HAVEit

Highly automated vehicles for intelligent transport

7th Framework programme
ICT-2007.6.1
ICT for intelligent vehicles and mobility services
Grant agreement no.: 212154

The future of driving.

Deliverable D21.5

Platform with integrated partner applications available

Version number	Version 1.0
Dissemination level	CO
Lead contractor	Continental Automotive GmbH
Due date	31.08.2010
Date of preparation	28.10.2010

Authors

Name	Company
Michael Gutknecht	USTUTT
Sergej Bahnmüller	USTUTT
Philipp Luithardt	USTUTT
SergiyPrutyanyy	USTUTT

Project Managers

Alfred Hoess
Continental Automotive GmbH
Siemensstr. 12
93055 Regensburg, Germany
Phone +49 941 790-5786
Telefax +49 941 790 99 5786
E-mail: Alfred.Hoess-EXT@continental-corporation.com

Holger Zeng
Continental Automotive GmbH
Siemensstrasse 12
93055 Regensburg, Germany
Phone +49 941 790 92330
Fax. +49 941 790 99 92330
E-mail: holger.zeng@continental-corporation.com

Project Co-ordinator

Reiner Hoeger
Continental Automotive GmbH
Siemensstrasse 12
93055 Regensburg, Germany
Phone +49 941 790 3673
Fax +49 941 790 99 3673
E-mail reiner.hoeger@continental-corporation.com

Revision and history chart

Version	Date	Reason
0.1	2010/08/25	Initial document
0.2	2010/08/26	Additional descriptions
0.3	2010/08/27	Internal Review
0.4	2010/09/14	Workaround after project review 1
0.5	2010/10/04	Finalisation after project review 2
1.0	2010/10/28	Final editing and submission to EC
	2011/10/06	Generation of reduced deliverable version for publication

Table of Contents

Executive summary

The main objective of the HAVEit project deals with improvement of safety in road traffic. Therefore, the following three key measures were defined, in order to address that topic: development of the next generation advanced driver assistance systems, optimal task repartition between driver and virtual co-pilot system, and development of failure tolerant, safe vehicle architectures. Especially within the scope of the latter, two safety critical by-wire systems are intended to be developed by the HAVEit consortium: a steer-by-wire system as well as a brake-by-wire system.

University of Stuttgart's challenging task is to develop a fail-operational computational platform core for both by-wire-systems mentioned above, ensuring the required safety for drive-train systems to adhere to both of those systems. Here, the overall development process is structured as described in the following.

The first step deals with the development of a generic, configurable platform core. This includes development of the required hardware components of the platform cores, development of a configurable redundancy management software package based on the redundancy management developed within the predecessor project SPARC, and development of a suitable configuration tool, allowing the generic platform core to be customised to the specific target system as needed. The development was successfully finalised in January 2010, when USTUTT presented its first generic platform core demonstrator (see HAVEit deliverable D21.3).

Having developed the generic platform core, second step deals with configuration of that platform cores depending on its specific target system. Here USTUTT worked together with the by-wire-system responsible partners in order to collect the required configuration data and to implement the by-wire systems' specific configurations utilising the above mentioned configuration tool. Main outcome of this process were two so-called bare platform cores, each configured for its specific by-wire system, but still without the partners' control applications. This step was successfully achieved in July 2010 by presentation of the bare platform core demonstrators (see HAVEit deliverable D21.4).

The third development step followed right after and treats integration of the partners' control applications. During August 2010, these applications were integrated into the software environment, which means that they were linked to the redundancy management. In addition, a first verification of the integration took place in the laboratory in order to ensure, that application on one side and redundancy management on the other side are interacting in a correct manner. In parallel, the application dependant "Application Management" modules were integrated into the software environment and proved for proper operation, too. By successfully finishing integration verification the final platform cores with integrated partner applications are available in the USTUTT laboratory. Outcome of this third development step is topic of the present HAVEit deliverable D21.5.

In the upcoming period, the system specific platform cores will be integrated into the partners' vehicles. While during verification in the laboratory the main focus lay on the correct interaction of redundancy management and control application, now functional tests will be driven in order to verify that the final systems – and in consequence the whole vehicles – perform in an adequate manner concerning the particular by-wire-functionality.

1 Introduction

The software being executed within each XCC can be systematically abstracted into the two software layers being shown in figure 1. First is the so called “Redundancy Management and OS layer”. This layer embraces all software modules being necessary to ensure the basic operation of the XCC like for example

- software service scheduling and timing
- provision and control of the XCC database access
- provision of access to the hardware environment
- timely synchronisation of the XCC to the own FlexRay bus

as well as those software modules providing functionalities which are closely related to the safety aspects like for instance

- ensuring integrity of the XCC input and output dataflow
- detection and treatment of failures within the XCC internal processing
- management of the platform core internal behaviour and
- management of the platform wide operation mode.

Within HAVEit WP2100 much effort was spend to develop a configuration process as well as a suitable configuration tool for most of the software modules of the “Redundancy Management and OS layer”. In consequence, by presenting the bare platform cores in the previous HAVEit deliverable D21.4, these software modules were successfully configured for the particular by-wire systems and tested to operate in the expected manner. Latter includes, for example, correct detection of failures within the input dataflow and proper treatment by means of failure signalling, adaption of performance, or reconfiguration of the platform core.

As indicated so far, except for handling the overall XCC input and output dataflow, the “Redundancy Management and OS layer” mainly focuses the platform core internal issues and is therefore mostly independent from the particular system functionality (e.g. brake-by-wire functionality), which is intended to be realised within the particular platform. This is due to the basic idea of our platform, to generally implement several, independent functionalities within the same computational platform core.

The system specific functionalities are encapsulated in the second software layer – the so called “Application Layer”. Here, the control laws (partner applications) of the related functionality are executed, as well as the related Application Management (AppMa) module for the system specific aggregates. Both – partner application (Appli) and AppMa – are typically generated manually. The partner applications are – as the name implies – provided by the partners responsible for the particular by-wire-system, while the related Application Management is generated by USTUTT for all specific system functionalities. Integration of Appli and AppMa in to the so far bare platform cores as well as integration verification are issues of the present deliverable.

In the following, we intend to give an overview about the Application Management in general and its interaction with the partner application (chapter 2). Right after, an overview about the integration task of the partner applications is given (chapter 3), followed by key conclusions (chapter 4).

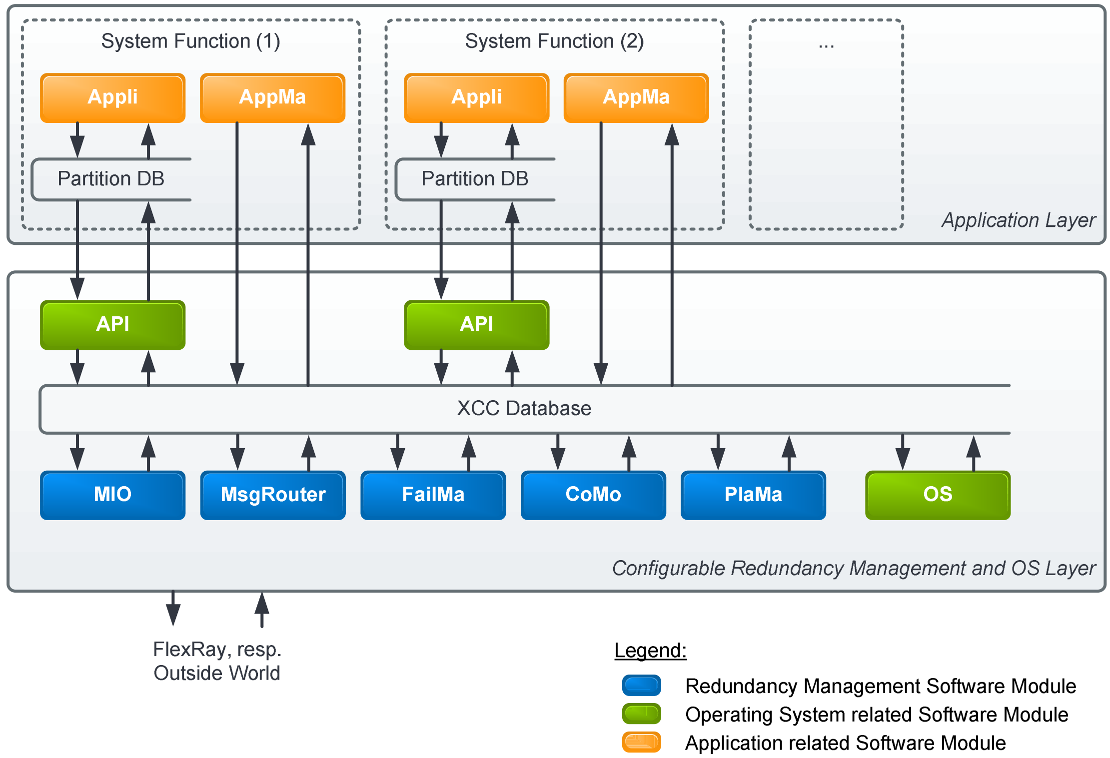

Figure 1: XCC Software Layers

4 Conclusions / Outlook

Aim of the development within HAVEit's WP 2100 is provision of a fail-operational computational platform for the by-wire systems of WP 4100 and WP 4200. Figure 9 gives an overview about the particular steps which have been specified in order to achieve that goal.

- **Step 1**
  First step covered specification and development of the redundant control computers (XCC), required to build up the fail-operational platform core. By successfully finalising that development, the basis for assembling the fail-operational platform cores was given from the hardware point of view. The outcome of that development has been presented within the HAVEit deliverable D21.1.
- **Step 2**
  In parallel to the hardware development, a configuration concept was developed, in order to make the Redundancy Management software environment, which has been developed in the predecessor project SPARC, generally generic. Latter covered not only the task, to make the software modules configurable, but also to provide an appropriate configuration process, which derives the full configuration data out of a system specific database. For further details about the concept, please see the HAVEit deliverable D21.2.
- **Step 3**
  Having developed the XCC hardware on one side, and having implemented the configurable Redundancy Management software modules as well as the related configuration process on the other side, the generic platform cores were created. Latter constitute a principally configurable fail-operational platform core, consisting of two XCCs communicating via two independent FlexRay busses plus the basic OS and Redundancy Management, not configured for any specific system functionality so far. The outcome has been summarised in HAVEit deliverable D21.3.
- **Step 4**
  Applying the configuration tool being presented in the previous step 3, the generic platform cores were configured for the target systems of WP 4100, respectively WP 4200. This means, the Redundancy Management was configured and its failure-detection and -treatment functionalities were verified concerning the particular system specific dataflow. Outcome of that development step were the so called bare platform cores, configured for the target systems (incl. communication, failure-treatment, and management of the whole platform), but still without the system specific control applications. The according result was presented in the HAVEit deliverable D21.4.
- **Step 5**
  Based on the bare platform cores shown in the previous step, the final platform cores were generated, by integration of the partner applications as well as the related Application Management modules. Especially by proving the application to be correctly integrated into the so far bare platform cores, the final (complete) platform cores were achieved. Latter is topic of the present deliverable D21.5.

By finally integrating the partner applications into the XCC software environment and by verifying that integration to be successfully, the last important development step within the scope of WP 2100 is achieved. Therefore, the present deliverable constitutes the last one of WP 2100. Having finalised development within the USTUTT laboratory, the resulting platform cores will now be integrated into the target vehicles of WP 4100 and WP 4200 during the next moths. Here, simulation of the system specific aggregates (as it was done in the laboratory) will no longer be necessary; instead, the platform cores will "see" real aggregates for the first time. In consequence, further improvement of the platform cores as well as functional verification of the partner Applications will be issues within the scope of the vehicle integration phase (WP 4100 and WP 4200), in order to finally achieve the very important HAVEit milestone M4 “vehicle integration successfully completed”.

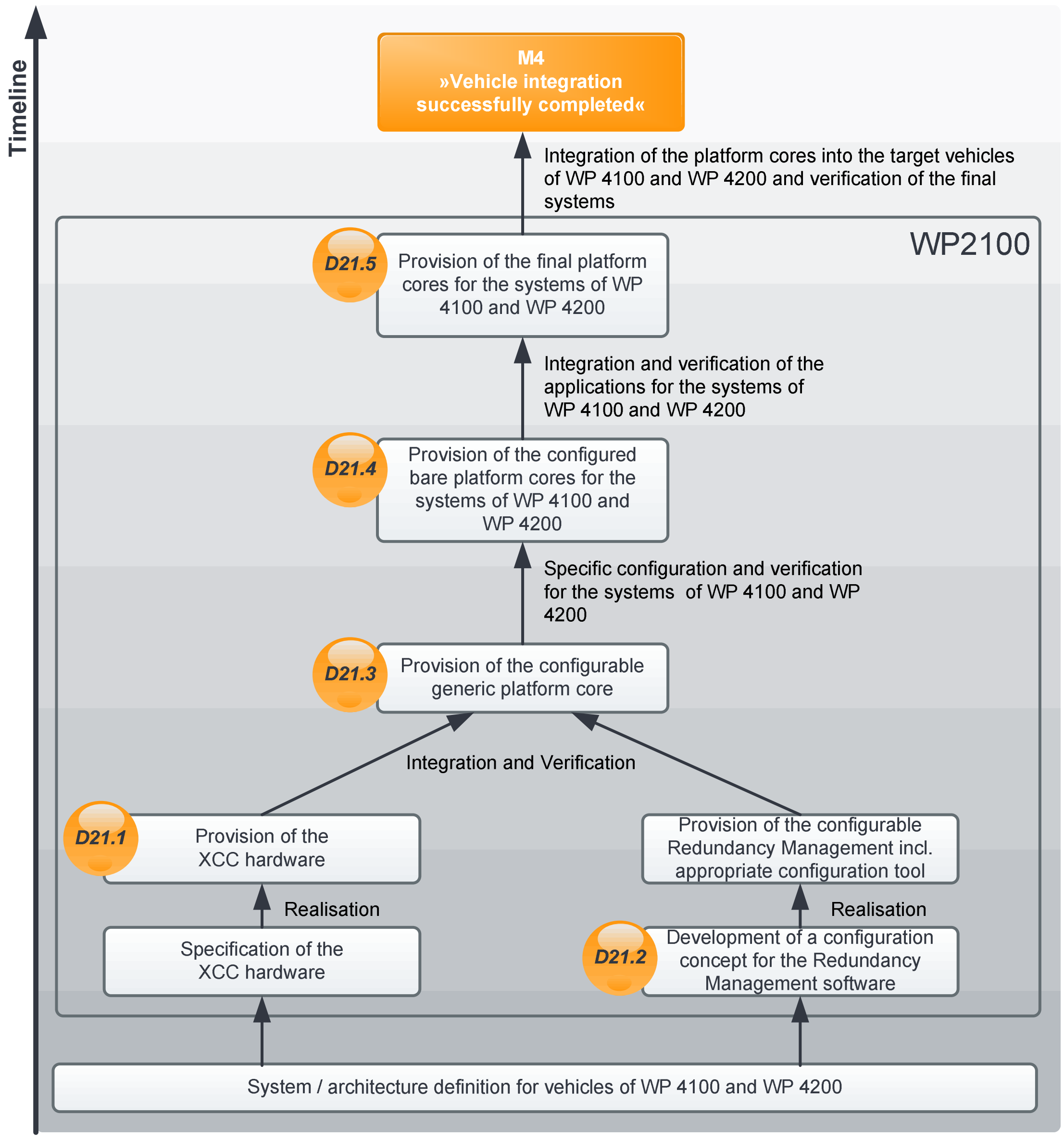

Figure 9: WP 2100 within the context of HAVEit development

References

[1] D21.1 “CSC and XCC HW-development complete”, HAVEit deliverable, 2009

[2] D21.2 “Software- and configuration-process-concept available”, HAVEit deliverable, 2009

[3] D21.3 “Generic platform core demonstrator in lab available”, HAVEit deliverable, 2010

[4] D21.4 “Bare platform tested and ready for the integration of partner applications”, HAVEit deliverable, 2010

Annex 1 Abbreviations

API	Application Programmer Interface
Appli	Application
AppMa	Application Management
BbW	Brake-by-Wire
CoMo	Consolidation Module
DB	Database
FailMa	Failure Management
HBOSS	USTUTT Host Simulator
MIO	Module Input / Output
MsgRouter	Message Router
OS	Operating System
PlaMa	Platform Management
RM	Redundancy Management
SbW	Steer-by-Wire
USTUTT	University of Stuttgart
WP	Workpackage
XCC	X-by-wire Control Computer