Document Title	Requirements on Operating System Interface
Document Owner	AUTOSAR
Document Responsibility	AUTOSAR
Document Identification No	718

Document Status	Final
Part of AUTOSAR Standard	Adaptive Platform
Part of Standard Release	17-03

Document Change History			
Date	**Release**	**Changed by**	**Description**
2017-03-31	17-03	AUTOSAR Release Management	• Initial release

Disclaimer

Table of Contents

1 Scope of this document

This document specifies the requirements of Adaptive Platform on the Operating System that are contained in the platform.

1.1 Document Conventions

The representation of requirements in AUTOSAR documents follows the table specified in [TPS_STDT_00078], see Standardization Template, chapter Support for Traceability ([1]).

The verbal forms for the expression of obligation specified in [TPS_STDT_00053] shall be used to indicate requirements, see Standardization Template, chapter Support for Traceability ([1]).

2 Acronyms and Abbreviations

There are no acronyms and abbreviations relevant within this document that are not included in the [2, AUTOSAR glossary].

3 Constraints and assumptions

3.1 Limitations

This chapter lists known limitations of `Operating System` Interface in terms of unimplemented requirements. The intent is to not only provide an indication how the requirement specification will evolve future releases.

The following requirements are described within this document but not otherwise considered in this release:

- [RS_OSI_00201]
- [RS_OSI_00202]
- [RS_OSI_00203]
- [RS_OSI_00204]

The functionality described above is subject to modification and will be considered for inclusion in a future release of this document.

3.2 Applicability to car domains

No restrictions to applicability.

4 Functional overview

The Operating System is responsible for run-time resource management (including time) for all `Applications` on and within the `Adaptive Platform`. This includes not only the `Adaptive Applications` that run on top of ARA provided by `Adaptive Platform`, but also the `Functional Clusters` that constitute the platform, which are also implemented as `Applicaitons`. The OS functions in cooperation with `Execution Management` which is responsible for platform initialization and the start-up / shut-down of Applications.

Note that this Operating System Interface (OSI) requirement specification contains two different categories. The first category contains the requirements that are directly needed by the `Adaptive Applications`. The other category contains the ones that are needed by the `Adaptive Platform` to realize implementation of `Functional Clusters`, especially the required mechanisms are difficult or inefficient to be implemented by other software entity than the OS. The most notable such `Functional Cluster` requiring the various OS mechanisms is `Execution Management`.

5 Requirements Tracing

The following table references the features specified in [3] and links to the fulfillments of these.

Feature	Description	Satisfied by
[RS_Main_00002]	AUTOSAR shall provide a software platform for high performance computing platforms	[RS_OSI_00101] [RS_OSI_00200] [RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203]
[RS_Main_00010]	AUTOSAR shall support the development of safety related systems.	[RS_OSI_00204] [RS_OSI_00206]
[RS_Main_00049]	AUTOSAR shall provide an Execution Management for running multiple applications	[RS_OSI_00101] [RS_OSI_00105] [RS_OSI_00206]
[RS_Main_00050]	AUTOSAR shall provide an Execution Framework towards applications to implement concurrent application internal control flows.	[RS_OSI_00101] [RS_OSI_00104] [RS_OSI_00200] [RS_OSI_00203] [RS_OSI_00204] [RS_OSI_00205]
[RS_Main_00060]	AUTOSAR shall provide a standardized software interface for communication between Applications	[RS_OSI_00100] [RS_OSI_00104]
[RS_Main_00106]	AUTOSAR shall provide the possibility to extend the software with new SWCs without recompiling the platform foundation	[RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203] [RS_OSI_00206]
[RS_Main_00200]	AUTOSAR specifications shall allow resource efficient implementations	[RS_OSI_00200] [RS_OSI_00201] [RS_OSI_00202] [RS_OSI_00203]
[RS_Main_00340]	AUTOSAR shall support the observance of timing requirements	[RS_OSI_00102] [RS_OSI_00205]
[RS_Main_00400]	AUTOSAR shall provide a layered software architecture	[RS_OSI_00100]
[RS_Main_00450]	AUTOSAR shall standardize access to general purpose I/O	[RS_OSI_00104]
[RS_Main_00460]	AUTOSAR shall standardize methods to organize mode management on Application, ECU and System level	[RS_OSI_00104] [RS_OSI_00105]

6 Requirements specification

This chapter describes all requirements driving the work to define the operating system's functionality.

6.1 Assumption of Use

This section describes application use cases that will run on the AUTOSAR Adaptive Platform. These use cases are not requirements to the AUTOSAR Adaptive Platform in the strict sense, but rather assumptions on properties of Application running on the AUTOSAR Adaptive Platform. These assumptions are used to motivate the further requirements, and provide hints for application developers to check if their use cases are covered in this specification document and which specific requirements are derived from those use cases.

The Operating System section defines requirements on the Operating System that applications can consider fulfilled in order to achieve their function.

6.2 General Requirements

[RS_OSI_00100] Operating System Interface ⌈

Type:	draft
Description:	The foundation of the Operating System provided to the Application shall be POSIX-compliant as defined by PSE51.
Rationale:	The defined functionality of the POSIX profile PSE51 defined by IEEE1003.13 [4] is provided by various off-the-shelf operating systems. The PSE51 profile is intended for embedded systems, with a single multi-threaded process, no file system, no user and group support and only selected options from more general IEEE1003.1 [5], which is the well-know POSIX standard. PSE51 offers functions for basic synchronized I/O, high-resolution timer, signals, semaphores, shared memory and threads. As the envisioned application software components will not require to fork new processes themselves, and only need limited direct access to files, the PSE51 profile is thought to be sufficient.
Dependencies:	–
Use Case:	Application portability.
Supporting Material:	IEEE1003.13 [4] and IEEE1003.1 [5]

⌋*(RS_Main_00060, RS_Main_00400)*

[RS_OSI_00101] Multi-threaded Execution ⌈

Type:	draft
Description:	The Operating System shall enable execution of multi-threaded Application.

Rationale:	`Application` can be multi-threaded. To maximize usage of high-performance multi-core CPUs and to obtain the application's computation results within the required time bounds, applications will parallelize their computations, and dynamically allocate their computations to multiple cores by using threads.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00002*, *RS_Main_00049*, *RS_Main_00050*)

[RS_OSI_00102] Time-Triggered Execution ⌈

Type:	draft
Description:	The `Operating System` shall facilitate time-triggered `Application` execution.
Rationale:	`Application` can be time-triggered. The OS needs to provide mechanisms to allow the time-triggered execution of applications. The triggers, need to contain at least external timers, but not limited to.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00340*)

[RS_OSI_00104] Reaction on Application-external Stimuli from devices ⌈

Type:	draft
Description:	The `Operating System` shall enable `Application` to react on external stimuli from devices.
Rationale:	`Application` will react on reception of functional data, signals and timers from the platform. Certain computations shall be executed in reaction on these application-external stimuli.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00050*, *RS_Main_00450*, *RS_Main_00060*, *RS_Main_00460*)

[RS_OSI_00105] Start of Execution Management ⌈

Type:	draft
Description:	The `Operating System` shall provide means to start the `Execution Management` functional cluster as first process.
Rationale:	`Execution Management` is responsible for startup and shutdown of all `Application`s of the Adaptive Platform.
Dependencies:	–
Use Case:	–

Supporting Material:	–

⌋(*RS_Main_00049*, *RS_Main_00460*)

6.3 Operating System Functionality Requirements

[RS_OSI_00200] The `Operating System` shall support common best-effort real-time scheduling strategies on thread level within a process. ⌈

Type:	draft
Description:	The `Operating System` shall support common best-effort real-time scheduling strategies on thread level within a process to guarantee a bounded jitter on execution dispatch to time critical applications.
Rationale:	Threads within a process shall be schedulable within specific time slice to a guarantee bounded jitter.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00002*, *RS_Main_00050*, *RS_Main_00200*)

[RS_OSI_00201] The `Operating System` shall provide mechanisms for system memory budgeting. ⌈

Type:	draft
Description:	The `Operating System` shall provide mechanisms to configure memory budgeting for each `Application` or for groups of `Application`s.
Rationale:	In order to ensure resource availability in the context of multithreaded application, the system integrator/architect may require a set of tools to configure memory budgeting for each `Application` or for groups of `Application`s.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00002*, *RS_Main_00106*, *RS_Main_00200*)

[RS_OSI_00202] The `Operating System` shall provide mechanisms for CPU time budgeting. ⌈

Type:	draft
Description:	The `Operating System` shall provide mechanisms to configure resource budgeting in terms of CPU time for each `Application` or group of `Application`s.

Rationale:	In order ensure schedulability in the context of multithreded application, the system integrator/architect may require a set of tools to configure CPU time allocated for each `Application` or for groups of `Application`s
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00002*, *RS_Main_00106*, *RS_Main_00200*)

[RS_OSI_00203] The `Operating System` should provide mechanisms for binding processes to CPU cores. ⌈

Type:	draft
Description:	The `Operating System` should provide mechanisms for binding individual processes or groups of processes to CPU cores.
Rationale:	In order to ensure correct task schedulability, the system integrator may require a set of tools to configure the CPU affinity.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00002*, *RS_Main_00050*, *RS_Main_00106*, *RS_Main_00200*)

[RS_OSI_00204] The `Operating System` shall support authorized operating system object access for the software entities which are allowed to do so. ⌈

Type:	draft
Description:	The `Operating System` shall provide access rights and permissions mechanisms to achieve secure data access and data exchange.
Rationale:	The `Operating System` consist of a collection of hardware and software objects, e.g. pipes, files. Safety or/and Security related requirements may be imposed to grant special access rights and permissions in order to avoid unauthorized access to communication channels or to ensure exclusive access to the application specific data stored persistently.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00010*, *RS_Main_00050*)

[RS_OSI_00205] The `Operating System` shall provide optimized mechanisms for running periodic, time-based loops. ⌈

Type:	draft
Description:	The `Operating System` shall provide mechanisms to let `Application`s do time-based recurring processing.

Rationale:	Several actions in an embedded device relate to periodic time-based processing. While POSIX includes an API do trigger a timer in a recurring manner, its use relies on signals, and is not optimized for simpler processing loops. OS-specific extensions may further allow this feature to have a lower impact on the system scheduling than POSIX signals.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00050*, *RS_Main_00340*)

[RS_OSI_00206] The `Operating System` shall provide multi-process support for isolation of applications. ⌈

Type:	draft
Description:	The `Operating System` shall provide mechanisms to let multiple `Application`s run isolated from each other.
Rationale:	Each process that participates in an `Application` may have a different level of robustness, safety and security level. As a consequence, an incorrect memory access from one `Application` execution shall not result in a corruption of memory in another `Application`, unless the data area is explicitly shared. In addition, a process may not access or read data from another process without explicit data sharing.
Dependencies:	–
Use Case:	–
Supporting Material:	–

⌋(*RS_Main_00010*, *RS_Main_00049*, *RS_Main_00106*)

7 References

[1] Standardization Template
AUTOSAR_TPS_StandardizationTemplate

[2] Glossary
AUTOSAR_TR_Glossary

[3] Requirements on AUTOSAR Features
AUTOSAR_RS_Features

[4] IEEE Standard for Information Technology- Standardized Application Environment Profile (AEP)-POSIX Realtime and Embedded Application Support
https://standards.ieee.org/findstds/standard/1003.13-2003.html

[5] Standard for Information Technology–Portable Operating System Interface (POSIX(R)) Base Specifications, Issue 7
http://pubs.opengroup.org/onlinepubs/9699919799/

[6] Requirements on Execution Management
AUTOSAR_RS_ExecutionManagement

[7] Requirements on Communication Management
AUTOSAR_RS_CommunicationManagement