching
Programming didactics:

- How do programming students approach new programming tasks?
  - How do they analyse the task, what steps do they take, what do they do in the presence of errors?
  - Are there differences with respect to gender, programming experience, self-expectations?

- What misconceptions are still present for students in their second/third course?
  - What is the relation between understanding/misconceptions and problem solving strategies?
Master Thesis Proposals

Eric Jul, Professor II
[ ericbj@ifi.uio.no ]

- Near-Far Cloud on Planetlab
- Design Patterns for Near-Far Cloud
- Spasialization of Sound in Argumented Reality
Near-Far Cloud on Planetlab

- Traditional Cloud computing involves a relatively thin client, e.g., a PC or a smartphone, accessing a faraway large data center, e.g., Google, Amazon.

- Near-Far cloud is a new Cloud Computing Model where there are compute and storage facilities close to the user at the edge of the cloud – not merely at a faraway data center.

- This project will show the benefits of the Near-Far Cloud, and specifically will show that mobility of storage and computing between can be beneficial to performance.

- The project will involve experiments with this new Cloud model using Emerald running on Planetlab.

- The project will be in cooperation with Bell Labs Ireland.
Design Patterns for Near-Far Cloud

- Traditional Cloud computing involves a relatively thin client, e.g., a PC or a smartphone, accessing a faraway large data center, e.g., Google, Amazon.

- Near-Far cloud is a new Cloud Computing Model where there are compute and storage facilities close to the user at the edge of the cloud – not merely at a faraway data center.

- This project will look at Design Patterns for this new cloud model.

- Experiments will involve Emerald on both local PC, nearby data centers and faraway Cloud centers.

- The project will be in cooperation with Bell Labs Ireland.
Spasialization of Sound in Argumented Reality

- Augmenting physical, real-world environment elements with computer-generated sensory input
  - Sound
  - Visuals
- Key requirement
  - Augmentation must be integrated as realistically as possible with the real world
- Smartphone AR
  - Limited processing power
  - Limited battery, even when CPU power is available

- Spatialisation: Make sound appear to originate from a certain point in the real space (or to move around), e.g., a horse galloping across in front of you should sound like it moves from, e.g., left to right.
  - Common technique in console games, PC games, etc.
  - Helps players orient themselves in the game world and improves their immersion

- Project: Add Spasialization of Sound to an existing AR application by moving the spatialization away from the smartphone and onto a nearby cloud computing element.

- Project will be in cooperation with Bell Labs Ireland and Haunted Planet Studios in Dublin

Figure 3. ILD, ITD and HTRF [14]

Haunted Planet Studios, 2013
Teamwork in software development
Effectiveness and Efficiency in agile software teams

- Improving the effectiveness and efficiency in software development will often involve improving the way the teamwork is organized. Examples of teamwork factors:
  - Team leadership
  - Communication
  - Coordination
  - Cohesion
  - Effort
  - Support
  - Balance of team member contribution
  - Self Management

How will the quality of these aspects of teamwork influence:
1. The quality of the software product (e.g. functionality, robustness, reliability, and performance)
2. The quality of the software project (time and cost)
3. The team member satisfaction

Yngve Lindsjørn
ynglin@ifi.uio.no
Scrum - example of an Agile process model

**Daily Standup – 3 Questions**
- What have you done since yesterday?
- What are you planning for tomorrow?
- Do you have any obstacles?

**Scrum Master**
- Coordinates the agile process and daily stand up
- Shields the team members from interruptions
- Coordinates SPRINT planning and the estimating of Sprint Backlog

**Scrum Team**
- Typically 5-9 persons
- Developers, testers, architects,…
- Self-organized

**Product Owner**
- Represents the customer
- Coordinates the product backlog, prioritize task (user stories) from the backlog.

---

![Sprint X Burn Down Chart]

<table>
<thead>
<tr>
<th>Work left</th>
<th>Time (days)</th>
</tr>
</thead>
<tbody>
<tr>
<td>800</td>
<td>100</td>
</tr>
<tr>
<td>700</td>
<td>200</td>
</tr>
<tr>
<td>600</td>
<td>300</td>
</tr>
<tr>
<td>500</td>
<td>400</td>
</tr>
<tr>
<td>400</td>
<td>500</td>
</tr>
<tr>
<td>300</td>
<td>600</td>
</tr>
<tr>
<td>200</td>
<td>700</td>
</tr>
<tr>
<td>100</td>
<td>800</td>
</tr>
</tbody>
</table>

---

![Daily Scrum Meeting]

**24 Hours**

**2-4 Weeks**

**Potentially Shippable Product Increment**
Teamwork in software development

3 master thesis:

1. Make a literature review of research on teamwork in software development.

2. Make an empirical study (qualitative and/or quantitative) of how teamwork factors effect team performance and team member satisfaction 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?
Common Variability Language (2 theses)

- Øystein Haugen offers two Master theses within Product Lines
- CVL is a language for describing software product lines. It has been up for standardization in the OMG (Object Management Group) Haugen has been organizing this effort.
- More info on CVL on http://variabilitymodeling.org
- Contact Øystein Haugen
  - oysteinh@ifi.uio.no or Oystein.haugen@sintef.no tel.: 913 90 914
- #1 Experiment with the Common Variability Language
  1) Perform case studies - toy examples and real examples
  2) Make intricate examples to highlight problematic areas and suggest mitigation
- #2 Product Lines vs. Adaptivity
  1) Investigate commonalities and differences
  2) Compare CVL (PL technology) and Diva ML (adaptivity)
Ketil Stølen tilbyr:

**MSc-oppgaver på sikkerhet, risiko og tillit**

Informasjon om pågående forskningsaktiviteter finner du her:

http://heim.ifi.uio.no/~ketils/

Vil du vite mer, ta kontakt via telefon 92216112 eller email kst@sintef.no
Masteroppgaver i BEKK

- Samarbeid med en av BEKKs mange faggrupper
- Dyktige og inspirerende veiledere
- Stort spenn i tema: Alt fra brukeropplevelse og mobil til skalering og NoSQL
- Typiske oppgaver er knyttet til utprøving eller forbedring av teknologier og metoder, ofte med fokus på åpen kildekode
BEKK - Definering av oppgave

• Oppgave defineres i samarbeid for å ivareta både studentenes faglige interesser og faggruppende behov

• Eksempler på tidligere oppgaver ved NTNU/UiO:
  – Mobile first og offentlige nettsteder
  – Katanoo - towards a highly scalable web analytics tool
  – DSL i Native Java
  – Testdrevet JavaScript
  – Open Source og søketeknologi for virksomheter
Neste skritt

- Send søknad til studentoppgaver@bekk.no og legg ved
  - CV
  - karakterutskrift(er)
  - informasjon om faglige interesser
- Ved spørsmål, ikke nøl med å kontakte BEKKs UiO-ansvarlige Fredrik Valdmanis:
  fredrik.valdmanis@bekk.no
+ oppgaver fra Medisinsk informatikk

- ved Thorbjørn Rognes