Xtext
Mentor: Sebastian Zarnekow (Sebastian.Zarnekow@itemis.de)
Min: 6 students. Max: 16 students

What is this?
Xtext (www.xtext.org) is the leading framework to develop domain specific languages (DSLs) and programming languages. It provides all the necessary means for the entire tool stack starting with a grammar definition, the linking infrastructure and the compiler up to decent IDE support for Eclipse, IntelliJ IDEA and also for Web based editors.
Xtext is widely used in academia and industry projects on all scales.

In many projects, Xtext DSLs are a key factor to a high level of code quality, productivity and expressiveness and therefore crucial to the success of the project. Working on Xtext means that you shape the tools for software developers and end users alike. Together with a team of smart, passionate and enthusiastic coders you'll work on one of the most innovative Open Source projects.

What is an example of what I can expect to learn/do?
Languages that are implemented with Xtext, will get extensible IDE support for IntelliJ and Eclipse essentially for free. However, domain specific languages may also target user groups that don't have a technical background and may be scared off by these tools. Also some programmers prefer lightweight editors such as vi or emacs.

To benefit from the advantages of domain specific languages in heterogeneous teams and for non-technical users, we want to provide plugins for editors like Sublime, Github Atom, Visual Studio Code, notepad++ or others. Syntax coloring, code completion, live validation and incremental compilation for Xtext languages should be made available for users of these editors, too.

Recommended Student Background
- Good Java Knowledge
- Familiar with unit testing (JUnit)
- Dependency Injection
- Comfortable when working with Java IDEs (e.g. Eclipse, IntelliJ)
The Eclipse Integrated Computational Environment (ICE)

Mentor: Jay Jay Billings (tzikis@codebender.cc)
Min: 5 students. Max: 10 students

What is this?
Eclipse ICE is a new Eclipse project focused on bringing the power of the platform to high-performance modeling and simulation. Most computational scientists do *everything* with command line tools and Vi or Emacs. We have developed Eclipse ICE as an alternative to give them a fast, modern workbench that lets them create input files, launch local and remote jobs on supercomputers, analyze and visualize results in 3D, and manage their data - all from Eclipse!

Eclipse ICE is currently deployed with simulators for nuclear reactors, quantum computers, batteries, and advanced materials, among others. Each of these projects provides plugins to the platform that makes it possible to do tasks far faster and precisely than was previously possible with the command line tools alone.

Finally, science is cool and we have cookies! Come through! ;-)

What is an example of what I can expect to learn/do?
ICE is broken into several pieces, including the UI, the "Core" that provides critical services, the developer plugins for different simulators and our data structures. Students will focus on re-developing and extending ICE's UI to make it highly customizable by developers.

Each data structure in ICE is rendered in a different way, depending on what is in the object. This context-specific rendering makes it easy for developers to get custom behavior without coding it and for users to get widgets they expect. It isn't always possible to detect how widgets should be drawn and sometimes very specific custom rendering is required. Students will help the ICE team modify the UI so that it uses Eclipse 4's dependency injection mechanism to inject highly customized widgets, giving developers the ultimate say in how their data is rendered if they so choose.

Recommended Student Background
Students should know Java.

Each student will be provided with extensive documentation on the Eclipse RCP and Eclipse ICE's API. Some of this will include books that we will mail to the students at our (ORNL's) expense.