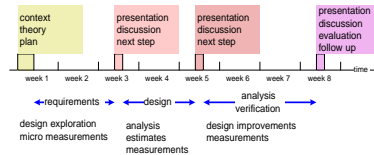


# Course Execution Architecture

-



Gerrit Muller

Embedded Systems Institute

Den Dolech 2 (Laplace Building 0.10) P.O. Box 513, 5600 MB Eindhoven The Netherlands

[gerrit.muller@embeddedsystems.nl](mailto:gerrit.muller@embeddedsystems.nl)

## Abstract

The course Execution Architecture (EA) is described. The program existing of 2 modules and 3 feedback and plan sessions is described. The course format, based mostly on hands on work in real products being created, is explained.

The course execution architecture is a joint effort of Ton Kostelijk and Gerrit Muller.

### Distribution

This article or presentation is written as part of the Gaudí project. The Gaudí project philosophy is to improve by obtaining frequent feedback. Frequent feedback is pursued by an open creation process. This document is published as intermediate or nearly mature version to get feedback. Further distribution is allowed as long as the document remains complete and unchanged.

All Gaudí documents are available at:  
<http://www.gaudisite.nl/>

version: 0

status: planned

February 10, 2011

# 1 Introduction

The architect designing an embedded systems has to address many different stakeholder concerns and design aspects. For some reason SW design tends to focus on functionality. Performance and real time behavior are the (arbitrary) outcome, mostly determined by the execution architecture.

This course focuses on fulfilling performance and real time requirements by designing an execution architecture. The main target of the course are the practical day to day issues of designing an execution architecture. The course is for a great deal hands on work, in the normal product environment on the normal product. During the course the participants should analyse, measure, explore and try requirements (problem side) and implementations (solution side). On a regular base the results are presented and reviewed, then a follow up plan for the next period is made and discussed.

A minimum of 2 persons per team are needed, to stimulate and enable the participants to make sufficient progress in the limited course time frame. Those 2 persons together should be able to cover the understanding of the most critical product needs, the existing hardware design (and its opportunities and constraints) and the existing software design. On top of that they should be able to modify the software design and run the modified design, to measure and to obtain results.

## 2 Program

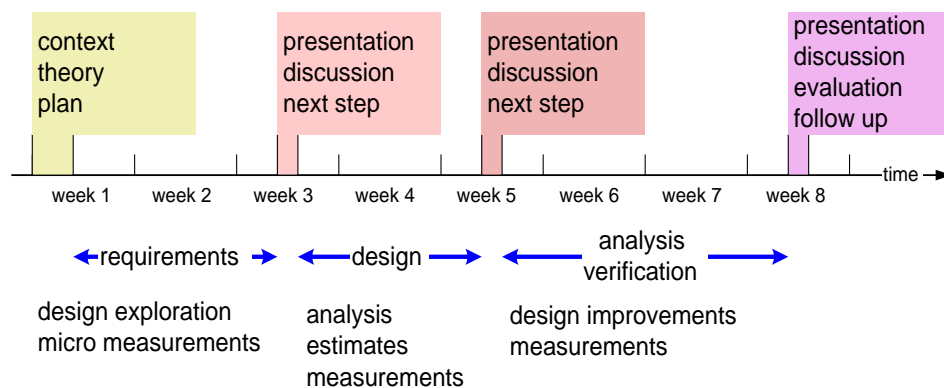


Figure 1: Program of the Execution Architecture Course

## 3 Rules during the course

The rules of the broadcast part are:

- Please write your questions/remarks/statements on yellow stickers and attach them at the end on the P-flip.

*These will be used in the interactive section for discussion and to increase insight.*

- Short clarification questions are welcome,  
*Discussion will take place in the interactive part.*
- Stupid questions don't exist. Learning is based on **safe** and **open** interaction.  
*Very individual-oriented questions can be referred to a break or after the session.*

The rules of the interactive and the practice part are:

- Your contribution is essential.
- Don't monopolize the time. Everyone, also the quiet people, should have the opportunity to contribute.  
*The facilitator will intervene if the contribution is limited to a small group of participants.*
- Respect the contribution of others.  
*Opinions can't be wrong, difference of opinion is normal and called plurality.*
- The course format is highly experimental and based on improvisation, constructive proposals are welcome.  
*It is your course! Regular evaluations will give the opportunity to influence the rest of the course.*

## 4 Evaluations

Basic part of learning is the evaluation of what has been done. The course will use 3 types of evaluations:

- Personal expectations
- Benefit and Concerns on a regular base
- The CTT evaluation form

The personal expectations are recorded at the very beginning of the course. At the end we look back at these initial expectations. This has a two-way evaluation effect:

**Personal** Did you start with the right expectation level? Was it realistic? Did you achieve the learning goals formulated in this expectation?

**Trainer and CTT** Did we communicate the right information to enable people to select this course? Do we apply the right selection criteria?

The benefit and concern evaluation method is based on the basic feedback method, which prescribes to start with formulating the strong points, before addressing the weaker issues. The idea is that improvement is based on building on the strong points and to change with respect to the weaker issues. A side effect is that everyone is forced to think also about the positive aspects, not only about the negative.

The benefit and concern evaluation is done regular, in the beginning with a high frequency, to be able to adapt the course directly.

The benefits and concerns are collected by a brainstorm or on yellow stickers. The rule is that one should always start with a benefit before mentioning a concern.

The benefit and concern method is widely used by CAP Gemini employees, often called B&C or Beer&Chips.

The CTT evaluation form is the "standard" CTT evaluation form which evaluates the different aspects of the course.

## 5 Acknowledgements

This course is a joint effort of Ton Kostelijk and Gerrit Muller, making it together has been very inspiring.

## References

- [1] Gerrit Muller. Requirements capturing by the system architect. <http://www.gaudisite.nl/RequirementsPaper.pdf>, 1999.
- [2] Gerrit Muller. The system architecture homepage. <http://www.gaudisite.nl/index.html>, 1999.
- [3] Gerrit Muller. Architectural reasoning explained. <http://www.gaudisite.nl/ArchitecturalReasoningBook.pdf>, 2002.

## History

**Version: 0, date: August 29, 2002 changed by: Gerrit Muller**

- document created, no changelog yet