ITEA
INFORMATION TECHNOLOGY FOR EUROPEAN ADVANCEMENT

PROJECT RESULTS

Open-source compiler extended

New GCC infrastructure cuts cost of quality assurance

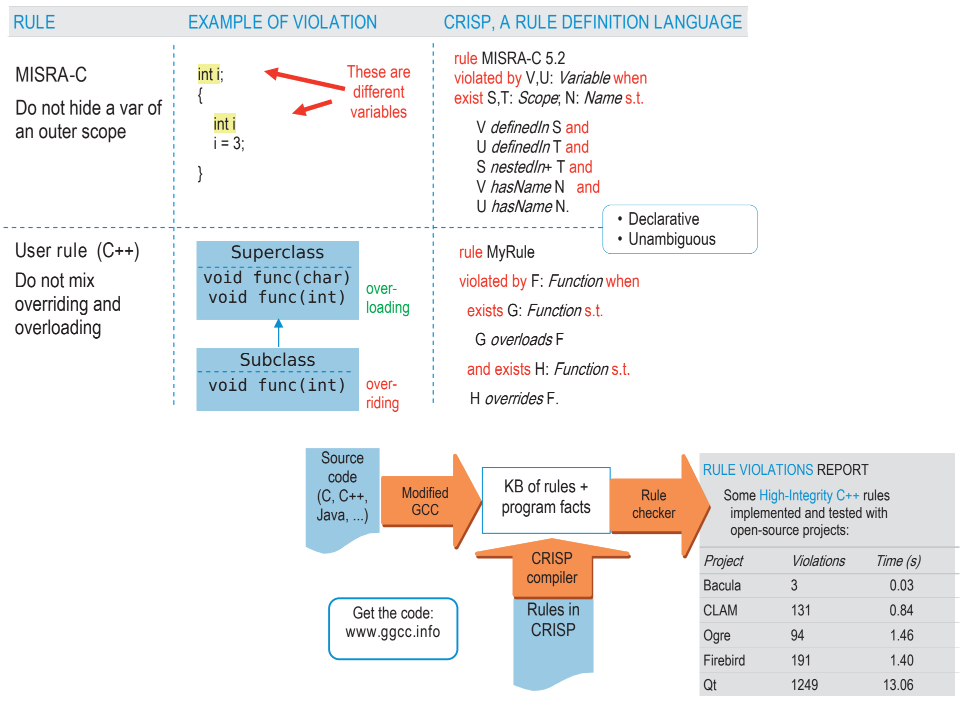
The GGCC project has added a new branch to the widely-used global GNU compiler (GCC) open-source compiler for Unix-like operating systems. Its capabilities allow global processing and optimisation of complete programs and libraries – including static analysis for early bug fixing. The expertise acquired could help Europe to profit from a range of valuable business opportunities.

Explosive growth in the software content of digital systems, coupled with increasing expectations of trouble-free operation, create escalating problems for program developers. Integration of components and validation of complete packages now consumes 70 to 90% of total development costs.

Large corporations spend heavily on proprietary systems for the most mission-critical applications in sectors such as aerospace. But even here, the investment required to achieve zero-defect quality can prove ruinous. Providers of enterprise, consumer and embedded software cannot usually afford the same levels of expenditure. Yet, while quality assurance remains a heavy burden, cutting corners may lead to brand-damaging numbers of faults in shipped products.

Building on international standard
Open-source tools are increasingly

CODING RULE CHECKING


RULE
EXAMPLE OF VIOLATION
CRISP, A RULE DEFINITION LANGUAGE
MISRA-C
Do not hide a var of an outer scope
These are different variables
User rule (C++)
Do not mix overriding and overloading
Superclass
Subclass
over-loading
over-riding
• Declarative
• Unambiguous
Source code (C, C++, Java, ...)
Modified GCC
KB of rules + program facts
Rule checker
CRISP compiler
Rules in CRISP
Get the code: www.ggcc.info

```
int i;
{
  int i
  i = 3;
}
```

```
rule MISRA-C 5.2
violated by V,U: Variable when
exist S,T: Scope; N: Name s.t.
  V definedIn S and
  U definedIn T and
  S nestedIn+ T and
  V hasName N  and
  U hasName N.
```

```
Superclass
void func(char)
void func(int)

Subclass
void func(int)
```

```
rule MyRule
violated by F: Function when
 exists G: Function s.t.
  G overloads F
 and exists H: Function s.t.
  H overrides F.
```

RULE VIOLATIONS REPORT

Some High-Integrity C++ rules implemented and tested with open-source projects:

Project	*Violations*	*Time (s)*
Bacula	3	0.03
CLAM	131	0.84
Ogre	94	1.46
Firebird	191	1.40
Qt	1249	13.06

GGCC (ITEA 05012)

Partners
AIRBUS France
AQUILINE
ANSWARE
BASE
BERTIN
CEA-LIST
IDI EIKON
INRIA-Futurs
Mandriva
MySQL A.B.
SICS
SQS
Syxoo
Telefónica I+D
UPM

Countries involved
France
Spain
Sweden

Start of the project
September 2006

End of the project
March 2009

PROJECT RESULTS

employed to resolve this dilemma. GCC, for example, has become the standard C/C++/Fortran/Ada/Java compiler for most modern Unix-like operating systems. This highly portable compiler suite offers exceptional hardware and vendor independence as it is able to generate code for almost every current instruction set architecture. With a massive user base and contributions from hundreds of professional developers, mainly in the USA, its content has grown tenfold over the past decade.

The principal objectives of GCC are code generation quality and compilation speed. Despite its huge expansion, an aspect that long remained unexplored was global static analysis and coding-rule checking for debugging and project-wide optimisation. Global analysis makes it possible to process complete compilation units such as programs or libraries together, and to manage the relationships between them. Coding-rule validation ensures programs conform to rules established for a particular industry, company or application. Cumulatively, they further improve coding efficiency, enhance program performance and reduce testing times, cutting costs and time to market.

Earlier work focused on academic prototypes or closed commercial products such as Klocwork and PolySpace. GGCC broke new ground by incorporating them into a popular mainstream platform as a basis for new tools and interfaces applicable across many industries.

Notable innovations
GCC's size, complexity and dynamic nature, plus the strict rules imposed by the GCC community, made this a daunting task. The SME-led consortium nevertheless achieved significant progress, developing a common framework to accommodate three main functions: analysis and optimisation, hazard detection and code validation.

GCC is written in C, which is not ideal for abstract interpretation – that is being independent of the system in which it is being implemented. It was therefore necessary to integrate a higher level language. GGCC opted for Lisp and developed a middle-end Lisp-to-C translator (MELT), which has been adopted as an official branch of the GCC development tree.

Within the project, the first simple static analysers were realised, as was a mature version of the global optimisation framework bringing greatly enhanced performance and tighter integration with the GCC internal application program interfaces. This is now on par with proprietary compilers on most targets, while preserving the advantages of retargetability, multi-language capability and openness.

Add-ons include Coding Rules in Sugared Prolog (CRISP), a logic-based domain-specific language making it possible to express coding rules easily even without relevant expertise.

GGCC successfully tested a prototype coding rule checker at full scale on Nokia's C++ Qt cross-platform library. The number of rules remains limited, but the results offer a promising indication of code quality

Europe can benefit
Many of the improvements have already helped GCC to become a leading platform for global optimisation projects that go way beyond the capabilities of competing compilers. By acquiring leading-edge expertise, the partners have created an opportunity for Europe to establish a strong position in an area with considerable potential.

ITEA 2 Office
High Tech Campus 69 - 3
5656 AG Eindhoven
The Netherlands
Tel : +31 88 003 6136
Fax : +31 88 003 6130
Email : itea2@itea2.org
Web : www.itea2.org

October 2009