Student: João Marco Maciel da Silva
Advisor (USP): Professor Fabio Kon
Student USP ID: 7577598
Institute: Institute of Mathematics and Statistics (IME-USP)
Course: Bachelors in Computer Science
Interchange: Università della Svizzera italiana (University of Lugano)

**Introduction**

This document presents a work plan for a study and research visit to the University of Lugano. João Marco received a fellowship paid by USP to spend 6 months (extendible) at the University of Lugano (from September 2012 or February 2013).

**About João Marco Maciel da Silva**

João Marco is in the 3rd semester of USP's computer science undergraduate course. Before USP, he had taken 3 semesters in electronic engineering at ITA (Aeronautics Technology Institute). João Marco also took some extra curricular courses in OO, web services, web apps, architecture, and design patterns. Besides, He manages and teaches (together with another student from the Bachelors in Mathematics course) a study group in preparation for the math olympiads (specifically undergraduate Brazilian math olympiads (OBMU) and IMC).

Since 2011, João Marco has been working in the Mezuro/Kalibro Project at CCSL-IME/USP (FLOSS Competence Center) under Prof. Fabio Kon's supervision. Mezuro is a plugin for Noosfero (other open source project) to use kalibro in that. Kalibro is a web service where one can select metric collectors (with Analizo and Checkstyle), set ranges with grades, mix them in new compound metrics and use customized configurations in projects hosted in a version control system (e.g., git, bazaar, svn).

**Personal Motivation for the Interchange**

The interchange opportunity will be a great chance for personal development, to experience different cultures, improve English proficiency, learn Italian, improve my research skills, and will lead to for further opportunities in the future.

**Motivation to University of Lugano**

The University of Lugano was listed as a partner from University of São Paulo interchange program. According to João Marco's background and Fabio Kon`s group interests João Marco has selected the University of Lugano to build bridges between Kon`s group and Michele Lanza`s group.

**Objectives with this Interaction**

- Take a set of course focused on Software Engineering (in particular, Software Evolution)
- Work on software visualization
- Collect data from tens of thousands of FLOSS projects and analyzing them
- Learn Michele Lanza's expertise with visualization with our tools and data
- Show to MIchele Lanza's group what Fabio Kon's group is doing at USP
- Find/Define opportunities for future collaborations between Lanza's and Kon's groups.

**Courses**

Here is a list of courses João Marco would like to attend (according to if they will be offered in the semester)

<u>Undergraduate Lectures:</u>

1. Software Development
2. Programming Languages and Software Design
3. Information and Knowledge Management I: DB (DataBases)
4. Software Atelier IV: Software Engineering of Web Applications
5. Net-centric Computing

<u>Graduate Lectures</u>
(If not possible to enroll this courses, just as an audit student):

1. Design 101
2. Software Design and Evolution
3. Software Architecture and Design/LAB
4. Software Quality/LAB

Some are equivalent to USP CS undergraduates course, some are personal interest and some are useful for the project that João Marco works for and his research. They are ordered by preference.

**Research topics**
(for interaction with Professor Michele Lanza`s group)

1. Studying the correlation between source code metrics and open source project activity (committers and commits) using the database of 42 thousand projects from sourceforge (c, c++ and java) that we collected.

2. Mapping the relationships between clean code concepts and source code metrics and study visualization techniques to depict these relationships.

3. Complement statistics results shown in Lanza's book to define intervals for other metrics, e.g., the ones supported by the Analizo tool we developed.

4. Look for possible intervals for C code metrics. We may also use the sourceforge projects database for that.

5. Compare Radu Marinescu's results with ours, evaluate all data in both tools (theirs and ours).

**Schedule**

| **Activities** | **M 1** | **M 2** | **M 3** | **M 4** | **M 5** | **M 6** |
|---|---|---|---|---|---|---|
| Courses (3 or 4 from the above list) | X | X | X | X | X | |
| Research 1 (from the above list) | X | X | | | | |
| Research 2 (from the above list) | | | X | X | X | |
| Writing paper (according to research results) | | | | | X | X |

*** Syllabi**

Undergraduate courses:

1 - Software Development

**Objectives**

The ultimate goal of software development is to learn how to develop a software product of a size that requires the work of a group of software developers.

Detailed objectives include:

- To learn different kinds of software processes and be able to define a process suitable for a medium size project;
- To learn the impact of group organisation and how to organize a small group of software development to optimize productivity and results;
- To learn how to analyse requirements and produce useful specifications;
- To learn how to define and assess the quality of the final product.

2 - Programming Languages and Software Design

**Prerequisites**

Algorithms and Data Structures and Programming Fundamentals II

**Contents**

Type systems, principles of object-oriented programming and design, design by contract, UML, design patterns, refactoring, concurrent system design, software architectures, component models, aspect-oriented programming.

**Objectives**

In this class techniques and tools are taught which help designing complex software systems that are maintainable, reusable, and extensible.

3 - Information and Knowledge Management I: DB (Data Bases)

**Objectives**

Querying, organizing, designing, and managing data collections, ranging from small to very large sizes, is a key aspect of the expertise of computer science professionals. The goal of this course is to introduce the fundamental concepts related to Data Base Management Systems, which constitute the core technology in several ICT applications, as well as in enterprises. The course will introduce data-centric abstractions, models, and languages that are useful in order to design and use databases.

The course moves from initial motivations - justifying the relevance of information systems in modern society - to the essential aspects of data management, which are covered by including both theoretical and practical aspects. At the end of the course, we expect the student to develop basic abilities about how to design and query a database,

and to be informed about the use of such components within information systems architectures.

**Contents**

An introduction to information systems (IS): motivations, users and their roles, examples of classical IS applications. An Introduction to Data Base Management Systems, and their main differences with respect to Search Engines.

The relational data model: structure, properties and constraints, formal query languages (relational algebra, relational calculus, logic programming), comparison of their expressive power. The SQL language: basic definitions and fundamental concepts. The Data Definition Language (DDL) and the Data Manipulation Language (DML). Simple and complex SQL queries. Views. Reasoning on the expressive power of the language and hints on its efficient execution by an SQL engine. Data abstractions, and the conceptual design of a database. The ER model: Entities, Relationships, Attributes, IS-A relations, Generalisations, Cardinalities, Identifiers. Top-down and bottom-up design methods. Mapping of ER schemas to relational schemas. Complete design of a database schema: from requirements, to the conceptual schema, to the relational schema, normalisation.

Issues about technologies of data base servers. Introduction to the problem of Data Quality assessment.

4 - Automata and Formal Languages

**Prerequisites:** Discrete Mathematics I

**Objectives**

The theory of automata and formal languages deals with the problem of modeling computation: what is a computer, and what are its fundamental capabilities?

Thus, it constitutes the basis for further studies on the theory of computability and complexity. Additionally, "Automata and Formal Languages" is a very practical course, as it provides knowledge of the models used in many branches of computer science, from scanners and lexical analyzers in compilers, to programs for designing digital circuits, and even in other areas such as linguistics. At the end of this course you will be very familiar with the main models of computations used today, you will understand how they are fundamental to further studies and you will be ready for a more advanced course on the theory of computation.

**Contents**

The course will cover finite automata, regular languages, regular expressions, deterministic and nondeterministic models, context-free grammars, pushdown automata, turing machines. As an application, we will see Flex and Bison, tools used to generate the front-end part of compilers.

5 - Software Atelier IV: Software Engineering of Web Applications

**Prerequisites**: Software Atelier III

**Co-requisites:** Programming Languages and Software Design

- **Objectives**
- To learn the principles of Web application development;
- To gain practical experience with Web application languages and frameworks;
- To learn the ideas behind Web 2.0 (social software, AJAX);
- To learn to build Web applications for innovative applications.

**Contents**

The fourth semester atelier is about open source web application development. The goal of the course is to learn how to work in a group to build innovative Web applications. The course will begin with 5 weeks of technology tutorial sessions. You will learn and use programming languages such as JavaScript, Scala and/or Ruby; data representation techniques (JSON, YAML, and XML); Web APIs; AJAX. Next, you will be introduced to modern web frameworks, which provide a convenient way of developing web applications. The rest of the semester you will work as a member of a team to create an innovative Web application. You will maintain an up-to-date documentation about your project. You will use different tools such as subversion and TRAC. You will get experience with testing, debugging, and deployment.

6 - Net-centric Computing

Prerequisites: Programming Fundamentals II

**Objectives**

This course introduces key computing concepts above the computer architecture, namely from the operating system up to the middleware. The goals of the course are to understand compex computing systems, both local and distributed, and to develop the knowledge and skills to successfully build this style of application.

**Contents**

This course covers topics in concurrency, synchronisation, scheduling, resource management, and models of distributed computing. We explore the core issues in depth from a theoretical perspective, build an understanding of existing algorithms to solve some of these core issues, and develop practical skills using tools for distributed application development. Detailed topics to be covered include but are not limited to semaphores, monitors, rendezvous, deadlock, client-server, peer to peer, message passing, shared memory, virtual execution environments, component models, and service-oriented architectures. Throughout the course, emphasis will be placed on theory as well as on putting theoretical knowledge to practice.

Graduate Courses (if not possible, just sit in on this classes):

7 - Design 101

**Objectives and Contents**

This course teaches the student the universal principles of design, how to recognize and appreciate the elegance of design present in nature and how this maps to human design in a variety of domains, such as graph design, table design, slide design, etc.

8 - Software Design and Evolution

**Pre-requisites:** Software Engineering

**Objectives**

This course provides students with an overview of design heuristics and puts them in an evolutionary context. It teaches students to design systems to withstand the inevitable decay and using reverse engineering and reengineering techniques, it lets students "see" software as more than just source code.

**Contents**

The Software Life-Cycle; Object-Oriented Design; Responsibility-driven Design; Designing Classes; Encapsulation; Design Patterns; Software Quality; The Law of Demeter; Software Performance; Software Metrics; Design Heuristics and Style Guidelines; Detection Strategies & Design Disharmonies; User Interface Design; Software Evolution; Reverse Engineering; Reengineering Patterns; Visualization.

9 - Software Architecture and Design/LAB

**Pre-requisites:** Software Engineering

**Objectives**

This class teaches the students to structure complex software systems using components and to keep track of the rationale behind their design. Architecture is not only necessary as the global blueprint to guide the design and manage the complexity of large software systems, but should also be seen as the focus of the main design decisions influencing the properties (modularity, maintainability, extensibility, portability, interoperability, reuse, performance) of the resulting system.

**Contents**

Introduction and Motivation: Why Software Architecture?; System Decomposition vs. Software Composition; Design Principles: Simplicity, Abstraction, Separation of Concerns, Encapsulation, Information Hiding; Patterns and Anti-Patterns: Avoiding Common Design Mistakes; Component Models and Composition Techniques; Architectural Views: Logical, Physical, Process, Development; Architectural Styles:

Monolithic, Layered, Client/Server, REST, Service oriented, Peer-to-Peer, Event-driven, Data-centric, Pipes/Filters, Space based, Plugin; Architectural Description Languages; Architectural Decision Modeling; API Design Techniques.

10 - Software Quality/LAB

**Pre-requisites:** Software Engineering

**Objectives**

The main goal of this course is to understand and master the different aspect of software quality. Students will learn quality goals and tradeoff, different testing and analysis techniques, and how to plan and monitor the quality process.

**Contents**

The course assumes basic knowledge of software test and analysis problems and techniques, classic functional testing techniques, classic structural testing techniques, inspection techniques, and classic testing tools. The course presents a detailed framework of software quality, describes core techniques, discusses problems and solutions, and provides a detailed description of the software quality process.

The main topics are:

- Quality activities, tradeoff and limitations
- Basic principles
- Finite models
- Dependence and data flow models
- Symbolic execution and proof of properties
- Finite state verification
- Data flow testing
- Model based testing
- Testing object oriented software
- Fault based testing
- Test execution (scaffolding and oracles)
- Program analysis
- Planning and monitoring the quality process
- Integration and component-based software testing
- Acceptance and System testing
- Regression testing
- Documentation and standards
- Test automation

Aluno: João Marco Maciel da Silva
Orientador: Professor Fabio Kon
NUSP: 7577598
Instituto: Instituto de Matemática e Estatística (IME-USP)
Curso: Bacharelado em Ciência da Computação
Intercâmbio: Università della Svizzera italiana (University of Lugano)

**Introdução**

Este documento apresenta um plano de trabalho/estudo para aplicação à Universidade de Lugano. João Marco conseguiu uma bolsa de intercâmbio pela USP para passar 6 meses (extendível) na Universidade de Lugano (a partir de setembro de 2012 ou Fevereiro de 2013).

**Sobre João Marco Maciel da Silva**

João Marco está no terceiro semestre do curso de graduação do bacharelado em Ciência da Computação da USP.
Antes da USP, Ele cursou 3 semestres em engenharia eletrônica no ITA (Instituto Tecnológico de Aeronáutica). João Marco também fez cursos extra curriculares em OO, web services, web apps, arquitetura e padrões de design.
Além disso, ele gerencia e orienta (em conjunto com outro aluno do bacharelado em matemática) um grupo de estudos para o olimpíadas de matemática (especificamente Olimpíada Brasileira de Matemática Universitária(OBMU) e IMO).

Desde 2011, João Marco tem trabalhado no projeto Mezuro/Kalibro no CCSL-IME/USP (Centro de Competência em Software Livre) sob a supervisão do Professor Fabio Kon. Mezuro é um plugin para Noosfero (outro projeto open source) para usar a kalibro nele. Kalibro atualmente é um web service onde você pode selecionar coletores de métricas de código (atualmente Analizo e Checkstyle), escolher escalas com notas, misturá-las com novas métricas compostas e usar suas configurações em projetos hospedados em serviços de controle de versão (e.g., git, bazaar, svn).

**Motivação Pessoal para Intercâmbio**

A oportunidade de intercâmbio será uma ótima chance para o desenvolvimento pessoal, conhecer diferentes culturas, melhorar o inglês, aprender italiano e será uma alavanca para outras oportunidades.

**Motivação para Lugano**

University of Lugano estava listada como universidade parceira da USP no programa de intercâmbio por mérito acadêmico. De acordo com o background de João Marco e o interesse do grupo do Fabio Kon João Marco selecionou a University of Lugano para interação entre o grupo de pesquisa do Fabio Kon e o grupo de pesquisa do Professor Michele Lanza.

**Objetivos com esta Interação**

Trabalho em visualização de software;
coletar dados de dezenas de milhares de projetos Open Source e analizá-los;
aprender com a especialidade do Professor Michele Lanza com visualização usando nossas ferramentas e dados;
mostrar para o grupo do Michele Lanza o que o grupo do Fabio Kon tem feito na USP;
achar/definir oportunidades de futuras colaborações entre os dois grupos.

**Lista das disciplinas que João Marco pretende cursar**
(De acordo com o oferecimento delas no semestre)

Disciplinas de Graduação:
- Software Development
- Programming Languages and Software Design
- Automata and Formal Languages
- Information and Knowledge Management I: DB (DataBases)
- Software Atelier IV: Software Engineering of Web Applications
- Net-centric Computing

Disciplinas de Pós-graduação (Se não for possível se matricular nestes, assistir como ouvinte):
- Design 101
- Software Design and Evolution
- Software Architecture and Design/LAB
- Software Quality/LAB

Alguns São equivalentes em comparação com as do curso de bacharelado em ciência da computação da USP, outras são interesses pessoais e outras são úteis para o projeto e pesquisa em que João Marco trabalha. As disciplinas estão ordenadas por ordem de preferência.

**Tópicos de pesquisa** (para interação com o grupo do Professor Michele Lanza)

1- Estudar correlação entre métricas de código fonte e atividade de projetos open source (committers and commits, desenvolvedores e desenvolvimento) usando o banco de dados de 42 mil projetos do sourceforge (c, c++ and java) que nós coletamos.

2- Mapear as relações entre conceitos de código limpo e métricas de código fonte e técnicas de visualização para descrevê estas relações.

3- Complementar os resultados estatísticos mostrados no livro do professor Michele Lanza para definir outros intervalos de métricas, e.g., as suportadas pela ferramenta Analizo que desenvolvemos.

4- Procurar possíveis intervalos de métricas para código em C. Nós podemos utilizar o banco de dados de projetos do sourceforge para isto.

5- Comparar os resultados de Radu Marinescu com os nossos, avaliar os dados em ambas as ferramentas (deles e nossas). Esta ideia foi sugerida pelo próprio Marinescu quando conversou com Antonio Terceiro, em abril passado.

**Cronograma**

| **Atividades** | **M 1** | **M 2** | **M 3** | **M 4** | **M 5** | **M 6** |
|---|---|---|---|---|---|---|
| Cursos (3 ou 4 da lista acima) | X | X | X | X | X | |
| Pesquisa 1 (da lista acima) | X | X | | | | |
| Pesquisa 2 (da lista acima) | | | X | X | X | |
| Escrever artigo (de acordo com os resultados da pesquisa) | | | | | X | X |

Student: João Marco Maciel da Silva
Advisor (USP): Professor Fabio Kon
Student USP ID: 7577598
Passport ID: FG122752
Institute: Institute of Mathematics and Statistics (IME-USP)
Course: Bachelors in Computer Science
Interchange: Università della Svizzera italiana (University of Lugano)

**Introduction**

This document presents a work plan for a study and research visit to the University of Lugano. I received a fellowship paid by USP to spend 6 months (extendible) at the University of Lugano (from September 2012 or February 2013).

**About Me**

I am in the 3rd semester of USP's computer science undergraduate course.
Before USP, I had taken 3 semesters in electronic engineering at ITA (Aeronautics Technology Institute). I also took some extra curricular courses in OO, web services, web apps, architecture, and design patterns. Besides, I manage and teach (together with another student from the Bachelors in Mathematics course) a study group in preparation for the math olympiads (specifically undergraduate Brazilian math olympiads (OBMU) and IMC).

Since 2011, I have been working in the Mezuro/Kalibro Project at CCSL-IME/USP (FLOSS Competence Center) under Prof. Fabio Kon's supervision. Mezuro is a plugin for Noosfero (other open source project) to use kalibro in that. Kalibro is a web service where one can select metric collectors (with Analizo and Checkstyle), set ranges with grades, mix them in new compound metrics and use customized configurations in projects hosted in a version control system (e.g., git, bazaar, svn).

**Personal Motivation for the Interchange**

The interchange opportunity will be a great chance for personal development, to experience different cultures, improve English proficiency, learn Italian, improve my research skills, and will lead to for further opportunities in the future.

**Motivation to University of Lugano**

The University of Lugano was listed as a partner from University of São Paulo interchange program. According to my background and Fabio Kon`s group interests I have selected the University of Lugano to build bridges between Kon`s group and Michele Lanza`s group.

**Objectives with this Interaction**

- Take a set of course focused on Software Engineering (in particular, Software Evolution)
- Work on software visualization
- Collect data from tens of thousands of FLOSS projects and analyzing them
- Learn Michele Lanza's expertise with visualization with our tools and data
- Show to MIchele Lanza's group what Fabio Kon's group is doing at USP
- Find/Define opportunities for future collaborations between Lanza's and Kon's groups.

**Commitment**

I declare that I will come back to Brazil at the end of the enrolled courses.

_____

João Marco Maciel da Silva