Gaudí Software Factory

- It is a place to build good software and to find the best way to build good software
- Organized as a software production factory
- Part of CREST (Center for Reliable Software Technology)
- Financed by Akademi of Finland, TEKES, Åbo Akademi, and TUCS
- Managed and run by TUCS Software Construction Lab
Problem

- How to build reliable and maintainable software in a research environment
  - Software is part of research project
  - Project has limited aims and time span
  - Students as programmers (M.Sc, Ph.D)
  - Turnover is big, programmers disappear after exam
  - →
  - Difficult to build and maintain large software systems
  - No proper software process, ad hoc methods used
  - Build mostly prototypes and demonstrators
Attempt at a solution: Gaudi

- Build a unit inside the research center, which functions as
  - A software factory, for producing software products following a well-defined software process
  - As a software laboratory, for carrying out experiments in software engineering
- The factory produces software needed by research groups
- The laboratory studies/improves the processes used in the factory
Gaudi software factory

CREST

Gaudí Software Factory  Research Laboratories

Software products

Tasks, product requests

Articles, tools, research results
Objectives for the Software factory

- Apply both established and new software processes
- Software processes should be
  - Flexible
  - Efficient and easy to learn
  - Lead to reliable software
  - Applicable in academy and in the industry
  - Incremental, evolutionary
Focus on Product Development

- The main activity is **product development**
- A product should be constructed following some specific practices
  - Extreme Programming, SFI, UML, …
- Programmers are supervised by a coach
  - Programmers are undergraduate students
  - Coach is a PhD. student
- There is a customer that has the last word on the product
  - In principle a researcher (who pays for the programming project)
  - Can also be industrial projects
  - Require active customer participation
Gaudi software laboratory

CREST

Gaudí Software Factory

Software products

Tasks, requests

Research Laboratories

Results of experiments

Software experiments

Software Construccion lab

Articles, tools, research results
Objectives for Software laboratory

- A framework for practical experiments in software engineering
  - Evaluate and try out software methods and processes
  - Carry out controlled experiments
  - Monitor experiments
  - Quantitative and qualitative evaluation of results
  - Use experimental results to improve on the software process
A software experiment

- A software product development is an experiment
- Product can be new, or it can be extension of an existing product
Experiments in Gaudi

- Limited size of experiment:
  - 4-6 programmers
  - 3-6 months time
  - full time or half time employment
  - generally 1 manyear/experiment
  - undergraduate students

- Clearly defined goals for product
  - no research, just build product
  - no educational aspects for students
  - no study credits for work done
Gaudi software process

- Extreme programming (XP) as basic process
- XP contains a lot of different methods
  - Some are compulsory in Gaudi
  - Some are being tried out, or adapted
- Complement process with other methods
  - Stepwise feature introduction, as architectural principle
  - Design by contract
  - UML as design language
  - Flexible programming language (Python)
Software produced, examples

- **Math Editor**
  - Shows a proof as an outline, checks the correctness of derivations, suggests applicable rules

- **Software Construction Workbench**
  - A tool to model and develop object-oriented software using Stepwise Feature Introduction methodology

- **Software Modeling Workbench**
  - A collection of tools to create and transform software models
Activities

- Summer 2001: Pilot test
  - 3 months
  - 4 ÅA students, 2 HUT students
  - One product

- Summer 2002
  - 4 months
  - 12 students (8 ÅA, 4 TY)
  - Three products

- Autumn 2002
  - 3 monts
  - 12 students
  - Three products

- Spring 2003
  - 3 months
  - 3 half-time projects
  - 12 students (11 ÅA, 1 TY)

- Summer 2003:
  - 3 months
  - 20 students
  - 6 coaches
  - 2 lab technicians
  - five products

- Summer 2004:
  - 6 months
  - One industrial project
  - 8 students

- Autumn 2004
  - 3 monts
  - 4 students
  - External project

- Spring 2004
  - 3 months
  - Industria project

- Summer 2005
  - 6 months
  - 8 students
  - One industrial
Math Editor

This is a sample file with some rules and expressions to try out derivations with. Also, try export File, Export LaTeX and have a look at the results. Pretty nifty, huh? It can be configured in Options, LaTeX Export Conversions.

Here are the rules:

- \( x < x + 1 \)
- \( x \geq 1 \rightarrow x \leq x^x \) for all \( x \)
- \( x ! = 0 \rightarrow x \text{ div } x = 1 \)
- \( x + y = 8 \)(\(x + y\)) for all \( x, y \)
- for all \( x, y \exists \) for \( x + y = x + y \)

L6 C5 Ins Rule less-than-rule
Stepwise Feature Introduction

- Software is constructed in thin layers
- Each layer
  - adds a new feature
  - is a working product
  - can be thoroughly tested and verified independently of the other layers
- Objective: Maximize flexibility, reliability and maintainability

<table>
<thead>
<tr>
<th>Week Term View</th>
<th>Week View</th>
<th>Term Model</th>
<th>Term View</th>
<th>Term Model</th>
</tr>
</thead>
<tbody>
<tr>
<td>ImageLayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageApplication</td>
<td>ImageView</td>
<td>ImageFileChooser</td>
<td>ImageFilter</td>
<td>ImageLabel</td>
</tr>
<tr>
<td>MenuLayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MenuApplication</td>
<td>MenuView</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TermLayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>TermApplication</td>
<td>TermView</td>
<td>TermModel</td>
<td>TermModel</td>
<td>Term</td>
</tr>
<tr>
<td></td>
<td>TermSelectDialog</td>
<td>TermDialog</td>
<td>TermOverlapDialog</td>
<td></td>
</tr>
<tr>
<td>WeekLayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>WeekApplication</td>
<td>WeekView</td>
<td>WeekModel</td>
<td>WeekModel</td>
<td>Week</td>
</tr>
<tr>
<td>ImageHelperLayer</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>ImageTestApplication</td>
<td>ImageAdapter</td>
<td>ImageInfo</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
```java
package mypackage;

public class TrafficLight {
    // Constructor

    public void initialize() {
        // Initialize components
    }

    public void cycle() {
        // Cycle through states
    }

    public void handleInput(byte input) {
        // Handle input from sensors
    }
}
```

# Define classes
- `Road` (represents a road segment)
- `Timer` (controls the timing of the traffic lights)
- `Green`, `Red`, `Yellow` (representing different light states)

# Define behaviors
- Transition between light states based on input
- Handle events such as button presses

```java
// Behavior of class TrafficLight

/* Behavior of class TrafficLight */

print("*** START ***");
print("*** END ***");
```

# Define processes
- `initialize`, `cycle`, `handleInput` methods
- `print` for logging

```java
// Define EVENT_initial

/* Behavior of class TrafficLight */

public void initialize() {
    // Initialize components
}

public void cycle() {
    // Cycle through states
}

public void handleInput(byte input) {
    // Handle input from sensors
}
```

Gaudí Facilities

- 280m² in Datacity B
  - Rooms furnished for team work
  - Large common area
- Own Equipment
  - Computers, bookshelf
- Support personnel

Datacity B, 4th floor
Experiences - 1

- The approach taken in Gaudi works well in practice
  - The factory produces maintainable software products of sufficiently high quality for research needs
  - It provides a good environment for experimenting with different software methods and practices
  - The results of the experiments are being used to improve the software process
Experiences - 2

- Mostly favourable experiences from using XP
  - Works well in a university setting
  - Students love it
  - Good experiences from using the XP customer model
- But XP needs to be complemented with
  - Overall software architecture design
  - Need to improve documentation
Spanish architect, one of the most creative architects in modern times. His style is often described as a blend of neo-Gothic and Art Nouveau, but it also has elements of Surrealism and Cubism.

Antoni Gaudí i Cornet 1852-1926
Gaudi future plans

Academic Gaudi

Industrial Gaudi

Experimental Gaudi