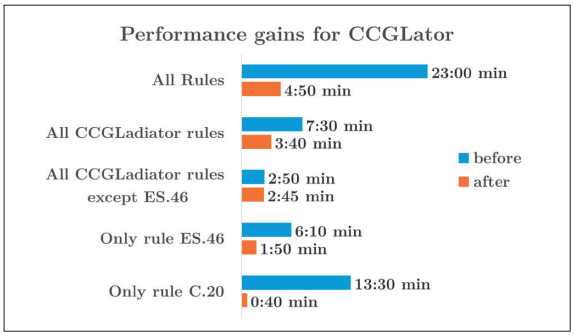
Rolf Bislin

Kilian Diener

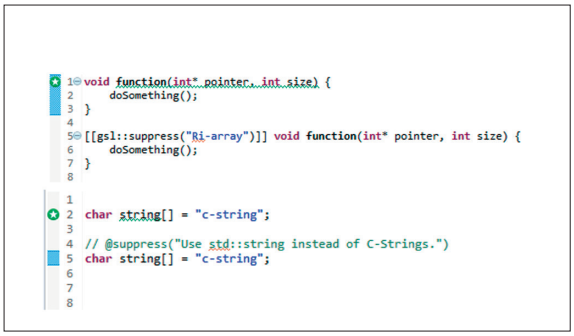
Graduate Candidates	Rolf Bislin, Kilian Diener
Examiner	Thomas Corbat
Co-Examiner	Lukas Felber, Quatico Solutions AG, Zürich, ZH
Subject Area	Software Engineering – Core Systems
Project Partner	Institute for Software, Rapperswil, SG

CCGLImperator

C++ Core Guidelines Plug-in for Cevelop



Runtime of the checked CCGLator rules before and after the improvements



```
void function(int* pointer, int size) {
    doSomething();
}

[[gsl::suppress("Ri-array")]] void function(int* pointer, int size) {
    doSomething();
}


char string[] = "c-string";

// @suppress("Use std::string instead of C-Strings.")
char string[] = "c-string";
```

Comments and attributes can be used to suppress specific plug-in rules

Introduction: The C++ Core Guidelines are a set of rules to encourage the use of modern C++, which is a simpler and safer subset of the language. In this bachelor thesis the already existing Cevelop plug-ins, CCGLator and GslAtorPtr, have been improved and extended. CCGLator supports programmers using C++ Core Guidelines in everyday programming. It was developed in two preceding term projects. GslAtorPtr was also created in a previous project and focuses on rules concerning the correct use of pointers.

Approach/Technologies: First, CCGLator has been tested on a real-world project to see how it performs on a large codebase. Based on this assessment several improvements have been implemented and errors have been fixed. Then the work was split into two parts. One was the addition of new rules to CCGLator and the other was the improvement of GslAtorPtr. Because the two plug-ins are very similar, the architectures of both could be compared during the implementation and flaws in either have been eliminated, resulting in a cleaner and more similar structure. Additionally, new insights were gained which resulted in improvements to both plug-ins.

Result: CCGLator now has better performance, reports fewer false positives and supports five additional rules. GslAtorPtr handles a wider range of function interfaces and supports more specific types in the quick fixes. Also, the added possibility to set attributes or comments to ignore specific rules provides convenient and coherent handling of ignoring rules in both plug-ins. Additionally, a lot of work was put into the infrastructure of both plug-ins to clean up the code and make the plug-ins perform better in general. Furthermore, a contribution to the Eclipse CDT project was made, which has already been merged into the project and is bound to be published in the next release.

«ES.50: Don't cast away const» checker and quick-fixes in action