International Journal of Advances in Computer Science & Its Applications – IJCSIA 2018
Copyright © Institute of Research Engineers and Doctors , SEEK Digital Library
Volume 8 : Issue 2- [ISSN : 2250-3765] - Publication Date: 28 December, 2018

Development of Embedded Systems - from a Practical Point of View

Wolfgang Fengler
Maxi Weichenhain

***Abstract*—The paper describes the development of embedded systems using a W process model as a model-based top-down design from function to structure, partitioning and implementation or realization. Early testing via simulation is an integral component. The development is integrated in the process management. All explained steps build on an example from precision weighing technology.**

***Keywords*—embedded system, design process, weighing machine, process management**

I. Introduction

The following article is mainly about the development of the application soft- and hardware as part of an embedded system. This essentially realizes the required functions regarding control and information processing for a technical embedding system, whereby both form a unit. The quality of development process is decisive for the quality of the product as well as for the development effort. In many cases, engineers are acting as the developers and embedded system and embedding system are developed in parallel and mutual influence. In more complex systems the implementation platform consists of a computer core, programmable logic (FPGA, field-programmable gate array) and classic hardware, whereby the limits between the four possibilities are partly variable. The development of software and FPGA implementation have a lot in common. Following a process model (W model, [1]) developed under the author's direction is used and applied for a meaningful real prototypical example.

II. Example weighing machine

A precision weighing machine (scale) is a sufficiently complex system to explain the method. It is a prototype with a special digitizing process that has been known for a long time (Fig. 1, [2]). The weight force acts via an appropriate mechanism on a quartz glass spring, which changes the angle relative to a fixed plate and, with appropriate lighting, produces movable interference stiffeners (light and dark stripes). These move in one direction when loaded and in the other when relieved. Photoelectronic scanning produces two pulse sequences (Fig. 2, x1, x2) which contain the relative change in the weight force (continuous flanks) and its direction (due to phase shift of x1 and x2).

Wolfgang Fengler
Technische Universität Ilmenau
Germany

Maxi Weichenhain
Technische Universität Ilmenau
Germany

The information processing must generate forward (fw) and backward signals (bw) as shown in Fig. 2, which then lead to the calculation of a weight corrected several times.

Dynamic correction is necessary. Fig. 3 shows possible value progressions of the intermediate values of the weight to be calculated based on the behaviour of the mechanical

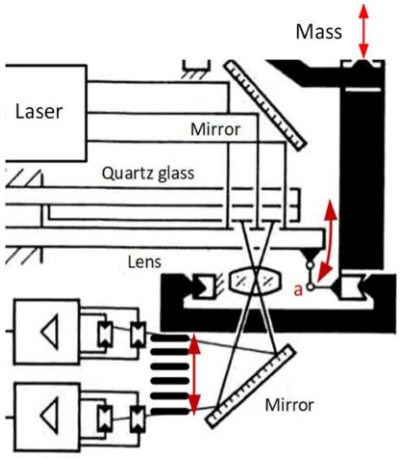

Figure 1: Weighing Machine: Prototype Schematic (partly from [2])

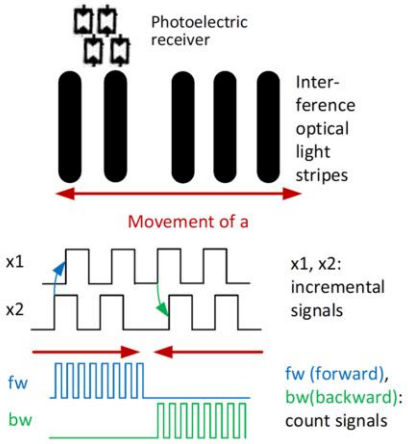

Figure 2: Weighing Machine: Signals

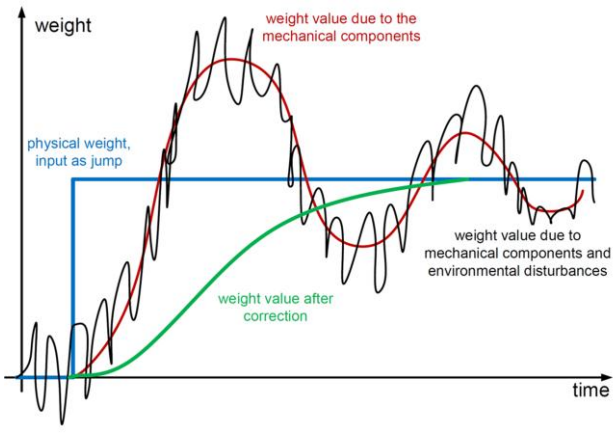

Figure 3: Weight Value After Several Corrections

components and environmental disturbances. The part of the information processing that generates the weight after the correction is used below for explanation.

III. The Design Process

The W process (Fig. 4) according to [1] is used, which is an extension of the known V process [3]. It is a top-down process that begins with the requirements specification followed by a functional design. The design is completed by the subsequent structural design with the corresponding partitioning in embedding system, software, programmable logic (FPGA (Field Programmable Gate Array)) and classical hardware. During design, model-based work is carried out at various abstraction levels, with early testing (left W-side, "simulation-driven design" [4]) when executable models are used, which has a positive effect on the development time. Between the abstraction levels, the operations "refinement", "coarsening" are necessary and formal verification is possible. On the right side of the W, integration and test bottom-up take place, whereby the test cases from the model-based simulations are the starting point.

For the function-oriented levels, the hierarchical design is indicated in Fig. 5. A generalized function block representation is used, similar to that used in Simulink® [5]. It is a hierarchical data flow graph with additional information in the function blocks.

Function block: rectangle

Content:

- **Function** is what shall the block do (symbolic name).
- **Restrictions** are what shall the block have to do not, or some addition values for the realise of the function which are not direct combined with the behaviour. For instance for a hardware block: electrical power consumption.
- **Subfunctions** are the functions that are created when refining the current function.
- **Input, Output** (both of blocks) are the connections between blocks with type (data type, not only types from programming languages, possibly hierarchical combined). All are directed.
- **Refinement**: Replace blocks with hierarchically subordinate blocks and connect their inputs and outputs to each other and/or to inputs and outputs of the higher-level block.
- **Coarsening**: Combining blocks and connecting their inputs and outputs to each other and/or to inputs and outputs of the resulting hierarchically superior block.

For the function-oriented levels of the example the hierarchical design is indicated in Fig. 6 and 7. The generalized function block representation is used. The refinement of the Digital information processing block (Fig. 6) is shown:

Three sub-blocks, these operate interfaces to the outside, possibly refined (here parameters) and additional internal connections.

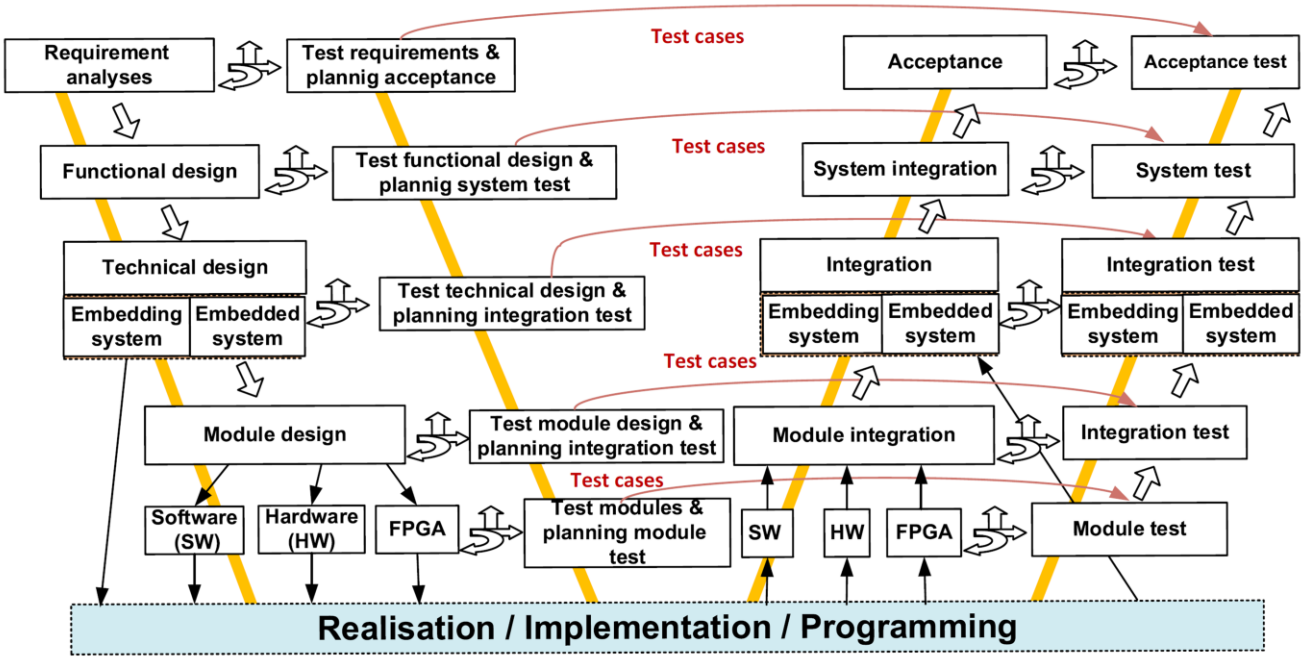

Figure 4: W-Process Model (from [7])

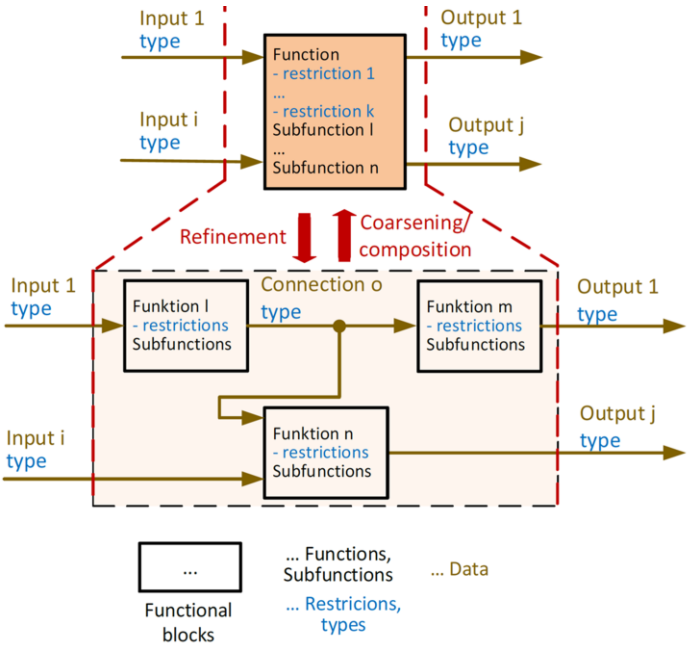

Figure 5: Function Blocks and Various Abstraction Levels

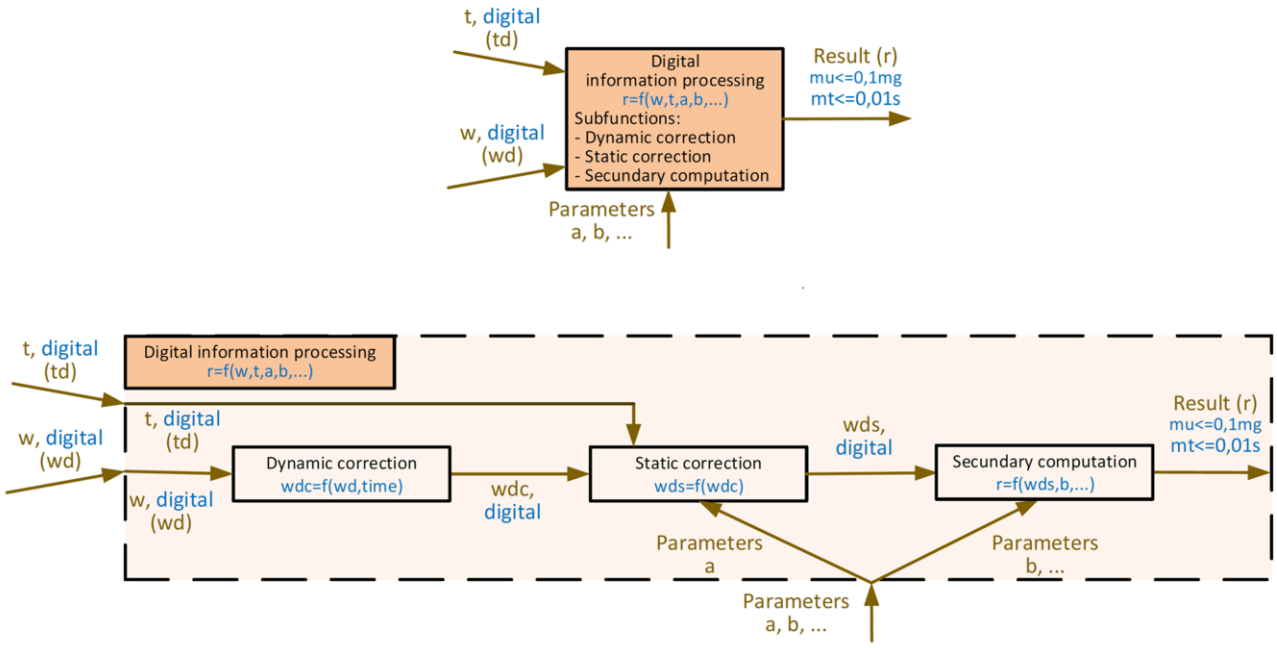

Figure 6: Design Example Digital Information Processing, Higher and Middle Level

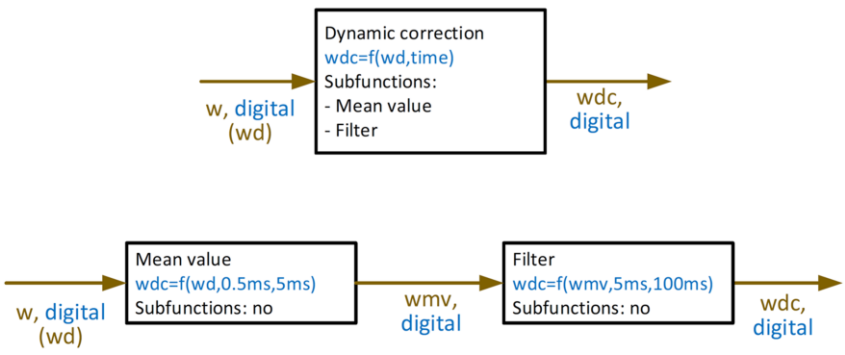

Figure 7: Design Example Dynamic Correction, Middle and Lower Level

It follows the refinement of block Dynamic correction (Fig. 7) that performs the correction of the behaviour of Fig. 3:

- Block Mean value: for the attenuation of smaller but higher-frequency ambient disturbances. Its output becomes input of the following:

- Filter: digital filter.

Both are atomic blocks, i.e. they are directly implemented, e.g. as library functions, but also design decisions are necessary: e.g. which filter type, which parameters etc. At the transition from the function blocks to the structure blocks (left side of the W, atomic blocks, to these a model is possible, which can be implemented directly as software or as hardware) the former are described with a model of the required function. This is shown by means of an example from the measurement logic, which serves to convert the signals fw, bw from x1 and x2 (both Fig. 2). A statechart function block is used for this (e.g. as in Stateflow® [6]). The corresponding statechart (f_b_discriminator) is shown in Fig. 8.

This can be realized by a C++ program (excerpt):

```
void f_b_discriminator::state_change()
{   const bool x1 = input_obj->sensor_read(0);
    const bool x2 = input_obj->sensor_read(1);

    //input

    if (((m_state == 0) & !x1 & x2) || ((m_state == 1)
    & x1 & x2) || ((m_state == 2) & !x1 & !x2) ||
    ((m_state == 3) & x1 & !x2))
            {
             m_output = false; //forward
             f_b_counter_obj->count(m_output);
            }

    if (((m_state == 0) & x1 & !x2) || ((m_state == 1)
    & !x1 & !x2) || ((m_state == 2) & x1 & x2) ||
    ((m_state == 3) & !x1 & x2))
            {
             m_output = true; //backward
             f_b_counter_obj->count(m_output);
            }

    //next_state_function

    if (!x1 & !x2) m_state=0;
    if (!x1 & x2) m_state=1;
    if (x1 & !x2) m_state=2;
    if (x1 & x2) m_state=3;
}
```

In a similar way the statechart can be implemented alternatively via a hardware description language (e.g. VHDL, Very High Speed Integrated Circuit Hardware Description Language) and FPGA.

IV. Project Management

The W process and its model represent the project implementation phase of the entire development project. (Fig. 9). The surrounding blocks in the illustration are specialized in weighing content, such as certain metrological characteristics in the project definition such as:

- measurement accuracy and reliability or speed of measurement.

The project management takes place parallel to the actual procedure according to the W model. Scheduling, monitoring or resource allocation are examples for important cross-connections. Especially if the individual development paths run in different organizational units after partitioning in an embedding system, software, programmable logic and classic hardware, suitable forms of synchronization and communication must be integrated, which above all reflect the mutual influences and possible repartitions.

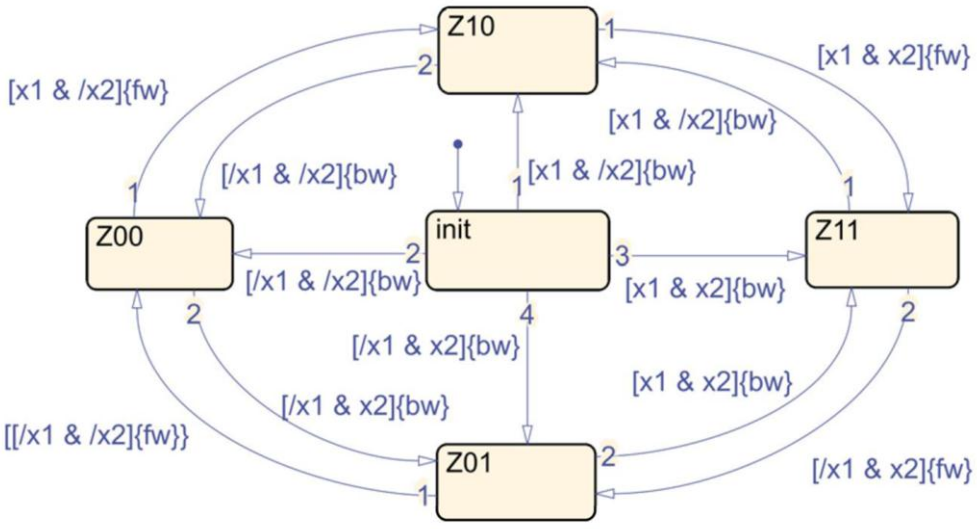

Figure 8: State Chart for Converting x1, x2 to fw, bw (see Fig. 2)

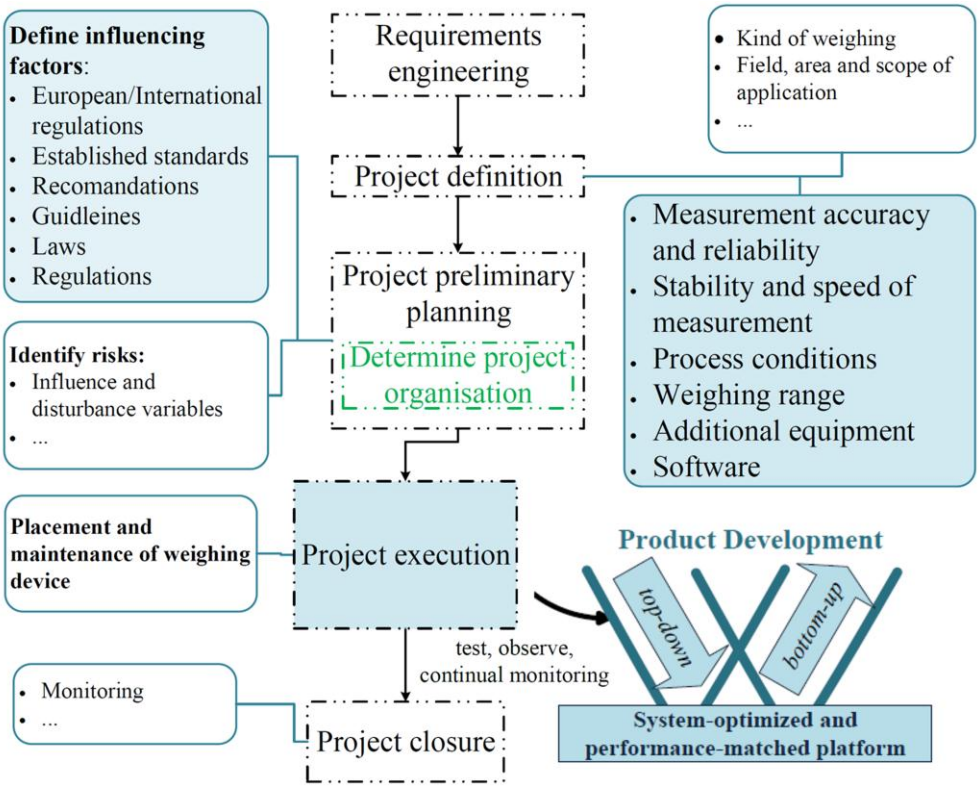

Figure 9: Project Management and W Process (from [7])

v. Summarization

It was shown that a W process enables an effective systematic approach in the development of an embedded system. A hierarchical model-based top-down design is useful, in which the models are simulated in parallel. A meaningful partitioning in embedding system, software, programmable logic and classical hardware leads to partial models, which are implemented and realized directly. The simulation test cases are reused for integration and testing according to the bottom-up principle.

Important design steps have been shown for an example from the precision measuring technique (here weighing technology).

Effective project execution according to the W process takes place in a project management that takes into account the essential aspects of the application area.

References

[1] I. Gushchina, W. Fengler, J. Ciemala, D. Streitferdt, Certifiable development process for information processing in the measurement domain on the base of 3W-models, Embedded Systems Design, 37th IEEE Annual Computer Software and Applications Conference, pp. 685-690

[2] G. Jäger, Elektronische interferentielle Kraftsensoren und ihre Anwendung in der Wägetechnik (Electronic Interferential Force Sensors and their Application in Weighing Technology, in German), Feingerätetechnik, Berlin, 32 (1983), pp. 243-245

[3] B. Boehm, Guidelines for Verifying and Validating Software Requirements and Design Specifications, Euro IFIP, P.A. Samet (editor), North-Holland Publishing Company, IFIP,1979, pp. 711-719

[4] D. Grüner, T. Baumann, B. Däne, W. Fengler, Simulation Driven Development of Complex Systems: Foundations for Simulation Based Design Workflows, Modelling and Simulation 2016 (ESM'2016), EUROSIS-ETI Publ., Ostend, Belgium, 2016, pp. 413-419

[5] Mathworks: Simulation and Model-Based Design, https://ch.mathworks.com/products/simulink.html , released on May 24, 2018

[6] Mathworks: Model and simulate decision logic using state machines and flow charts, https://ch.mathworks.com/products/stateflow.html, released on May 24, 2018

[7] I. Kaiser, W. Fengler, M. Weichenhain, Development of Embedded Systems in the Domain of Measurement Technics, Autonomous Systems 2017, Proceedings of the 10th GI Conference, pp 201-213

About Author (s):

Wolfgang Fengler,
university professor for computer architecture and embedded systems,
Maxi Weichenhein,
phd student,
Technische Universität Ilmenau