

Fast / Software Engineering



Prof . Dr. Michel Chaudron
Chalmers/Gothenburg Univ.
Leiden Univ.

Supervisor



Hafeez Osman
Phd candidate
Software Design Recovery



Ramin Etemaadi
Phd candidate
Software Architecture
Optimization



Ana Fernandez Saez
Phd candidate
Software Maintenance

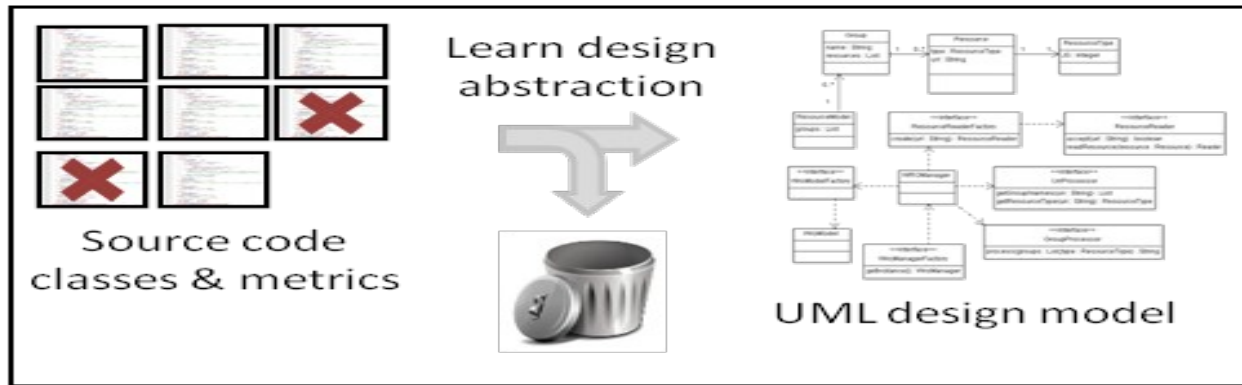


Bilal Karasneh
Phd candidate
Software Design Metrics



Dave Stikkolorum
Phd candidate
Software Design Education

Project : Exploring feature extraction for class diagram abstraction



Supervisor	Hafeez Osman
LIACS:	Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	aislimau@yahoo.com hosman@liacs.nl chaudron@liacs.nl
Room	150

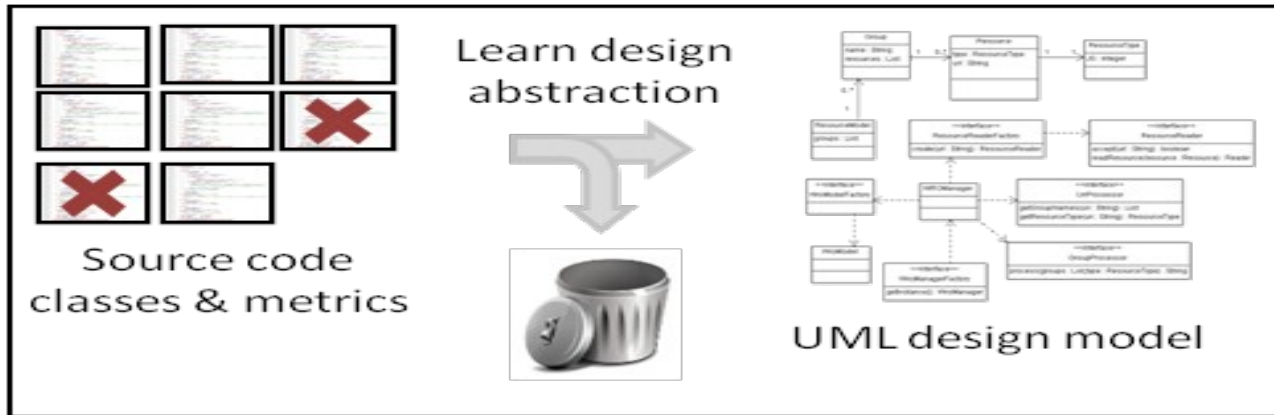
Task: Exploring feature extraction method for condensing reverse engineered UML class diagram.

Tool Infrastructure: WEKA

Summary: There is a range of techniques available to reverse engineer software designs from source code. However, these approaches generate highly detailed representations. Reverse engineered class diagrams are typically a detailed representation of the underlying source code. This makes it hard for software engineers to understand what the key elements in the software structure. The condensing of reverse engineered representations into more high-level design information would enhance the understandability of reverse engineered diagrams.

Prerequisites (helpful, not mandatory): Data mining Feature Extraction/Identification
Programming skill

Project : UML Class Diagram Abstraction



Supervisor	Hafeez Osman
LIACS:	Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	aislimau@yahoo.com hosman@liacs.nl chaudron@liacs.nl
Room	150

Task : Visualizing UML Class Diagram with abstraction feature

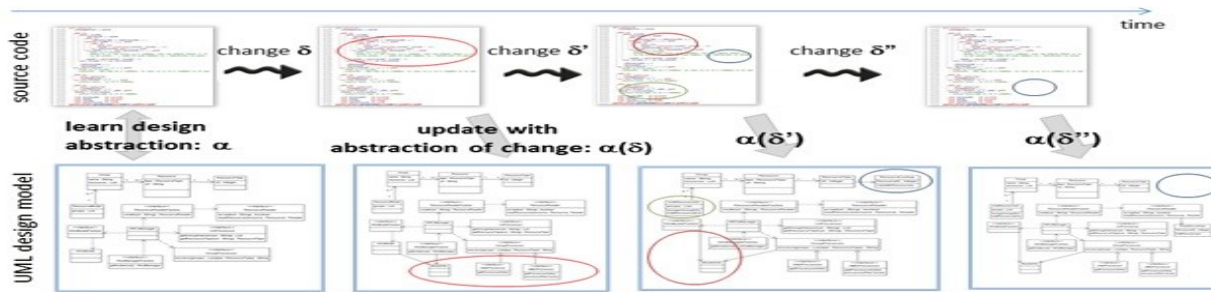
Tools Infrastructure : GUI (NClass), Web services

Summary: Unified Modeling Language (UML) is a widely used modeling language in the software industry. However, it is well known that up-to-date design documentation is important in later stages of development and especially also in the maintenance phases. One popular technique in recovering the design is reverse engineering. Nevertheless, current reverse engineering techniques do not adequately solve this problem. This project specifically aims at reconstructing the class diagrams from source code in such a way that unnecessary detail that results from reverse engineering is eliminated.

Prerequisites (helpful, not mandatory): C#, Java, *Creative*

Final Product: User interface for UML class diagram abstraction tool

Project :UML Class Diagram - Automated Synchronization based on Source Code Artifacts Evolution



Supervisor	Hafeez Osman
LIACS:	Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	aislimau@yahoo.com hosman@liacs.nl chaudron@liacs.nl
Room	150

Aim : Recovering and abstract all design for every system versions.

Tools Infrastructure : .Class Abstraction Tool

Summary: Maintaining correspondence (between design and implementation) is particularly challenging because over time an implementation tends to evolve considerably from its initial design. Design models produced during the design phase are often forgotten during the implementation phase-under time pressure usually-and thus present major discrepancies with their actual implementation frequently. Tool support during maintenance, re-engineering or re-architecting activities has become important to decrease the time that software personnel spends on manual source code analysis and helps to focus attention on important program understanding issues.

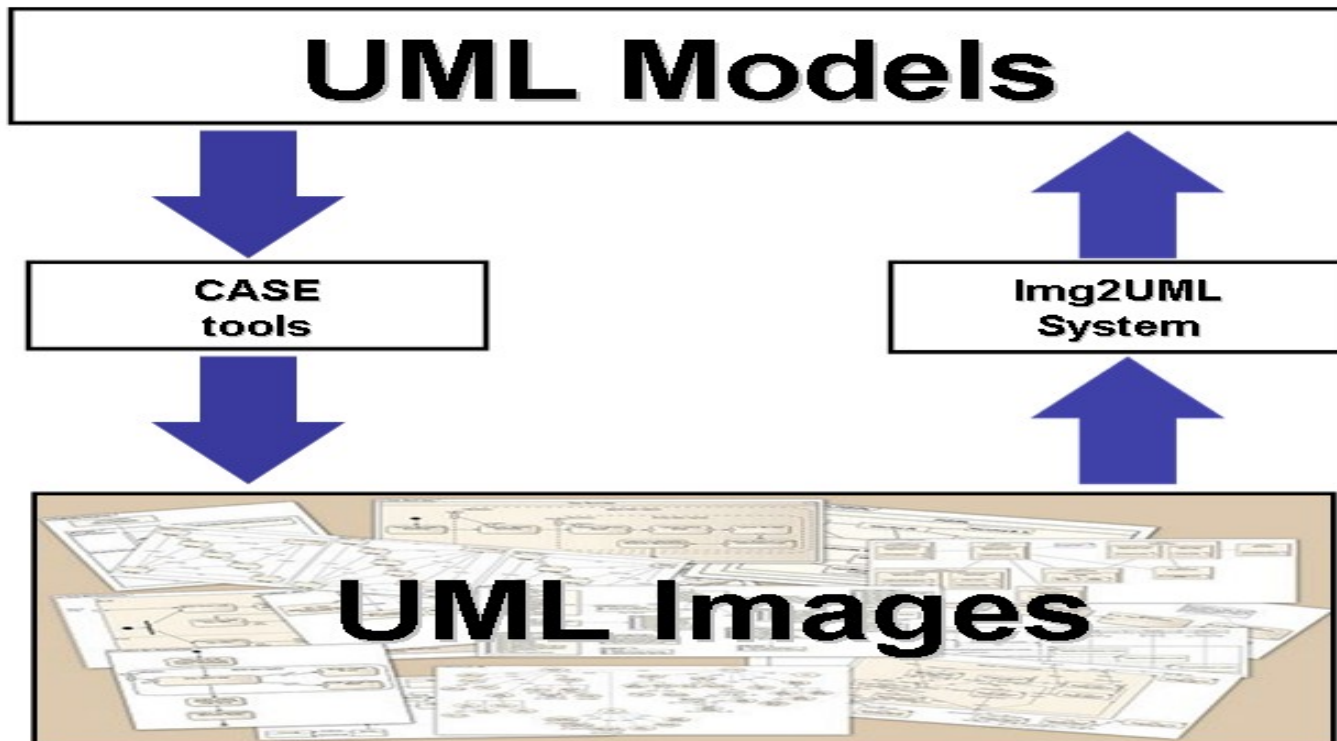
Prerequisites (helpful, not mandatory): Programming Skills, UML

Final Product: UML class diagram recovery tool.

Project: Online Img2UML

- **Overview:**

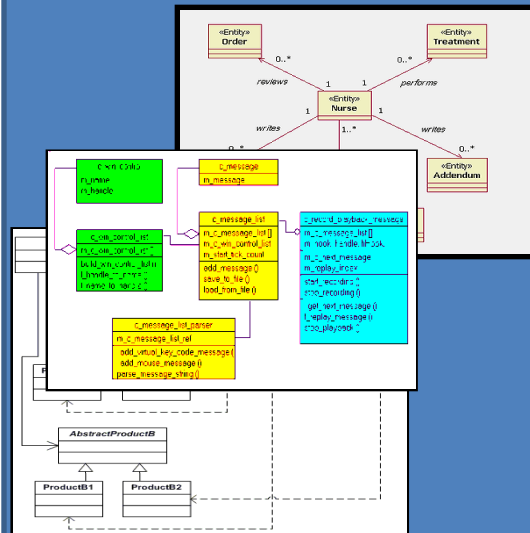
- *Img2UML System:*



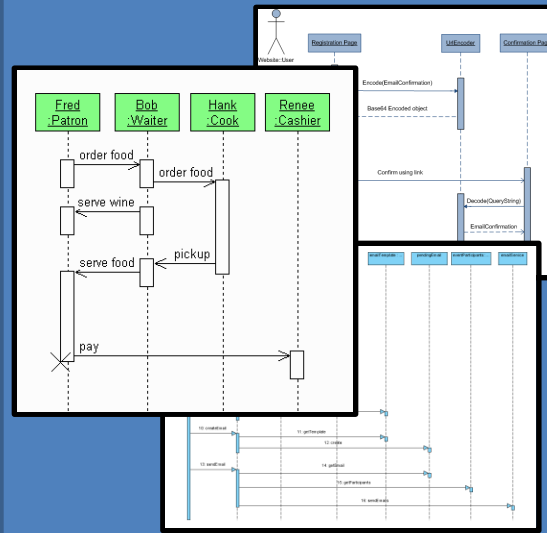
Project: Online Img2UML

- **Overview:**
 - *Img2UML System:*

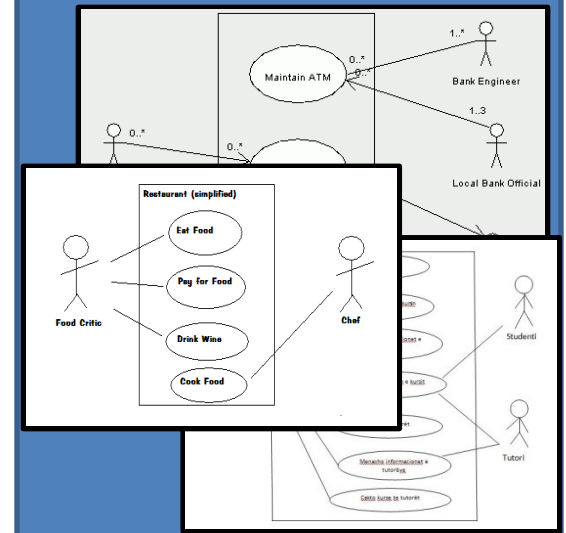
VB.net



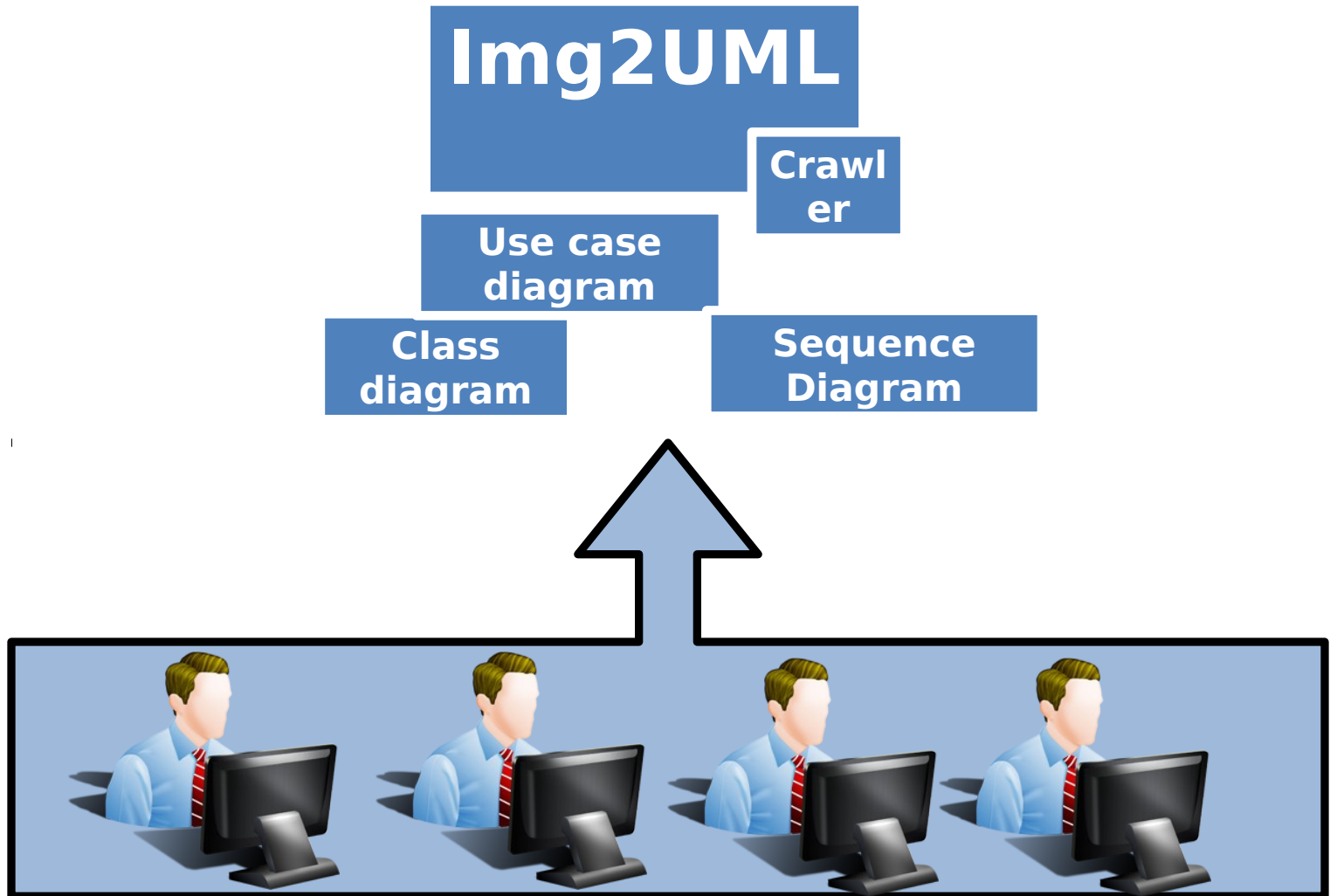
C#



C++



Project: Online Img2UML



Project: Online Img2UML

- **Wanted:**

- **Make a web interface for Img2UML system.**

- **Requirement**

- **Web programming ..**

Contacts:

Bilal Karasneh: bkarasne@liacs.nl, bkarasne@gmail.com

Dave Stikkolorum: drstikkolorum@liacs.leidenuniv.nl

Experiment / tool plug-in : Software Design Behavior of students (and professionals)



Supervisor LIACS:	Dave Stikkolorum Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	d.r.stikkolorum@liacs.leidenuniv.nl chaudron@liacs.nl
Room	150

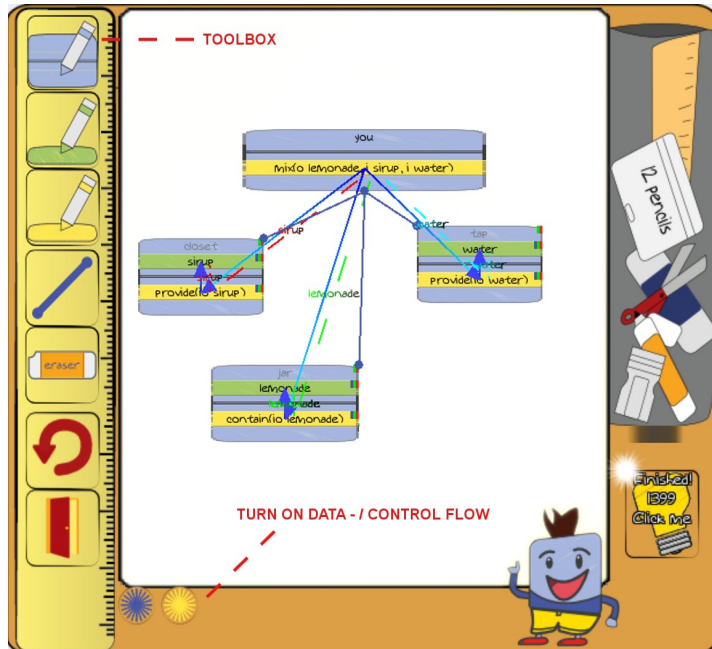
Task: Conduct an experiment that analyses the behavior of students when they perform a software design task using a modeling tool with a home-made plug-in.

Tool: Visual Paradigm (VP)

Summary: We want to explore the behavior of students when they perform a software design task in a modeling tool. In particular we want to identify typical subtasks a student performs and maybe profile students. We are interested in what kind of differences exist between professionals and novices. In order to monitor this behavior we have to develop a plug-in for VP. A proof of concept of this plug-in is already available.

Prerequisites (helpful, not mandatory): Java / OO programming, statistics

Web port of Edu Game 'Art of Software Design'



Supervisor LIACS:	Dave Stikkolorum Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	d.r.stikkolorum@liacs.leidenuniv.nl chaudron@liacs.nl
Room	150

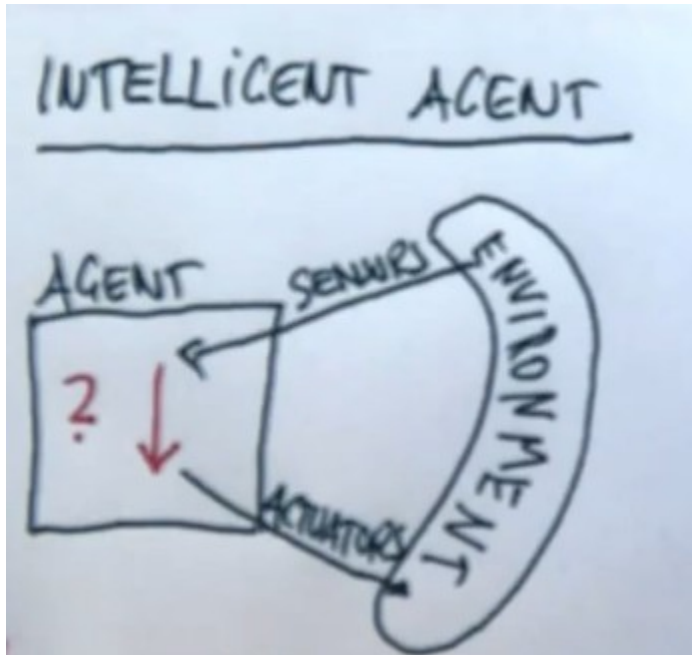
Task: Redesign and implement a web based educational game for learning software design principles and patterns

Prerequisites (helpful, not mandatory): Web languages like HTML(5), Javascript, software design principles and - patterns

Summary: In 2012 our game 'the Art of Software Design' was rewarded at the Models2012 conference gamification contest. Momentarily the game is a proof of concept and has to be developed further. We want to have the game available for a large number of students all over the world and, as much as possible, cross platform.

Additional info : <http://aosd.host22.com/>

Feedback (teachable) agent for: 'the Art of Software Design'



Supervisor LIACS:	Dave Stikkolorum Prof Dr. Michel Chaudron
Start date:	Immediately
Contact:	d.r.stikkolorum@liacs.leidenuniv.nl chaudron@liacs.nl
Room	150

Task: Design and implement a (teachable) agent that give feedback to a student while learning in 'the Art of Software Design'

Prerequisites (helpful, not mandatory):
Web languages like HTML(5), Javascript, data structures, algorithms, design patterns

Summary: giving feedback during the learning process helps students to improve their knowledge about the subject they are studying. For educational software feedback can be of big value. In a game format feedback can improve the gameplay. We want to include the possibility of feedback on our game 'the Art of Software Design' in the form of an ai (teachable) agent.

Additional info : <http://aosd.host22.com/> - <http://www.teachableagents.org/>