



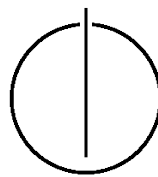
FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Informatics

Adding C++ Support to MBEDDR

Zaur Molotnikov





FAKULTÄT FÜR INFORMATIK

DER TECHNISCHEN UNIVERSITÄT MÜNCHEN

Master Thesis in Informatics

Adding C++ Support to MBEDDR

C++ Unterstützung für MBEDDR

Author: Zaur Molotnikov
Supervisor: Dr. Bernhard Schätz
Advisor: Dr. Daniel Ratiu
Date: September 16, 2013



Ich versichere, dass ich diese Diplomarbeit selbständig verfasst und nur die angegebenen Quellen und Hilfsmittel verwendet habe.

München, den 16. September 2013

Zaur Molotnikov

Acknowledgments

If someone contributed to the thesis... might be good to thank them here.

Abstract

An abstracts abstracts the thesis!

Contents

Acknowledgements	vii
Abstract	ix
1 Introduction	1
2 Foundations	3
3 Technologies in Use	5
4 Projectional C++ Implementation	7
5 Evaluation	9
6 Conclusion	11
Appendix	15
Glossary	15
Bibliography	17

1 Introduction

In embedded programming the C++ programming language is widely spread, [3]. Being a general purpose programming language, C++ does not provide any special support for an embedded systems programmer.

By changing the language itself, together with a tool set for it, it is possible to get a better environment for specifically embedded programming. As an example, a subset of C++, called Embedded C++ can be brought, [2]. The approach taken in Embedded C++, namely, stripping very many core features of C++ off, allows for a higher degree of optimizations by compiler possible. Simpler C++ was intended to ([1]) allow higher software quality through better understanding of the limited C++ by programmers, higher quality of compilers, better suitability for the embedded domain. This approach, however, has been criticized by the C++ community ([4]), specifically for the inability of the limited language to take advantage of the C++ standard library, which requires the C++ language features, absent in Embedded C++.

Another approach to modify a language, to get it more suitable for the embedded development, is the opposite to the one, taken in Embedded C++. It consists of extending the language with constructions specific to the domain. Such approach is taken, for example, in the MBEDDR Project ([5]), for C programming language. As an example of extensions to C language state machines and decision tables can be brought.

Additionally, the integrated development environment (Integrated Development Environment, e.g. MS Visual Studio, Eclipse, etc. (IDE)) can be improved to support domain specific development. Various analyses can be built in into the code editor in order to detect inconsistencies, or simply “dangerous” constructs. Certain code formatting, or standard requirements could be enforced, and many more.

A mixture of all three techniques could be combined in an attempt to achieve a “better” C++ for embedded development. A special IDE can be created together with a new C++ language flavor, which prohibits the most “dangerous” C++ constructs, and allows for modular creation of extensions.

Before the chance to create extensions to the C++ language, the C++ language limited flavor itself together with a special IDE have to be created. This is the problem to be solved in this Master Thesis.

2 Foundations

3 Technologies in Use

JetBrains MPS is used.

4 Projectional C++ Implementation

5 Evaluation

6 Conclusion

Appendix

Glossary

Embedded C++ is a language subset of the C++ programming language, intended to support embedded software development. 1

IDE Integrated Development Environment, e.g. MS Visual Studio, Eclipse, etc.. 1

JetBrains MPS is a language engineering environment allowing to construct incrementally defined domain specific languages. 5

MBEDDR Project is a JetBrains MPS based language workbench, representing C language and domain specific extensions for the embedded software development. 1

Bibliography

- [1] Embedded C++. Objectives behind limiting c++,
<http://www.caravan.net/ec2plus/objectives/ppt/ec2ppt03.html>.
- [2] Embedded C++. Official website, <http://www.caravan.net/ec2plus/>.
- [3] VDC Research. Survey on embedded programming languages,
http://blog.vdcresearch.com/embedded_sw/2010/09/what-languages-do-you-use-to-develop-software.html.
- [4] Bjarne Stroustrup. Quote on embedded c++, http://www.stroustrup.com/bs_faq.html#ec++.
- [5] Markus Voelter, Daniel Ratiu, Bernhard Schätz, and Bernd Kolb. mbeddr: an extensible c-based programming language and ide for embedded systems. In *SPLASH*, pages 121–140, 2012.