



Evolution Management and Process for
Real-Time Embedded Software Systems

Essentials and requisites for the management of evolution Part 1: Overview

Deliverable D1 part 1
Edited by Peter Kaiser

31 March 2003
Version 1.0
Status final

Public Version



I T E A

INFORMATION TECHNOLOGY

FOR EUROPEAN ADVANCEMENT

This document is part of the work of the EUREKA Σ! 2023 – ITEA 00103 project EMPRESS.
Copyright © 2002-2003 EMPRESS consortium.

Authors/Partners:

Partner	Author	Email
Fraunhofer IESE, Germany	Peter Kaiser	Kaiser@iese.fhg.de

Document History:

Date	Version	Editor	Description
31 Mar 03	1.0	Stefan Van Baelen	Public version based on internal version 1.0

Filename: D1.1_v1.0_Public_Version.doc

Abstract

EMPRESS will provide innovative methods and tools to cope with the management of evolution of embedded systems. As a foundation for coming work, an overview on the current state-of-the-art, i.e. the current knowledge in the academia, and the current state-of-the-practice, i.e. the current processes and methods applied in the industry, is given.

This is the first of five parts of this deliverable. It gives an introduction on the project by stating current problems, the EMPRESS goals tackling the problems, the benefit of applying the new methods, and the way to achieve the goals (the work packages).

Keywords:

EMPRESS, embedded software, current problems

Table of Contents

Abstract	3
Table of Contents	4
1 Introduction	5
2 Overview on EMPRESS	5
2.1 Problem.....	5
2.1.1 Automotive Domain	6
2.1.2 Avionics Domain	6
2.1.3 Consumer Electronics Domain	6
2.2 Idea	6
2.3 Benefit.....	7
2.4 Action	8
3 Conclusion	8

1 Introduction

1 Introduction

EMPRESS will provide innovative methods and tools to cope with the management of evolution of embedded systems. As a foundation for coming work, an overview on the current state-of-the-art, i.e. the current knowledge in the academia, and the current state-of-the-practice, i.e. the current processes and methods applied in the industry, is given.

This deliverable consists of five parts:

- Part 1 (this document) gives an overview on the project and summarizes the most important industry needs and research questions;
- Part 2 describes state-of-the-art, state-of-the-practice, public projects, industry needs and research questions in the area of “Evolution of the Component Architecture” (this corresponds to WP 2 of EMPRESS);
- Part 3 focuses on “Evolution of Requirements and Incremental Validation” (this corresponds to WP 3 of EMPRESS);
- Part 4 focuses on “A Formal Base for Evolution” (this corresponds to WP 2 of EMPRESS);
- Part 5 provides a comprehensive glossary for the EMPRESS project.

2 Overview on EMPRESS

In this chapter an overview on EMPRESS is given using the PIBA (Problem, Idea, Benefit, Action) template: In **Problem** the project is motivated. In **Idea** the goals of EMPRESS are given. **Benefit** describes the advantages that one will have by employing the EMPRESS methods. In **Action** the way towards realizing the goals is presented, including an overview on the work packages.

2.1 Problem

The software’s role in embedded systems becomes more and more important, distinguishing features of systems are more and more implemented by software. Embedded systems are no longer pure closed, stand-alone systems, but have to interact with all kinds of devices in a changing environment, i.e. each environment needs customized versions of the embedded software. Many environments need life updates for uninterrupted operation at low cost.

High quality of embedded systems becomes crucial while the complexity increases. Therefore systematic approaches for software development and quality assurance required. Quality becomes even more important as software controls increasingly safety critical functions.

Shorter time-to-market is necessary to be economically successful in many application domains developing embedded software. Hence, time and effort to adapt or evolve software have to be reduced.

Basically, current development approaches for embedded real-time systems do not address the evolution of systems. While in DESS an development process for this kind of systems has been defined, this process does not provide means to cope with changing systems. However, only few systems are developed from scratch, and it is to costly to not reuse assets of existing

2 Overview on EMPRESS

systems for new versions. On the other hand, systems change over time, not only due to additional user requirements, but also due to customisation needs, architectural issues, new versions of used components, changes in resource constraints, or port to another hardware.

In the following chapter some examples are given showing problems with evolution of systems in different application domains.

2.1.1 Automotive Domain

At the time being there is only “limited” reuse of software – both from the point of view of the software in the final product, and the specifications describing the functionality. There are lots of different model types, country variants, and releases. For each instance, specifications and code has to be provided. However, systematic and effort saving ways to reuse all kinds of intermediate (specifications) and final (software) products are not available.

This problem will become more severe in the future, as the amount of (embedded) software in cars will increase dramatically in the next years.

2.1.2 Avionics Domain

Systems in airplanes, e.g. control display and management systems, are of very safety critical nature. Therefore high effort is put on validation and verification. As there is no systematic approach to trace changes in systems, each change requires a full validation and verification job. E.g., if the implementation of a change requires 10% of the initial development effort, validation and verification effort equals 100% of the initial V&V effort.

This situation will worsen, as the customers demand more and more customized products, leading to more and more variants.

2.1.3 Consumer Electronics Domain

For digital media (e.g. DVD), there are lots of different products, while on the other hand there are many common functions, e.g. play, stop, fast forward. Different products come into existence due to products for different media, product variations, country variants, releases (changes over time, often triggered by new user requirements). Currently there is no structural approach for incremental development, i.e. only limited reuse and high V&V effort.

2.2 Idea

The Goal of EMPRESS is to develop a methodology and process for real-time embedded software development that supports management of evolution in a flexible and dynamic way. A coherent set of methods covering the whole product life cycle will be developed. As the focus is on evolutionary aspects, EMPRESS deals with phases, which are most affecting evolution and change (see Figure 1).

2 Overview on EMPRESS

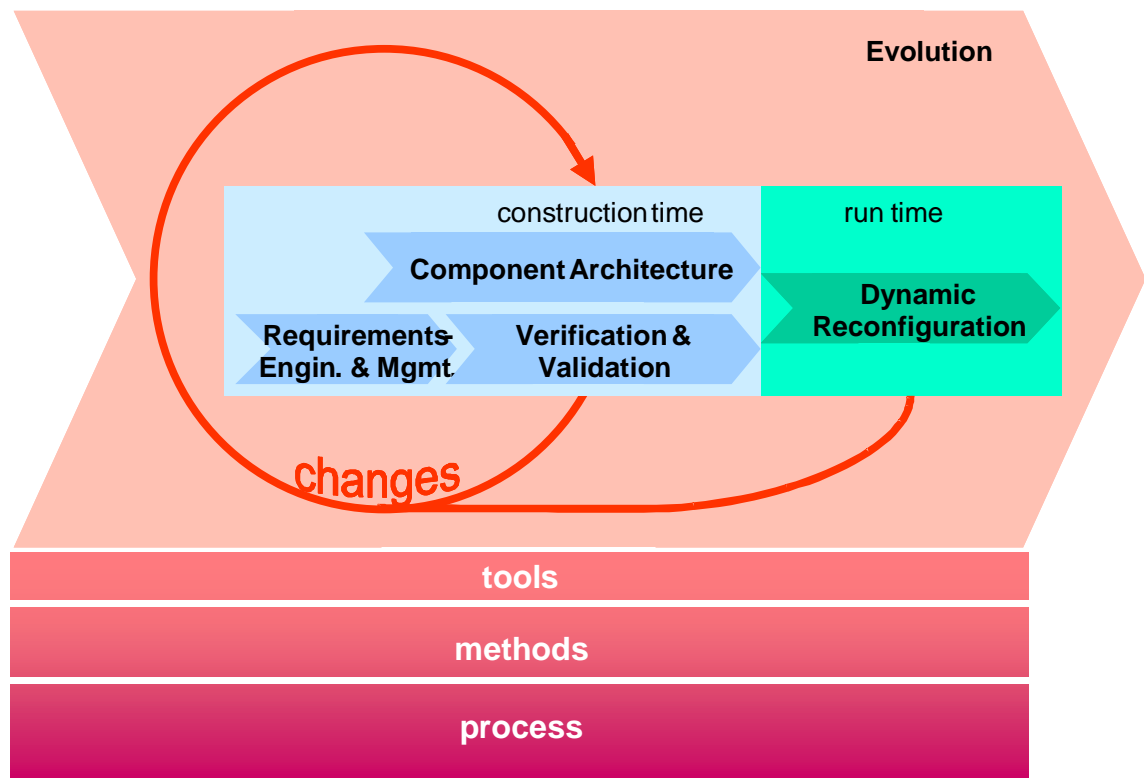


Figure 1: Overview on EMPRESS

During construction time, the system and the software are developed. Employing an appropriate component-based architecture is a major prerequisite to cope with evolution, e.g. to isolate changes to parts of the system. In the development process, changes are manifested in the requirements phase. In this phase one can determine the impact of the change - provided an appropriate traceability mechanisms are in place. Crucial for high quality software is verification and validation – to minimize the effort while guaranteeing the quality specific re-verification and re-validation are required.

During run time, reconfiguration should be possible as dynamic as required. Stop the system, load new (validated) software, restart the system is the easiest way to do it. However, often a system cannot be stopped, e.g. a satellite or a pacemaker. Specific infrastructure and components are necessary to update a system without stopping it.

2.3 Benefit

Using the EMPRESS methods and process a company developing embedded software will benefit in several ways:

- **A c c e l e r a t e d** and more efficient development and adaptation of systems (versions/product lines)
- **H i g h q u a l i t y** systems containing software
- **R e d u c e d** effort for updating running systems

3 Conclusion

2.4 Action

To achieve its goals, in EMPRESS five work package are defined (see Figure 2).

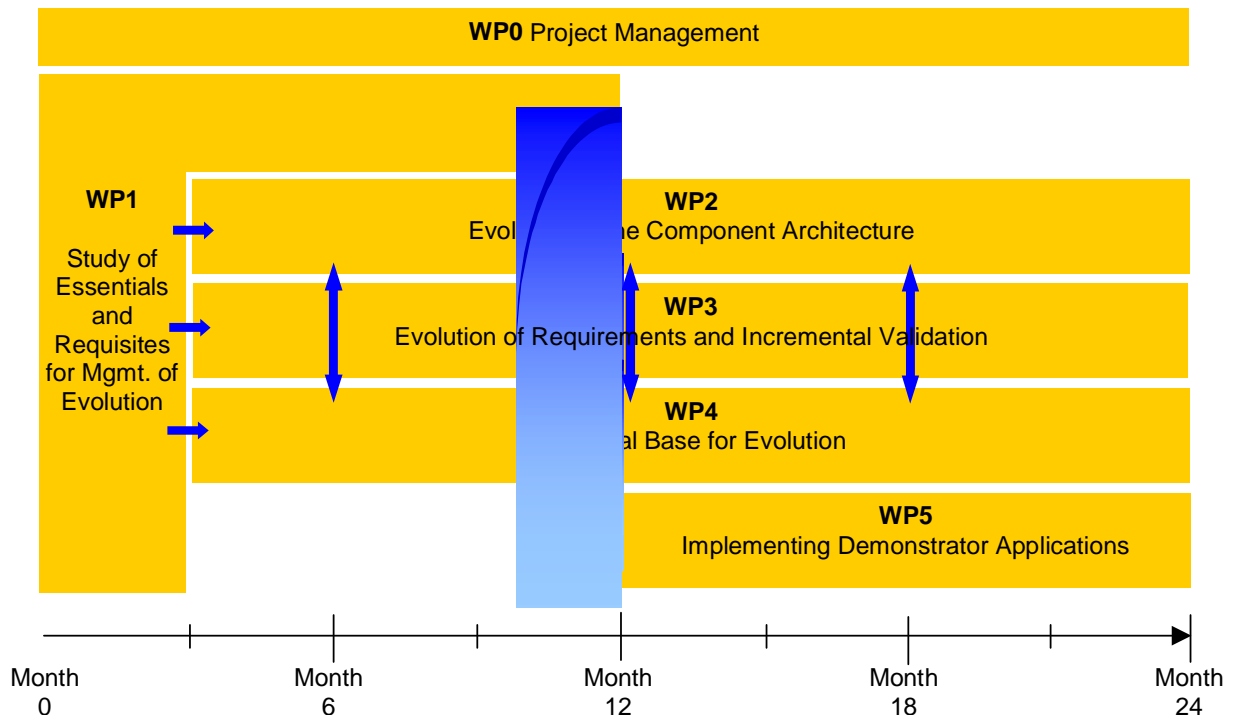


Figure 2: Workpackages

In WP1 the state of the practice (industry view), the state of the art (academic view), related projects (e.g. EMPRESS builds on the results of DESS), industry needs (what is required by the industry near- or midterm), and research questions (which areas are not understood well today) will be determined.

The architectural issues and component based development are addressed in WP2. As life updates have strong requirements on the components, this topic is also part of WP2.

Requirements engineering (elicitation, classification) and management (traceability, change management) are the first part of WP3. (Re-)validation and (re-)verification of changed systems are the second part of it.

Formal approaches are dealt with in WP4. As a basis for the WP2 and WP3 methods models and notations for changing products and change processes are developed.

The applicability and usefulness of developed methods will be demonstrated on WP5.

3 Conclusion

EMPRESS is one of a series of projects in the ITEA context dealing with embedded software development. Synergies are expected by building on results of some projects (DESS, @TERMINALS), learning of some projects through partners participating in several projects

3 Conclusion

(CAFÉ, ESAPS), and cooperating via e.g. joint workshops (MOOSE).

In EMPRESS, we will use methods and processes of all of the above mentioned projects whenever possible, will enhance them to cope with evolution, and we will develop new methods, to fill remaining gaps. In this way EMPRESS will come up with a coherent set of methods and a process enabling companies to efficient and effective deal with the evolution of there embedded real-time software.