Document Title	Specification of Update and Configuration Management
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	888

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-10

Document Change History

Date	Release	Changed by	Description
2017-10-27	17-10	AUTOSAR Release Management	• Initial release

Disclaimer

This work (specification and/or software implementation) and the material contained in it, as released by AUTOSAR, is for the purpose of information only. AUTOSAR and the companies that have contributed to it shall not be liable for any use of the work.

The material contained in this work is protected by copyright and other types of intellectual property rights. The commercial exploitation of the material contained in this work requires a license to such intellectual property rights.

This work may be utilized or reproduced without any modification, in any form or by any means, for informational purposes only. For any other purpose, no part of the work may be utilized or reproduced, in any form or by any means, without permission in writing from the publisher.

The work has been developed for automotive applications only. It has neither been developed, nor tested for non-automotive applications.

The word AUTOSAR and the AUTOSAR logo are registered trademarks.

Table of Contents

1 Introduction and functional overview

This software specification contains the requirements, functional description and Adaptive Platform Service interfaces required to realize the Update and Configuration Management functional cluster of the AUTOSAR Adaptive Platform. The Update and Configuration Management functional cluster has the responsibility of updating, installing and removing software from an Adaptive Platform in a safe and secure way while not sacrificing the dynamic nature of the AUTOSAR Adaptive Platform. The Update and Configuration Management functional cluster is responsible for:

- Checking preconditions to ensure an installation can be performed safely
- Validating the outcome of a software update to the Adaptive Platform
- Providing a way to revert the Adaptive Platform to a known functional state in case of failure

In addition to updating and changing software on the Adaptive Platform, the Update and Configuration Managementcluster is also responsible for updates and changes to the Adaptive Platform itself, including all functional clusters, the underlying posix OS and it's kernel with the responsibilities defined above.

In order to allow flexibility in how the Update and Configuration Management cluster is used, it will expose its functionality via ara::com service interfaces, not direct APIs. This ensures that the user of the Update and Configuration Management cluster does not have to be located on the same ECU.

2 Acronyms and Abbreviations

The glossary below includes acronyms and abbreviations relevant to the Update and Configuration Management that are not included in the [1, AUTOSAR glossary].

Abbreviation / Acronym:	Description:
UCM	AUTOSAR Adaptive Update and Configuration Management functional cluster

3 Related documentation

3.1 Input documents & related standards and norms

[1] Glossary
AUTOSAR_TR_Glossary

[2] Requirements on Update and Configuration Management

AUTOSAR_RS_UpdateAndConfigManagement

[3] Specification of Manifest
AUTOSAR_TPS_ManifestSpecification

3.2 Related specification

See chapter 3.1.

4 Constraints and assumptions

4.1 Limitations

Management of application life cycle during update process isn't addressed in this release. Currently it is the responsibility of the Adaptive Application triggering the update process.

The UCM receives a locally available software package for processing. The software package is downloaded by another application, i.e. there is no ara::com interface for transferring software packages directly to the UCM.

No security aspects are considered yet.

Validation requirements, IE what to validate and what information is required to perform validations are not considered in this release.

Meta-data, configuration data or manifests contained inside a `Software Package` is mentioned in many places in this document. This is to showcase where such information will be stored for implementers, however the form or content of this meta-data are not considered in this release.

A rollback to a stable version of the platform is not yet considered in this specification.

4.2 Applicability to car domains

No restrictions to applicability.

5 Dependencies between functional clusters

The UCM functional cluster exposes services to client applications via the ara::com middleware. The dependencies to other functional clusters hasn't been considered in this release.

6 Requirements Tracing

The following tables reference the requirements specified in [2] and links to the fulfillment of these. Please note that if column “Satisfied by” is empty for a specific requirement this means that this requirement is not fulfilled by this document.

Requirement	Description	Satisfied by
[RS_UCM_00001]	UCM shall support installing new software on Adaptive Platform	[SWS_UCM_00001]
[RS_UCM_00002]	UCM shall support reporting software information for an Adaptive Platform	[SWS_UCM_00004]
[RS_UCM_00003]	UCM shall support updating installed software on Adaptive Platform	[SWS_UCM_00003]
[RS_UCM_00004]	UCM shall support uninstalling software	[SWS_UCM_00002]
[RS_UCM_00005]	UCM shall make sure that data stored by uninstalled software is deleted	[SWS_UCM_00002]
[RS_UCM_00006]	UCM shall validate Software Package during processing	[SWS_UCM_00005]
[RS_UCM_00007]	UCM shall check dependencies of a Software Package	[SWS_UCM_00007]
[RS_UCM_00008]	UCM shall support recovering mechanism in case of failed update process	[SWS_UCM_00008]

7 Functional specification

7.1 Technical Overview

One of the declared goals of Adaptive AUTOSAR is the ability to flexibly update the software and its configuration through over-the-air updates. The UCM is therefor responsible for updates of Adaptive Platform Applications and changes to the Adaptive Platform itself, including all functional clusters and the underlying OS. To support changes in the software on an Adaptive Platform the UCM functional cluster realizes an Adaptive Platform service that handles software update requests. The UCM functional cluster provides service interfaces that expose its functionality to retrieve Adaptive Platform software information and consistently execute software updates.

To do so the following functionalities are provided:

1. Software update services (see 7.2.1)
2. An interface for querying the Adaptive Platform on its currently installed software (see 7.2.2)
3. Update consistency checks along with precondition checks (see 7.2.3)

However the UCM functional cluster is not responsible to:

- retrieve software updates via the `Diagnostic Protocol` or expose diagnostic services via `UDS`. This could be implemented by a Diagnostic Application or another Adaptive Platform application, see Figure 7.1.
- initiate the update process. UCM realizes a service interface to achieve this operation but it is not the initiator of the update process.
- verify that the vehicle is in a correct state before executing a software update procedure on demand.
- communicate with other Adaptive Platforms or Classic Platforms within the vehicle. Therefore management of software dependencies between different physical or virtual ECU software platforms is out of UCM's scope.

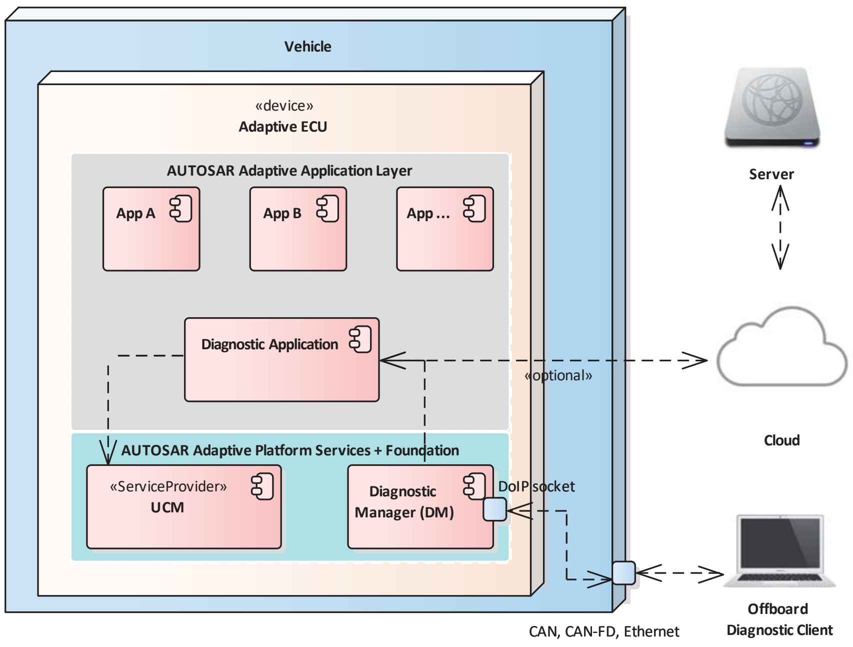

Figure 7.1: Architecture overview for diagnostic use case

7.2 Software Package processing

This section presents a short summary of the relationship between the UCM and the `Software Package`.

Beside application and configuration data each `Software Package` contains a manifest providing meta data like package name, version and other meta information needed for processing the package. As described in [3] manifests are specified by a meta model which already foresees to put meta data of a `Uploadable Software Package` below meta-class `SoftwareCluster`. As for now a `SoftwareCluster` does not yet specify properties specific for UCM.

The data content of a `Software Package` can contain, for example, one or several Adaptive Applications, kernel or firmware updates, or updated configuration and calibration data.

7.2.1 Software update

The specification of the UCM functional cluster covers the basic functionality of a package manager, similar to the ones found in Linux based system, i.e. the UCM realizes the installation, update or uninstallation of software on an Adaptive Platform.

[SWS_UCM_00001] Software installation ⌈ The method `ProcessSwPackage` in the service interface `sub:PackageManagement` provided by the UCM shall expose the functionality for installing a `Software Package`. In this case the `Software Package` is extending the Adaptive Platform with new software. ⌋*(RS_UCM_00001)*

[SWS_UCM_00002] Software uninstallation ⌈ UCM provides a service interface that shall expose the functionality to uninstall software from the Adaptive Platform. In this case the contents of the `Software Package` is configuration data that describes which data to remove from the Adaptive Platform. In addition, the uninstallation process shall purge the persisted data owned by the removed applications contained in the `Software Package`. ⌋*(RS_UCM_00004, RS_UCM_00005)*

[SWS_UCM_00003] Software update ⌈ UCM shall provide a service interface that exposes the functionality in order to update the software of the Adaptive Platform. In this case the `Software Package` is modifying software currently present on the Adaptive Platform. ⌋*(RS_UCM_00003)*

7.2.2 Information reporting

[SWS_UCM_00004] Report software information ⌈ To realize a coherent update process UCM shall provide a service interface, (`GetInstalledSw`), that enables querying the Adaptive Platform about its current software information. ⌋*(RS_UCM_00002)*

[SWS_UCM_00006] Software update action reporting ⌈ In order to indicate the status of an update action the UCM functional cluster shall provide a reporting mechanism that indicates a software update action success or failure on the action completion (see `GetInstallationResult`). ⌋*()*

7.2.3 Software update consistency

The specification of the UCM describes the functional cluster behavior and interfaces that ensure the consistency of the Adaptive Platform software during and after the software modifications. In the following requirements a software update action is a software installation, uninstallation or update (see [SWS_UCM_00001], [SWS_UCM_00002] and [SWS_UCM_00003]).

[SWS_UCM_00005] Software Package content validation ⌈ To realize a coherent update process the UCM shall validate the `Software Package` before executing a software update action by use of cryptographic signatures. ⌋*(RS_UCM_00006)*

[SWS_UCM_00007] Software Package dependency validation ⌈ To realize a coherent update process the UCM shall validate that the dependencies for the `Software Package` are met on the Adaptive Platform before executing a software update action. ⌋*(RS_UCM_00007)*

[SWS_UCM_00008] Recovery mechanism ⌈ As the UCM functional cluster cannot guarantee a fail-safe update action, a recovery functionality shall be implemented by the UCM. The recovery process must revert the Adaptive Platform software to a previous known functional state, e.g. by restoring a file system snapshot created before the update action. ⌋*(RS_UCM_00008)*

8 API specification

8.1 Type definitions

This chapter lists of types provided by the `UCM`.

8.1.1 InstallationRequestIdentifierType

Name	InstallationRequestIdentifierType
Kind	Type
Derived from	uint32
Description	Represents a handle identifier used to reference a particular installation request.

8.1.2 ProcessingResultType

Name	ProcessingResultType
Kind	Type
Derived from	uint8
Description	Represents the result of processing a software package.

	SUCCESS	0x00	Software package has been successfully processed.
	INVALID_MANIFEST	0x01	Package manifest could not be read.
	INSUFFICIENT_MEMORY	0x02	Not a enough disk space to perform installation.
	MISSING_DEPENDENCIES	0x03	Package dependencies are not fulfilled.
	RESERVED	0x04-0xFF	Reserved for future use.

8.1.3 SwInfoListType

Name	SwInfoListType
Kind	Type
Derived from	List of type SwInfoType
Description	Represents a list of software packages that are described with their name and version.

8.1.4 SwInfoType

Name	SwInfoType		
Kind	Struct		
Description	Represents the information of a single software package needed for package management		
	Name	***Type***	***Description***
	Name	String	Name of the software package
	Version	String	Version of the software package

8.1.5 PackageManagerTaskType

Name	PackageManagerTaskType
Kind	Type
Derived from	uint8
Description	Represents what task is currently executing

	IDLE	0x00	No task is being executed
	INSTALL	0x01	Installation, Update or Uninstallation is currently in progress
	ROLLBACK	0x02	Update has failed and recovery is in process
	RESERVED	0x03-0xFF	Reserved for future use.

8.1.6 PackageManagerStatusType

Name	PackageManagerStatusType		
Kind	Type		
Derived from	uint8		
Description	Represents the state of current installation process		
	READY	0x00	UCM is ready to start a new installation process
	BUSY	0x01	UCM is currently in the middle of an installation process
	FINISHING	0x02	All packages have been installed, but the update hasn't been committed yet (this state is currently unused)
	ABORTING	0x03	Package installation failed, and UCM is currently recovering to the previously working configuration (this state is currently unused)
	RESERVED	0x04-0xFF	Reserved for future use.

8.2 Service Interfaces

8.2.1 PackageManagement

This service interface offers methods for basic package management tasks.

8.2.1.1 Methods

Name	ProcessSwPackage

Description	Read a SW Package and process it. The installation steps are derived from the description of the SW package meta data.		
Parameters	FilePath	***Description***	File path of the SW package archive
		Type	String
		Direction	IN
	SizeOfPackage	***Description***	Size of the SW package
		Type	Uint64
		Direction	IN
	Checksum	***Description***	Checksum of the SW package archive
		Type	Variable size array of uint8
		Direction	IN
	InstallationRequestId	***Description***	Handle ID for the installation request
		Type	InstallationRequestIdentifierType
		Direction	OUT

Name	GetInstallationResult		
Description	This method returns the result after a SW package has been processed.		
Parameters	InstallationRequestId	***Description***	Handle ID of the request
		Type	Uint32
		Direction	IN
	Result	***Description***	Result of the processing
		Type	ProcessingResultType
		Direction	OUT
	LogFilePath	***Description***	File path of the log file that contains the log messages of the SW package processing
		Type	String
		Direction	OUT

Name	GetInstalledSw		
Description	This method returns the installed SW packages on the same adaptive platform instance.		
Parameters	InstalledSw	***Description***	List of installed software (name and version)
		Type	SwInfoListType
		Direction	OUT

8.2.1.2 Fields

Name	CurrentStatus
Description	Contains the current status of the package management unit.
Type	PackageManagerStatusType
HasGetter	yes
HasSetter	no
HasNotifier	yes

Name	CurrentTask
Description	Contains the current task of the package management unit.
Type	PackageManagerTaskType
HasGetter	yes
HasSetter	no
HasNotifier	yes