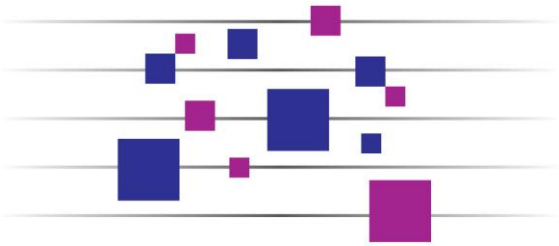

CONCERTO

Guaranteed Component Assembly with Round Trip Analysis for Energy Efficient High-integrity Multi-core Systems

Project Number 333053

D3.1 – Extension of the CONCERTO modelling capabilities to support non-functional properties

Version 1.1

9 May 2014
Final

Public Distribution

UPD, UNIFI, TCS, Atego, EADS, Aicas, INT, BME, AEN, ISEP, CSW, SINTEF, OTG, MDH, TOG

Project Partners: **AENSys, Aicas, Atego, Budapest University of Technology and Economics, Critical Software, EADS, Intecs, ISEP, Maelardalens University, OTG Solutions, SINTEF, Thales Communications & Security, The Open Group, University of Florence, University of Padova**

Every effort has been made to ensure that all statements and information contained herein are accurate, however the Partners accept no liability for any error or omission in the same.

© 2014 Copyright in this document remains vested in the CONCERTO Project Partners.

DOCUMENT CONTROL

Version	Status	Date
0.0	Initial coarse-grained document structure	27 Nov 2013
0.1	Initial baseline	19 Dec 2013
0.2	With coarse description of requirements and derived actions	31 Jan 2014
0.3	Revision of requirements and redistribution to other WPs	04 Feb 2014
0.4	Introduction revision	10 Feb 2014
0.5	Stabilized version for technical partners contributions	17 Feb 2014
0.6	Revision with intended partner contribution for harmonization	03 Mar 2014
0.7	Harmonization draft	06 Mar 2014
0.8	Second harmonization draft	13 Mar 2014
0.9	First review release	21 Mar 2014
1.0	Internal review release	28 Mar 2014
1.1	Document ready for external review	09 May 2014

TABLE OF CONTENTS

LIST OF ABBREVIATIONS

Acronym	Unwound form
ADL	Activities of Daily Living
API	Application Programming Interface
ASIL	Automotive Safety Integrity Level
COTS	Components Off-The-Shelf
CPU	Central Processing Unit
FI^{4}FA	Formalism for Incompletion, Inconsistency, Interference and Impermanence Failures Analysis
FMEA	Failure Mode, Effects, and Analysis
FMECA	Failure Mode, Effects, and Criticality Analysis
FPTC	Failure Propagation and Transformation Calculus
FTA	Fault Tree Analysis
HW	Hardware
M2C	Model-to-code transformation
M2M	Model-to-Model transformation
PIM	Platform Independent Model
PSM	Platform Specific Model
SIL	Software Integrity Level
SW	Software
TLC	Telecommunication
WCET	Worst Case Execution Time
XML	Extensible Markup Language

EXECUTIVE SUMMARY

This deliverable addresses the portion of user requirements pertaining to WP3 and in particular the tasks related to non-functional and system properties support where CONCERTO will extend the results of CHESS by targeting heterogeneous multi-core systems, and include additional applicative domains. The inclusion of industrial domains, the introduction of new standards/versions (e.g. ISO 26262 in the automotive domain and DO-178C in the aerospace domain), and the gap between single core and heterogeneous multi-core systems, demand further efforts to extend CHESS results to properly support dependability and, more specifically, safety properties. Moreover, investigations must be targeted to appropriately model system/hardware characteristics, notably architecture, memory, and energy consumption. These challenges are captured by the elaboration of user requirements provided in this deliverable. Finally, this deliverable defines actions to address the derived requirements and maps the activities of WP3 partners to each action.

1. INTRODUCTION

The purpose of this document is to elicit the portion of user requirements pertaining to WP3, i.e. the work package addressing non-functional and system properties support. In this respect, user requirements are contextualized in the CONCERTO technical background in order to derive fine-grained requirements and define corresponding development actions to be mapped to technical partners.

CONCERTO is aimed at building its results on top of the CHESS project results. CHESS targeted the construction of distributed and single core systems for telecommunication, aerospace, railway, and automotive domains. CONCERTO will extend the results of CHESS by targeting heterogeneous multi-core systems, and include additional applicative domains, notably telecare and petroleum. The inclusion of those new industrial domains, the introduction of new standards/versions (e.g. ISO 26262 in the automotive domain and DO-178C in the aerospace domain), and the gap between single core and heterogeneous multi-core systems, demand further efforts for developing new techniques and for extending CHESS results to properly support dependability and, more specifically, safety properties in the CONCERTO Project. Moreover, investigations will be devoted to appropriately model system/hardware characteristics, notably architecture, memory, and energy consumption.

The main challenges to be covered in this deliverable are:

1. Extending the modelling language to support a richer and more detailed set of dependability characteristics of the system;

2. Extending the modelling language to support adequately safety standards;

3. Extending the language to support system/hardware modelling;

4. Extending analysis support to provide enhanced evaluations of dependability and safety properties of the system.

These challenges are captured by the elaboration of user requirements provided in this deliverable. In some case requirements also entail and overlap with those of other work packages. In order to minimize this overlapping, the context of each deliverable, which sets the ground to discuss the user requirements, is clearly bounded with respect to the other deliverables. In particular, this deliverable mainly deals with dependability and safety aspects of the system under development, while timing issues are left to be covered in WP4 deliverables.

Given the nature of the properties taken into account in this deliverable, their development crosscuts the tasks definition for the work package. This means that this deliverable covers all the Tasks in WP3 as follows:

- identifies the CONCERTO user requirements for WP3, and possibly reallocates some of those preliminary assigned to WP3 to other work packages, notably WP2 and WP4, to be more appropriately addressed;

- puts them in the context of the CONCERTO technical background so as to break them down in sufficiently fine-grained derived requirements;
- defines actions to address the derived requirements and map the effort of WP3 partners to each action.

2. CONSTRAINTS

This section discusses CONCERTO requirements related to non-functional and system properties support. In particular, next section puts requirements in the technical context that will be exploited as a baseline to develop the desired features. A subsequent analysis will be devoted to illustrate each requirement in the list and to clarify the set of derived requirements resulting from the contextualization in the technological framework.

2.1 OVERVIEW

2.1.1 Methodology

The user requirements pertaining to this deliverable are put in the context of the technology baseline of the CONCERTO Project that builds on top of the CHESS Project. The overall landscape of the project is depicted in Figure 1, while the context of this document is illustrated in Figure 2. Figure 1 shows the constellation of artefacts and transformation utilities the CONCERTO landscape is made-up-of. Notably, it depicts the definition of the modelling language, the transformations supporting model-based analysis, and the code generation and execution platform, including monitoring facilities.

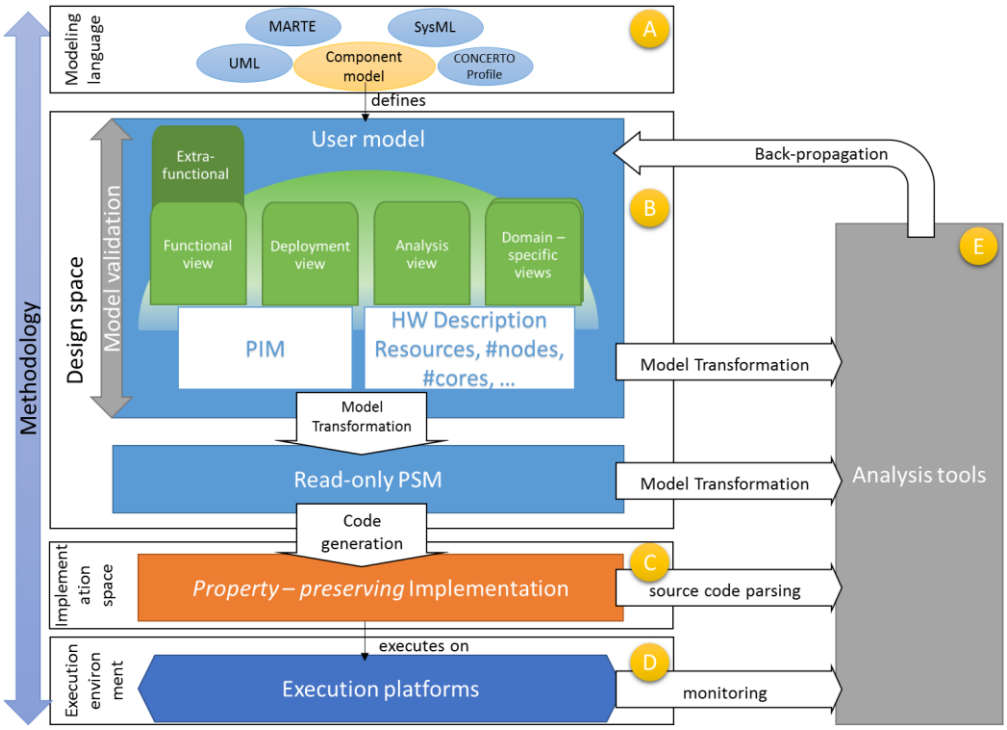

Figure 1. The CONCERTO landscape

On the other hand, Figure 2 shows where and how dependability analysis is exploited. In particular, model transformations produce appropriate inputs for analysis tools that perform their evaluations and back propagate them towards user models by means of other transformations.

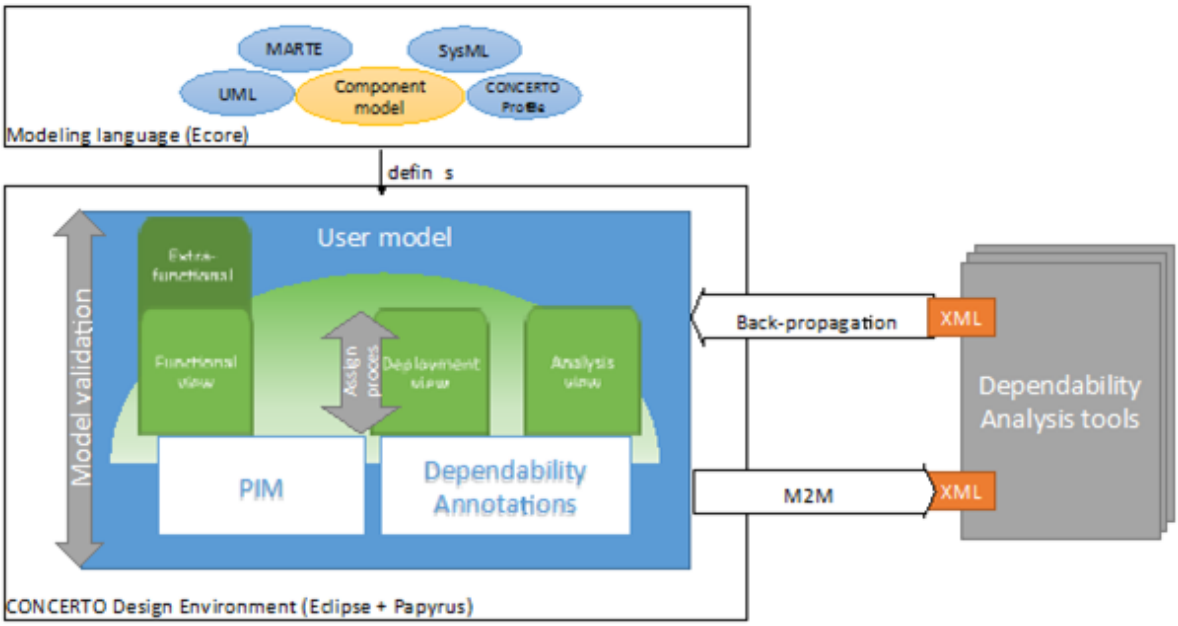

Figure 2. The Dependability analysis support available in CONCERTO

A modelling language defines the set of concepts by which real phenomena can be abstracted for a precise development purpose. In the context of CHESS before and currently CONCERTO, the modelling language provides a design space that drives the user actions by means of modelling views and an online model validation (A). Design views enhance separation of concerns, keeping the developer focused only on a particular aspect of the system (component definition, behaviour, timing properties, and so forth) and reducing the complexity of the modelling task (B).

The user model consists of the platform independent model (PIM) and the platform description (description of the characteristics of the computational hardware and other devices, number of cores per CPU, number of nodes on the network, etc.). Following the CHESS inheritance, the modelling language also defines read-only platform specific model(s) (PSM). The PSM can be exploited as intermediate artefact to both perform finer-grained kinds of analysis, or to generate property-preserving implementations on a selected platform (C).

The user then can trigger the execution of various model based analyses. Depending on the analysis, the user model is transformed through a series of model transformations into a proper input for the analysis tool of choice (C). The results of the analysis tool are then propagated back to the user model by means of back-propagation transformations, which use traces generated by the model transformations when executed in the forward direction. In fact, in order to associate analysis results with the attributes of PIM model elements, the transformations need to produce artefacts that can be related with each

other. Given the semantic gap between design and analysis models and tools, often the transformations can establish many-to-many relationships between design and analysis, thus requiring disambiguation information.

Eventually, source code parsing and monitoring routines (D) provide additional inputs for (design level) fine-grained analysis of those system characteristics that cannot be predicted before concrete code is generated, notably CPU and memory budgets, resource utilization, and so forth.

2.1.2 CHESS technology heritage

The modelling language is defined using the Ecore metamodel provided by the Eclipse Modeling Framework. The design environment is based on the Eclipse framework, in particular it is based on an extended version of the Papyrus editor implemented for the CHESS project. In CHESS, the dependability concerns are supported via the dependability sub-profile. This profile permits users to enrich their functional models. This enrichment enables dependability analysis such as FTA, FMECA, FMEA, State-Based, Wide Data-flow, Call-graph, and Failure Logic Analysis (in particular FPTC and FI4FA). Once the dependability characteristics have been adequately annotated in the user model, and in particular in the dependability analysis view, a set of model-to-model transformations extract the necessary information to be inputted in a plethora of corresponding dependability analysis tools, notably Item Toolkit, PRISMA, DEEM, and so forth. In general, data elicited from models is passed in terms of XML files to the analysis tool; in turn, each tool stores computed results in XML formats that can be interpreted and mapped back to user models, also leveraging on tracing information generated in the forward direction.

2.1.3 Extension needs for CONCERTO

CONCERTO will deal with dependability aspects in the context of heterogeneous multi-core embedded systems. In this respect, it will extend CHESS state-of-the-art in both analysis approaches and modelling support in order to provide convincing evidence in compliance with the applicable safety standards. A detailed error model capable of modelling dependencies between error behaviour and conditions of the system or its environment will be needed in order to empower simulation-based approaches for the assessment of error effects.

More generally, a deep extension and refinement of state of the art error models will be required for enabling hazard analysis and risk assessment. Care will have also to be taken that these innovations in modelling and analysis techniques, including advances in state-based and failure logic based techniques, form a coherent and consistent whole that avoids concept redundancy, misconception and misuse.

Eventually, as will become evident in the requirements list illustrated later on in this document, CONCERTO will have to tackle new domain-specific dependability aspects, notably those coming from the medical and petroleum domains, that were not investigated in the realization of the CHESS modelling framework. In this respect, extensions to both the modelling language and the analysis tools are expected.

2.2 USER REQUIREMENTS

This section summarizes the user requirements that were mapped to extension of the CONCERTO framework to support selected non-functional properties and system level details. Moreover, it shows some changes produced to better clarify the separation of goals between WP3 and the interconnected WP2 and WP4.

2.2.1 Summary

Req. No.	Overall Priority	Category	Requirement	Discussed Modifications
2	SHALL	Analysis	CONCERTO shall provide techniques to specify the availability information of component and calculate system level availability.	
1	SHALL	Analysis	CONCERTO shall provide a mechanism to compute system-level response times for dedicated scenarios.	Moved to WP4
06 / 10	SHALL	Analysis	CONCERTO platform shall analyse the schedulability of the architecture, or the performance that can be expected.	Moved to WP4
14 /16	SHALL	Back propagation	CONCERTO shall support the back propagation of the results of safety analyses on model elements.	
23	SHALL	Code Generation	CONCERTO shall capture specific configuration information necessary to generate code adapted to a specific execution-platform.	Shared with WP4
17	SHALL	Code Generation	CONCERTO shall provide code generators for configuring different elements of the telecare architecture.	Shared with WP4
18	SHALL	Code Generation	CONCERTO shall support code generation for mobile devices.	Shared with WP4
80	SHALL	Code Generation + Run-time Monitoring	Run-time environments shall provide APIs and mechanisms for checking and enforcing per-software component access policies at runtime.	Moved to WP4
25	SHALL	Component Libraries	CONCERTO shall provide support to model particular HW elements as part of the execution platform like sensors, mobile local hubs, centralized decision support systems.	
68	SHALL	Methodology + M2M	CONCERTO shall support the precise definition of the workflow for the transformation from PIM to PSM.	
50	SHALL	M2M + Analysis + Back propagation	CONCERTO shall define an explicit computational model to aggregate the safety properties of the barriers.	
37	SHALL	Metamodel	Support shall be provided for all timing parameters such as period, deadline, WCET, offset and jitter.	Moved to WP4
36	SHALL	Metamodel	Some UML diagram / MARTE stereotypes shall be supported to describe activity/dependencies.	Shared only between WP2 and WP4
51	SHALL	Metamodel	For safety, the system modelling language shall have sufficient expressiveness to capture "hard" as well as "soft" barriers.	

Req. No.	Overall Priority	Category	Requirement	Discussed Modifications
27	SHALL	Metamodel	All the elements of the architecture shall have information indicating availability.	
47	SHALL	Metamodel	CONCERTO platform must support timing annotations for hard real-time tasks.	Moved to WP4 that derives reqs for WP2
44	SHALL	Metamodel	CONCERTO shall support the specification of safety reliability attributes such as failure rates, failure modes, etc... on model elements.	
46	SHALL	Metamodel	CONCERTO component model shall support the implementation of TLC systems.	Moved to WP2
39	SHALL	Metamodel	CONCERTO shall support the association, propagation and inheritance of ASIL (Automotive Safety Integrity Level) on model elements.	
53	SHALL	Metamodel	The component models should capture the dynamic aspects of the barriers systems.	
58	SHALL	Metamodel	Access permissions to system resources shall be supported as an extra-functional property.	Moved to WP4
33	SHALL	Metamodel	All system components (hardware and software) shall have their criticality specified as an attribute.	
32	SHALL	Runtime	CONCERTO shall support programs of different criticality	Moved to WP4
34	SHALL	Metamodel + M2M	CONCERTO shall support non-functional properties from the ASTRIUM developed computational model.	Moved to WP4 deriving reqs for WP2
61	SHALL	Methodology	CONCERTO shall support ASIL decomposition techniques.	
60	SHALL	Methodology	Autosar application layer development shall be mapped to the CONCERTO modelling flow.	
62	SHALL	Methodology + Metamodel + M2M + Analysis + Code generation	Matlab/Simulink use shall be supported to incorporate behavioural modelling into CONCERTO functional components.	Will derive reqs for WP2 and 4
29	SHALL	Model validation	Constraints shall be captured in a constraint language with precise semantics.	Moved to WP2
35	SHALL	Methodology +Metamodel + Analysis + M2M + Back propagation + Platform	CONCERTO shall support scheduling parameters and scheduler modelling: fixed priority, time-table driven, and composition of both (two level scheduler)	Moved to WP4
3	SHALL	Model validation	Validation of model for schedulability analysis should issue precise messages in order to understand when a model annotation is missing or model is not properly built.	Moved to WP4
76 / 63	SHALL	Model Validation	Early validation (on the PIM) of the model shall be supported, both on a functional and on an extra-functional	

Req. No.	Overall Priority	Category	Requirement	Discussed Modifications
			point of view.	
79	SHALL	Platform + Deployment view + Metamodel + Methodology	Deployment in the Telecare domain shall cover heterogeneous target platforms. From the same high-level PIM, automated generation of configuration files will be targeted for heterogeneous platforms. This implies the co-existence of several PSMs at the same time.	
89	SHALL	Run-Time Monitoring	The CONCERTO runtime shall raise an alert when a given precondition is not respected.	Moved to WP4
9	SHOULD	Analysis	It should be possible to perform flow analysis within a CONCERTO architecture.	
11	SHOULD	Back propagation	All results of availability and timing analysis should be back propagated to the front-end engineering UML models.	To be split between WP3 (availability) and WP4 (timing)
21	SHOULD	Code Generation	Containers for accessing native vehicle interfaces should be generated and should enforce access permissions to native services.	
19	SHOULD	Code Generation	CONCERTO should support code generation for decision support systems.	
20	SHOULD	Code Generation	CONCERTO should support the precise definition of the workflow for the transformation from PSM to code.	
26	SHOULD	Component Libraries	Resources used in Aerospace like ARINC653 buffers or semaphores and all NFP (size, queuing policy) and access services associated to the resources should be supported.	Moved to WP4
43	SHOULD	Metamodel	CONCERTO should support the specification of diagnostic attributes (detailing failure codes, diagnostic services, etc) for the model elements.	
42	SHOULD	Metamodel	CONCERTO should support the specification of communication attributes for model elements.	
40	SHOULD	Metamodel + M2M	CONCERTO should support the import of specific format for the specification of communication attributes.	
54	SHOULD	Methodology	Execution resources (tasks and mutexes, or others) should be allocated on component ports.	Moved to WP2
52	SHOULD	Methodology + Metamodel	The component models should capture safety properties at different levels of detail/abstraction.	
75	SHOULD	Methodology	Autosar (Run-Time Environment + Basic Software) should be used as a PSM.	Derives reqs for WP2
45	SHOULD	Methodology + Metamodel	Hierarchical composition of components should be supported.	Moved to WP2
86	SHOULD	Platform	The OSEK model of computation should be included in the PSM.	
85	SHOULD	Platform	CONCERTO PSM should support specific network communication.	

Req. No.	Overall Priority	Category	Requirement	Discussed Modifications
90	SHOULD	Run-Time Monitoring	The CONCERTO runtime should raise a warning if a deviation is detected in the specific Activities of Daily Living (ADL) sample of a patient.	Moved to WP4
41	MAY	Metamodel + M2M	CONCERTO may support the import of specific formats for the specification of diagnostic attributes.	

Table 1. Summary of users' requirements appointed to WP3.

2.2.2 Elaboration

R2 - CONCERTO shall provide techniques to process the availability information of component and calculate system level availability.

This requirement has impact on both the modelling language and analysis techniques. In order to process availability information and calculate system-level availability, first of all the CONCERTO modelling language shall provide means to specify availability information for components (R2.1). Deriving system level availability metrics requires also the ability to specify how the failures of individual components contribute to the failure of system-level functionalities. Error propagation relations between components, either at the same hierarchical level (R2.3), or from subcomponents to higher-level components (R2.4) shall be specified. Finally, the generation of an analysis model capable of actually using this information to calculate system-level metrics is needed (R2.5). Fault-tolerance mechanisms (R2.2) shall be introduced to avoid failures by stopping and or mitigating error propagation.

R1 - CONCERTO shall provide a mechanism to compute system-level response times for dedicated scenarios.

Based on the system architecture annotated with the required information CONCERTO shall be able to compute response times (e.g., WCET) for dedicated execution scenarios. This requirement has been moved to WP4 since clearly pertaining to timing characteristics of the system.

R 06 / 10 - CONCERTO platform shall analyse the schedulability of the architecture, or the performance that can be expected

See elaboration provided in deliverable D4.1. Any domain specific concerns shall be taken into account. As for R1, the requirement has been moved to WP4.

R14/16 - CONCERTO shall support the back propagation of the results of safety analyses on model elements.

Back-propagation of results of safety analysis requires algorithms to isolate specific properties or numerical values from results obtained by analysis techniques (R14.1), and modelling elements to accommodate such results as property values in the CONCERTO model (R14.2). In this respect, the CHESS Dependability View shall be aligned/enriched based on the refinement/extension of safety support provided in CONCERTO.

R 23 - CONCERTO shall capture specific configuration information necessary to generate code adapted to a specific execution-platform

The modelling language shall support the definition of configuration information for the underlying platforms (R23.1). That configuration information shall be assigned to the nodes of the platform architecture in the Deployment View and validated prior to the code generation (R23.2, R23.3). Code generators shall understand the configuration (R23.4) and generate code accordingly (to be tackled in WP4).

R17 - CONCERTO shall provide code generators for configuring different elements of the telecare architecture.

The CONCERTO modelling language shall make it possible to define, on a relatively high abstraction level, the different requirements and settings of the telecare applications (R23.1, R23.2, R23.3). CONCERTO shall provide different code generators (model processors). Based on the CONCERTO models the model processing solution shall generate the configuration files, e.g. XML files, which configure the different elements of the telecare architecture, i.e. server-side components and client-side components (R17.1).

The solution aids the definition of telecare applications related settings in one place using a comfortable user interface and language (addressed in this WP), and generates all the required configuration artefacts for the different elements of the architecture (to be addressed in WP4).

R18 - CONCERTO shall support code generation for mobile devices.

The CONCERTO model processing module makes it possible to define (R23.1, R23.2, R23.3) and execute those model processors that work from the same platform-independent software models and generate source code for different mobile platforms (R18.1). The generated source code will be based on the mobile platform-specific libraries. This means that the generated code will contain parameterized function calls targeting to the appropriate library methods. These platform-specific libraries will be developed by the mobile platform specialists and can be possibly reused by both the generated and manually written source code blocks.

R80 - Run-time environments shall provide APIs and mechanisms for checking and enforcing per-software component access policies at runtime.

This requirement has been moved to WP4, as clearly pertaining to timing characteristics of the system.

R25 - CONCERTO shall provide support to model particular HW elements as part of the execution platform like sensors, mobile local hubs, centralized decision support systems

Modelling hardware elements means that the modelling language shall be capable of defining hardware devices (like sensors and mobile local hubs as specified by the requirement) (R25.1) and network services (like centralized decision support systems)

(R25.2). To do so predefined components shall be defined in a component library and made application-independent in order to be easily reused.

Additionally, the modelling language should be extendable (e.g., extension point mechanism) in order to provide means to capture vendor-specific details (R25.3).

As an additional requirements to the allocation process due to its sensor network like structure, the telecare domain specific architecture modelling language should support the separation of functions from their executing platform elements as the functions are required to be dynamically reconfigured during operations (R25.4).

R68 - CONCERTO shall support the precise definition of the workflow for the transformation from PIM to PSM.

In Model-Driven Engineering complex transformations are typically defined as a chain of model transformation steps as it needs to bridge a large abstraction gap between the high-level platform independent models (PIM) and the platform (implementation) specific models (PSM). These model transformation steps can be either: fully automated, semi-automated or completely user-driven (R68.1).

CONCERTO shall adapt the modelling language to specify the PIM-to-PSM transformation chain as a workflow (R68.2).

R50 - CONCERTO shall define an explicit computational model to aggregate the safety properties of the barriers.

Safety properties of barriers and related risks on petroleum installations will typically depend on a number of different factors. For example, the risk of inflammable gas being released and catching fire in a given area depends on the number and location of gas detectors in the area, the type, quality and maintenance status of these detectors, the type of activity that takes place in the area (i.e. welding), the equipment in the area (as some types of equipment may produce sparks), and so on. Even if each of these factors are assigned a risk level or other kind of quantitative or qualitative value, it will be extremely hard for human operators to obtain an overall understanding of the risk level or safety barrier quality based on such a multitude of data during runtime. We therefore need to identify suitable safety properties for barriers as well as computational models to aggregate low-level information into a simple aggregated view to facilitate human decision making. Similarly to R2, it requires the ability to specify safety properties of system elements (R50.1), as well as the ability to specify fault tolerance structures and mechanisms (R2.2) and error propagation relations (R2.3 and R2.4). Finally, the generation of an analysis model capable of actually aggregating safety properties of components to system-level safety properties is needed (R50.2).

R37 - Support shall be provided for all timing parameters such as period, deadline, WCET, offset and jitter

This requirement has been moved to WP4 since pertaining to timing properties.

R36 - Some UML diagram / MARTE stereotypes shall be supported to describe activity/dependencies

Activity diagrams are already part of the CHESS modelling language. In particular in CHESS an activity diagram is associated to a given provided operation and is used to specify the operations that are called when the operation itself is executed; this information is used by analysis to retrieve the WCET closure for an operation. This requirement has been removed from WP3, since it is shared by WP2 and WP4.

R51 - For safety, the system modelling language shall have sufficient expressiveness to capture "hard" as well as "soft" barriers.

By "soft" barriers we mean human and organizational factors, such as procedures to be followed by workers and their training level, while "hard" barriers refers to physical components such as gas detectors. ISO 17776 states the following: "Barriers may be physical (materials, protective devices, shields, segregation etc.) or non-physical (procedures, inspection, training, drills, etc.)." To properly model this aspect, the CONCERTO modelling language should allow the specification of system elements that represent physical equipment and human/organizational aspects and the association of safety properties to them (R51.1).

R27 - All the elements of the architecture shall have information indicating availability.

This requirement overlaps with R2.1 and extends R2. Early model-based availability analysis requires that the modelling language used for specifying the architecture holds all availability specific information. This means that the metamodel defined for capturing the system architecture shall provide elements to specify failure rate, repair/recovery time and error propagation probability.

R47 - CONCERTO platform must support timing annotations for hard real-time tasks

This requirement will be covered by WP4, which will entail derived requirements impacting WP2, i.e. the language support provided by CONCERTO.

R44 - CONCERTO shall support the specification of safety reliability attributes such as failure rates, failure modes, etc. on model elements.

This requirement extends requirements R2.1 and R50.1, taking into account further and more detailed dependability properties of system components. For certain kind of analysis it is useful to specify more detailed dependability properties, e.g., different failure modes; the CONCERTO modelling language shall support the specification of such properties (R44.1). In supporting different kind of dependability properties, the language should provide means to avoid inconsistencies between them (R44.2). For exam-

ple, properties like *failure rate*, *repair delay*, and *availability* of a component are related and may cause inconsistencies if no constraints are imposed on the model. Similarly, consistency needs to be enforced between results back-annotated from analysis techniques, and dependability information that is later added by the modeller (R44.3).

R46 - CONCERTO component model shall support the implementation of TLC systems.

This requirement will be covered in WP2 since it pertains to modelling language matters.

R39 - CONCERTO shall support the association, propagation and inheritance of ASIL (Automotive Safety Integrity Level) on model elements.

CONCERTO shall allow ASIL property to be allocated to requirements and components (R39.1), and the ability to derive the ASIL of a model element when possible (R39.2), e.g., a component would have the higher ASIL among the ASILs associated with its functions.

When modelling a satisfy relationship from a component to a safety requirement, then the ASIL specified for the requirement shall propagate to the component.

Starting from a safety requirement (or safety goal) coming with its proper ASIL and the error models provided for the designed components and their propagation (see R44), (automated) fault tree analysis can be used (see R14/16) to understand how the failures of low-level elements can cause system level function failures. This analysis is needed to discover the criticality of the components.

See also R61.

R53 - The component models should capture the dynamic aspects of the barrier systems.

The quality of barrier systems will change over time. In some cases this may happen quickly, for example if gas detectors are bypassed because of maintenance work. In other cases, it is a more gradual process. To ensure that safety risk level or barrier quality assessments remain valid, these dynamic aspects need to be taken into consideration by allowing safety properties to depend (partly) on external data that change during the system lifetime, such as data from error and maintenance databases (R53.1).

R58 - Access permissions to system resources shall be supported as an extra-functional property.

This requirement will be covered in WP4.

R33 - All system components (hardware and software) shall have their criticality specified as an attribute.

The system shall support at least two major types of components according to their criticality – "critical" components, and "non-critical" components (R33.1). The critical components should be further subdivided using SIL (R33.2).

R32 - CONCERTO shall support programs of different criticality

It is often desired to use large pieces of software such as COTS libraries and even full Operating Systems to perform complex and intensive work that is not actually critical (it's non-critical) to the system. In this context non-critical means that there is no hard requirement for the work performed by this component to be actually done. Also due to the usual complexity of these components, they cannot be trusted to not interfere with other components. By non-trusted we understand that the component can perform the work erratically, and eventually affect the rest of the system.

Combining the two types of components must ensure that an unexpected behaviour from the non-critical component does not affect the overall health of the system, so that critical components (for example, a car ABS component) will still work as expected, and not be disturbed by the failure of the non-critical.

The complexity and effort for safety-critical or similar certification of a system that includes critical components obstacles the use of those non-critical components - certification is applied to the whole system, even if some of the components have different criticality levels.

It is however possible to have a non-critical, non-trusted component running simultaneously with other critical components if proper partitioning is in place. Spatial and time partitioning are required for full isolation between the non-critical component and the other, critical, components. Shared resource usage must also be addressed, to ensure the partitioning to a high degree of confidence. This will allow certification of the critical components only, provided that the partitioning system is also certified.

This requirement will be covered in WP4.

R34 - CONCERTO shall support non-functional properties from the ASTRIUM developed computational model.

This requirement will be covered in WP4, since referring to timing properties that will potentially impact also the expressive power of the modelling language (WP2).

R61 - CONCERTO shall support ASIL decomposition techniques.

ASIL decomposition allows for redundant elements to share the responsibility of meeting a given ASIL. ASIL decomposition allows decreasing ASILs and so development costs. Argumentation of independence shall be provided for the redundant elements, e.g. in the form of independency requirements (R61.1).

While modelling ASIL decomposition, CONCERTO modelling environment shall be able to check ISO 26262 compliant ASIL rules (R61.2).

Patterns for redundancy could be automatically provided to the modeller, allowing (semi) automatic ASILs assignment rules (R61.3).

Automotive domain-specific configuration shall be taken into account when deriving intermediate models for code generation (R61.4).

R60 - Autosar application layer development shall be mapped to the CONCERTO modelling flow.

Autosar component model shall be mapped to the CONCERTO component model (to be elaborated in D2.2) (R60.1).

This requirement is also related to R42 and R43.

R62 - Matlab/Simulink use shall be supported to incorporate behavioural modelling into CONCERTO functional components.

CONCERTO shall be able to map functional blocks modelled with Simulink to components behaviour (R62.1).

R29 - Constraints shall be captured in a constraint language with precise semantics

CONCERTO shall define a high-level, declarative constraint language that can be used to specify and validate static structural constraints over the various CONCERTO models. The aim of these constraints is to provide early model-based validation tightly integrated and executed next to the design/development process. This requirement has been moved to WP2, since pertaining to the model validation support.

R35 - CONCERTO shall support scheduling parameters and scheduler modelling: fixed priority, time-table driven, and composition of both (two level scheduler)

This requirement has been moved to WP4 since pertaining to timing issues.

R3 - Validation of model for schedulability analysis should issue precise messages in order to understand when a model annotation is missing or model is not properly built.

As R35, this requirement is moved to WP4.

R76 / 63 - Early validation (on the PIM) of the model shall be supported, both on a functional and on an extra-functional point of view.

Early validation about extra functional concerns is already covered by other requirements (e.g. see R14/16, R6/10).

R79 - Deployment in the Telecare domain shall cover heterogeneous target platforms. From the same high-level PIM, automated generation of configuration files will be targeted for heterogeneous platforms. This implies the co-existence of several PSMs at the same time.

This requirement overlaps with requirements R18 + R19, i.e., R79 will be realized based on the solutions that cover R18 + R19 (See also R23).

R89 - The CONCERTO runtime shall raise an alert when a given precondition is not respected.

In a telecare application it is important to perform measurements (e.g., blood pressure, blood glucose, weight) on the patients. These measurement processes may need specific preconditions (e.g., sitting and relaxed during the measurement) to be met to be successful. CONCERTO runtime shall support raising alerts when a given precondition is not met during a measurement. The local decision support system should be responsible for the evaluation of these measurement preconditions. This requirement has been moved to WP4 since pertaining to runtime issues.

R9 - It should be possible to perform flow analysis within a CONCERTO architecture.

The idea is to be able to analyse the data flow and control flow across the system architecture, in order to perform fault propagation analysis. To fulfil this requirement, the CONCERTO modelling language should allow both data flow (R9.1) and control flow (R9.2) to be specified in the system architecture. In order to evaluate the impact of fault propagation on the system architecture, CONCERTO should provide a way to specify faults in the system architecture (R9.3). It should then be possible to apply analysis techniques taking into account such information and providing information on fault propagation as output (R9.4).

R11 - All results of availability and timing analysis should be back propagated to the front-end engineering UML models.

Analogously to what is supported in CHESS, CONCERTO shall pursue the automated annotation of design models with results coming from availability analysis tools. Depending on the required features, back propagation transformations will be re-used or developed to trace analysis values back to the user space. The same is expected for timing analysis, which will be covered in WP4.

R21 - Containers for accessing native vehicle interfaces should be generated and should enforce access permissions to native services.

This requirement will be covered in WP4.

R19 - CONCERTO should support code generation for decision support systems.

CONCERTO code generation (model processing) solution should facilitate either by traversing-based solution (using an API to reach and process the models) or by template-based solution to support optional textual output generation from the CONCERTO software models. Decision support systems can be driven in different ways. CONCERTO should support this process by generating the appropriate input models for decision support systems from CONCERTO models.

R20 - CONCERTO should support the precise definition of the workflow for the transformation from PSM to code.

This requirement extends R28 as the language provided there should be extended to also support the definition of M2C transformation chains.

R26 - Resources used in Aerospace like ARINC653 buffers or semaphores and all NFP (size, queuing policy) and access services associated to the resources should be supported.

This requirement has been moved to WP4 since covering timing/runtime issues.

R43 - CONCERTO should support the specification of diagnostic attributes (detailing failure codes, diagnostic services, etc) for the model elements.

This requirement will be covered by investigating and potentially mapping the Autosar specification for the diagnostic attributes specification to CONCERTO (R43.1).

R42 - CONCERTO should support the specification of communication attributes for model elements.

CONCERTO should give the possibility to specify a Network Communication matrix in order to detail how functional communication (e.g. communication between the Body Computer and the Engine Control ECUs) and extra functional communication (e.g. error diagnostic messages) should be delivered (R42.1).
Autosar specification devoted to communication attributes should be checked and mapped to CONCERTO.

R40 - CONCERTO should support the import of specific format for the specification of communication attributes.

See R42. CONCERTO should support the import of DBC format specifying the CAN networks and messages infrastructure (R40.1).

R54 - Execution resources (tasks and mutexes, or others) should be allocated on component ports.

This requirement is moved to WP2 since pertaining to Deployment choices.

R52 - The component models should capture safety properties at different levels of detail/abstraction.

Safety risk levels and barrier quality can be viewed and assessed on different levels of abstraction. In some cases we want to know the overall status for the whole installation, while in other cases we are concerned about one particular aspect of a safety risk or barrier. Being able to view and present models at different levels of abstraction facilitates support for different users and contexts. This can be achieved through support of hierarchical composition of components, assuming that safety properties can be assigned to components at all levels. The requirement is therefore covered by R45.

R75 - Autosar (Run-Time Environment + Basic Software) should be used as a PSM.

An Autosar compliant Basic Software should be used in order to demonstrate the use of CONCERTO in an Autosar project, as alternative an Autosar model should be generated starting from the CONCERTO PSM (the latter in case an Autosar compliant Basic Software will not be available in the project) (R75.1).

R45 - Hierarchical composition of components should be supported.

Hierarchical composition of dependability and safety properties should be supported, e.g., it should be possible to derive failure modes of a composite components based on port delegations, and failure modes of its subcomponents. This requirement has been moved to WP2 since pertaining to metamodeling issues.

R86 - The OSEK OS model of computation should be included in the PSM.

The compatibility of CONCERTO PSM should be investigated wrt OSEK (R86.1).

R85 - CONCERTO PSM should support specific network communication.

CONCERTO PSM should support CAN, LIN, FlexRay, 1553 bus, AFDX network etc. network communication (R85.1).

R90 - The CONCERTO runtime should raise a warning if a deviation is detected in the specific Activities of Daily Living (ADL) sample of a patient.

Activities of Daily Living (ADL) in healthcare refer to daily self care activities (e.g., functional mobility, personal hygiene, eating). In a telecare application it is very important to analyse uploaded ADL data and compare it to previous ADL samples to detect deviations. The CONCERTO runtime should support raising warnings if a deviation is detected in the ADL sample. The decision support system is responsible for ADL data evaluation and the detection of deviations. This requirement has been moved to WP4 since pertaining to runtime monitoring issues.

R41 - CONCERTO may support the import of specific formats for the specification of diagnostic attributes.

Covered by R43.

2.3 DERIVED REQUIREMENTS

In this section we list the requirements derived from the users' needs recalled in the previous section. Moreover, the requirements are mapped to corresponding areas of the CONCERTO technological framework impacted by them. It is worth noting that this list ignores derived requirements coming from requirements moved to other work packages.

Req. No.	Category	Derived Requirement
R2.1	Metamodel	CONCERTO shall allow availability information to be

Req. No.	Category	Derived Requirement
		specified for system elements
R2.2	Metamodel	CONCERTO should allow redundancy and fault-tolerance mechanism to be specified
R2.3	Metamodel + Methodology	CONCERTO should allow error propagation information between system components to be specified.
R2.4	Metamodel + Methodology	CONCERTO should allow to specify/derive dependability information at system level based on information specified on subcomponents.
R2.5	M2M + Analysis	CONCERTO shall allow the evaluation of system-level availability
R14.1	Design space	CONCERTO shall define views where safety analysis results are shown
R14.2	Metamodel	CONCERTO shall provide modelling attributes to support the back-annotation of dependability properties evaluated by the analysis techniques with particular focus on availability and safety
R14.3	Backpropagation	CONCERTO shall provide back-annotation of properties evaluated by dependability analysis techniques with particular focus on availability and safety
R23.1	Design space	CONCERTO shall capture domain specific configuration information necessary to generate code for domain-specific platforms
R23.2	Deployment	Domain-specific configuration information shall be assigned to the nodes of the platform architecture in the Deployment View
R23.3	Model Validation	Domain-specific configuration information shall be validated prior any code generation step is attempted
R17.1	Code generation	Telecare domain-specific configuration shall be taken into account when deriving intermediate models for code generation
R18.1	Code generation	Mobile devices domain-specific configuration shall be taken into account when deriving intermediate models for code generation
R25.1	Metamodel + Component libraries	The modelling language shall allow defining hardware devices like sensors, mobile local hubs
R25.2	Metamodel + Component libraries	The modelling language shall allow defining network services to represent, for example, centralized decision support systems.
R25.3	Metamodel + Component libraries	The modelling language should be extendable in order to provide means to define vendor specific details.

Req. No.	Category	Derived Requirement
R25.4	Metamodel + Component libraries	The modelling language should support dynamic reconfiguration of allocations for the telecare domain.
R68.1	Methodology + M2M	The transformation chain definition language shall support: fully automated, semi-automated and completely user-driven transformation steps.
R68.2	Methodology + M2M	CONCERTO shall provide a modelling language to precisely define the PIM-to-PSM transformation chain as a workflow.
R50.1	Analysis	The analysis framework should provide a notion of safety/risk model encapsulation to support compositional safety/risk analysis from which computational models can be established.
R50.2	Analysis	The analysis framework should allow analysts to configure/define computational models for aggregation of safety properties tailored to each system at design time.
R51.1	Metamodel	The modelling language should provide constructs for associating safety/risk properties with all types of components, including components that represent physical equipment or human/organizational elements.
R27		*This requirement overlaps with R2.1*
R44.1	Metamodel	CONCERTO shall support the detailed specification of dependability attributes (with particular focus on safety/availability), such as failure rates, failure modes, etc.
R44.2	Metamodel + Model Validation	CONCERTO should ensure the consistency between dependability properties (with particular focus on safety/availability) specified at different levels of detail
R44.3	Metamodel + Back Propagation	CONCERTO should ensure the consistency between the dependability properties (with particular focus on safety/availability) back-annotated in the model as a result of the analysis, and properties which are added by the modeller.
R39.1	Metamodel	CONCERTO shall support the association of ASILs (Automotive Safety Integrity Levels) on model elements
R39.2	Metamodel + Methodology	CONCERTO shall support the derivation of the ASIL of a model element from already specified information, when possible
R53.1	Metamodel	CONCERTO shall allow the dynamic properties of barriers to be specified through parameterization in the sense that safety properties of components can depend on external data.
R53.2	Analysis	CONCERTO shall allow the dynamic properties of barriers to be checked.
R33.1	Metamodel	CONCERTO shall support at least two major types of

Req. No.	Category	Derived Requirement
		components according to their criticality – "critical" components, and "non-critical" components
R33.2	Metamodel	The critical components shall be further subdivided using SIL
R61.1	Metamodel	CONCERTO modelling language shall allow to represent argumentation of independence for the redundant elements, e.g. in the form of independency requirements
R61.2	Constraint	CONCERTO modelling environment shall be able to check ISO 26262 compliant ASIL decomposition rules
R61.3	Design space	CONCERTO shall provide patterns for redundancy allowing (semi) automatic ASILs assignment rules
R61.4	M2M	CONCERTO shall take automotive domain-specific configuration into account when deriving intermediate models for code generation
R60.1	Metamodel	Autosar component model shall be mapped to the CONCERTO component model
R62.1	Design space/ Metamodel	CONCERTO shall be able to map Simulink functional blocks to components behaviour
R9.1	Metamodel + Methodology	CONCERTO should support the specification of data flow in the system architecture
R9.2	Metamodel + Methodology	CONCERTO should support the specification of control flow in the system architecture
R9.3	Metamodel	CONCERTO should support the specification of faults for fault propagation analysis.
R9.4	M2M + Analysis	CONCERTO should provide analysis techniques to perform fault propagation analysis based on data flow and control flow information
R11.1	Back Propagation	Availability analysis results should be propagated in the user modelling space
R19.1	Code generation	*Overlaps with R17*
R20.1	Code generation	CONCERTO should support the definition of M2C transformation chains
R43.1	Metamodel/ Methodology	Autosar specification for diagnostic attributes specification should be checked and mapped to CONCERTO.
R42.1	Metamodel/ Methodology	CONCERTO should give the possibility to specify a Network Communication matrix in order to detail how functional communication (e.g. communication between the Body Computer and the Engine Control ECUs) and extra functional communication (e.g. error diagnostic messages) should be delivered (Autosar specification should be checked here).

Req. No.	Category	Derived Requirement
R40.1	Metamodel/ Methodology	CONCERTO should support the import of DBC format specifying the CAN networks and messages infrastructure.
R52		*This requirement overlaps with R2.4 and R44.1*
R52.3	Metamodel	The modelling language should allow safety properties to be associated with (and retrieved from) components at all nesting levels.
R75.1	M2M	An Autosar compliant Basic Software should be used in order to demonstrate the use of CONCERTO in an Autosar project, as alternative an Autosar model should be generated starting from the CONCERTO PSM (the latter in case an Autosar compliant Basic Software will not be available in the project)
R86.1	Platform	The compatibility of CONCERTO PSM should be investigated wrt OSEK
R85.1	Platform	CONCERTO PSM should support CAN, LIN, FlexRay, 1553 bus

Table 2. Sub-requirements derived from Table 1.

The list of derived requirements shown above is mapped on top of the CONCERTO technological framework, as shown in Figure 3.

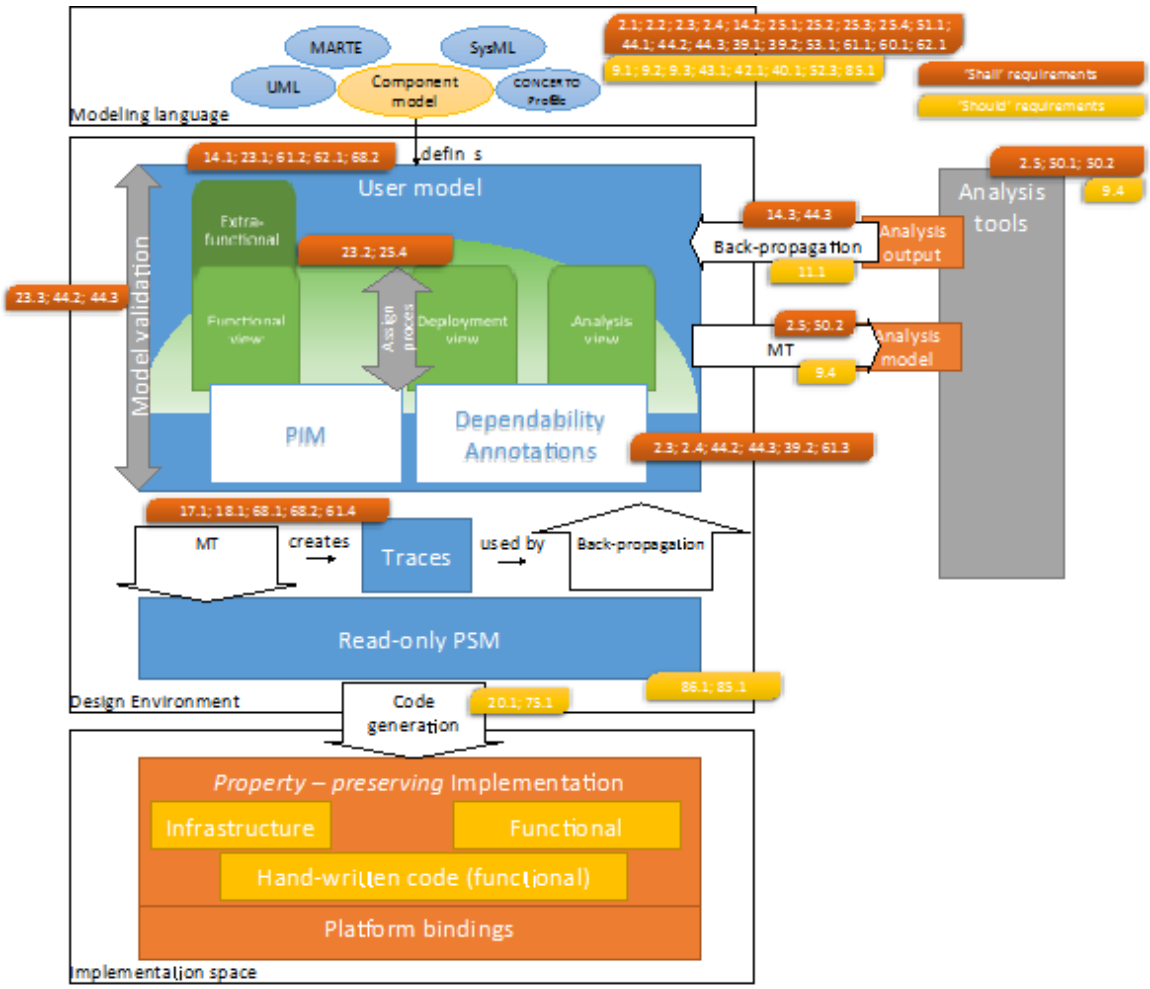

Figure 3. The derived requirements mapped on the CONCERTO framework

3. UNIFIED PRESENTATION OF FEATURES TO BE SUPPORTED

3.1 NON-FUNCTIONAL PROPERTIES

The CONCERTO modelling language is required to support the specification of dependability properties, with a particular focus on quantitative dependability properties and requirements. The current version of the language supports the specification of quantitative dependability properties (e.g., fault occurrence rates) and requirements mainly in the context of the state-based analysis technique. Extensions in CONCERTO will aim at a more integrated support for quantitative dependability properties, and a better integration with results back-annotated from the execution of analyses. Also, some integration with the specification of system requirements could be pursued.

The current version of the "Error Model" has some limitations that should be discussed and possibly overcome in CONCERTO:

- First of all, failure modes and error models themselves should be specified as "reusable" concepts. This is currently possible only to a certain extent. For

example, it should be possible to define a partition of possible failure modes on a certain interface, and apply it to all the components that implement the same interface;

- Second, some elements cannot currently be referenced in an error model associated to a component. The possibility to include them in a refinement of the CONCERTO error model should be investigated:
  - It is not possible to reference a specific failure mode occurring on a required port. Since failure modes are associated with components, when defining the error model it is not possible to know (and thus to reference) which kind of failures could occur on such port.
  - (For software components) It is not possible to reference failure modes of the hardware element on which the software element is (will be) allocated.
  - It is not possible to reference a failure mode of a subcomponent.
- Last, the error model could be improved by adding the possibility to specify common redundancy patterns (e.g., Triple Modular Redundancy, TMR).

The consistency of dependability information that is specified on CONCERTO models has to be tackled. This aspect would improve the consistency of analysis models that are generated from a CONCERTO model, and help the modeller in the specification process. To improve the consistency and harmonization of dependability properties several aspects of the modelling language and framework could be considered, e.g.:

- The "Error Model" facility and FPTC [1] annotations describe similar concepts, however currently no constraints are imposed on their consistency;
- Results that have been back-annotated from quantitative dependability analysis (e.g., state-based analysis) can be in conflict with quantitative properties specified by the modeller;
- It should be possible to automatically derive dependability properties of a composite component from the properties of its subcomponents. For example, failure modes of a composite component could be automatically derived from failure modes of its subcomponents.

The refinement of back-annotation procedure for dependability properties is needed, especially concerning the state-based analysis techniques. Currently, evaluation results for quantitative dependability properties (e.g., reliability, availability) are back-annotated in the ad-hoc model element that is used to specify the metric(s) to be analysed. Refinement will investigate more integrated approaches for attaching metrics of interest to model elements, and consequently to back-annotate the obtained results. Moreover, refinement could also consider the back-annotation of statistical properties of results (e.g., confidence interval, accuracy).

Extension and refinement of the state-based analysis technique are needed in order to: i) simplify its application, ii) integrate with other techniques, and iii) support new applicative domains. In particular, possible extensions to the current available technique are:

- Simplification of input parameters. Some of the input parameters are difficult to obtain in an industrial context, and it should be possible to run the analysis without the need to specify all of them. When possible, default values should be devised and used in the model transformation process;

- Extension/refinement of metrics that can be evaluated, also considering requirements from new domains;

- The possibility to use additional information to build the state-based analysis model, e.g., the possibility to use also FTPC annotations as input could be investigated.

The provision of work-products related to the safety processes mandated by the standards need to be supported. This includes the provision of a safety-related view allowing for modelling safety-related information necessary to perform hazards analysis as well as safety integrity level estimation. For instance the attributes needed to estimate Automotive Safety Integrity Levels (ASILs) could be supported (namely, controllability, exposure and severity).

More extensively, different views could be conceived as associated to the different phases of the safety life-cycle whenever safety-related work-products are required. These views could support the provision of cross-domain as well as domain-specific work-products.

The current dependability view, which is available within the CHESS toolset, allows safety engineers to model the failure behaviour of single (hardware/software) components and then analyse via the application of FPTC analysis (or via the application of FI^{4}FA [2], when incompletion/inconsistency/interference/impermanence-related failures might occur) the failure behaviour at system level. More specifically, FPTC (as well as FI^{4}FA) analyses how failures propagate/transform themselves throughout the system. The analysis results are then back-propagated.

Additional language constructs could be provided to support hazards analysis entirely (e.g. concepts of hazard, harm, etc.). These constructs could support cross-domain hazards analysis. Moreover, additional language constructs could be provided to support domain-specific needs.

3.2 SYSTEM MODELLING

The current support of modelling and analysis for safety concerns needs to be enhanced. In particular, the definition of safety/dependability properties shall be associated with all types of components, like software, hardware, and non-hardware/software components (e.g., organizational entities). Failure logic analysis (e.g. FPTC) could also be extended to take into consideration a broader concept for the term “component” (i.e.

by including also organizational units as well as human factor related aspects (towards an MTO model, Människa, Teknik och Organisation)). This extension would enable a safety expert to specify the failure behaviour of humans (as well as organizational units) involved in socio-technical systems. Typical failure modes conventionally used for humans should be considered and classified according to the more generic value/time/provision-based classification.

Hardware modelling capabilities need to be extended to support particular hardware elements, heterogeneous hardware and sensor systems, typically related to a specific applicative domain. As a consequence, both the deployment view and model transformations currently available will have to be enhanced to take into account (domain-specific) detailed deployment configurations and complex workflow-driven transformation chains.

4. PLAN FOR REALIZATION

Figure 4 represents the technical actions needed to provide a solution for the derived requirements discussed in Section 2.3.

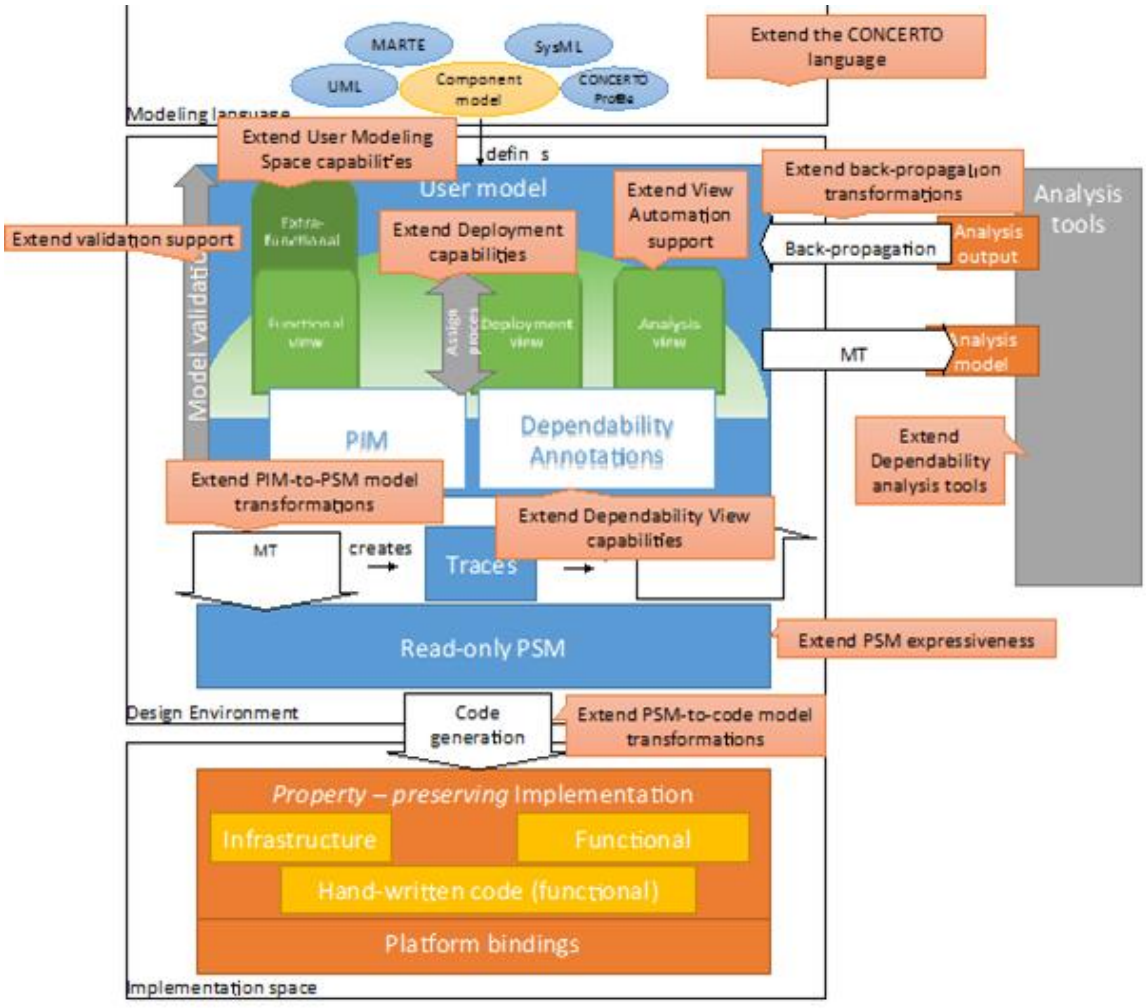

Figure 4. Visualization of impacts of required extensions on the CONCERTO framework.

In particular, required extensions and/or additional implementations are put in the context of the CONCERTO technological framework. They are classified as:

- Extensions to the CONCERTO language, when metamodel extensions/refinements have to be investigated in order to allow the specification of a certain concept;
- Extension to CONCERTO dependability view capabilities, when additional/refined properties to be analysed demand additional information to be modelled in form of decorations, or require some form of automation support;
- Extension of the CONCERTO dependability analysis capabilities, when refinements are needed in order to properly cover additional properties evaluations;
- Extensions to the user modelling space, when design space facilities need to be enhanced, notably adding new views, diagrams, import features, and so forth;
- Extension of back-propagation transformations, when transformations have to be extended/refined in order to properly propagate computed results back to the design models;
- Extension of the deployment capabilities, when additional deployment information needs to be provided;
- Extension of model validation support, when rules and constraints tailored to model validation need to be extended/refined;
- Extensions of PSM expressiveness, when the expressiveness of the read-only PSM needs to be extended/refined in order to enable additional domain-specific analysis;
- Extensions of PIM-to-PSM model transformations, when additional information has to be considered in the generation of intermediate models, often containing platform-specific details, that will be exploited to generate code later on;
- Extensions of PSM-to-code generation, when code generation support has to be extended/refined in order to take into account additional information coming from design models.

It is worth noting that in general a requirement may impact several parts of the CONCERTO technological framework. Moreover, multiple derived requirements may need the same kind of extension in order to be satisfied. In this respect, the list of actions that follows groups the requirements by the kind of extensions required to the CONCERTO framework.

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
Extend the CONCERTO language	R2.1	SHALL	CONCERTO shall allow system level availability information to be derived	UNIFI, MDH, BME, INT	INT: metamodel support MDH: Failure logic analysis when omission failures are involved
	R2.2	SHALL	CONCERTO shall allow system level redundancy and fault-tolerance mechanism to be specified	UNIFI	UNIFI: modelling support, extensions to the error model feature
	R2.3	SHALL	CONCERTO shall allow error propagation information between system components to be specified.	UNIFI MDH	UNIFI/MDH: modelling support, extensions to the error model and propagation specification.
	R2.4	SHALL	CONCERTO shall allow to specify/derive system level dependability information based on information specified on subcomponents.	UNIFI MDH	UNIFI/MDH: investigate which dependability/safety properties of composite components can be derived from their subcomponents.
	R14.2	SHALL	CONCERTO shall provide modelling attributes to support the back-annotation of dependability properties evaluated by the analysis techniques with particular focus on availability and safety	UNIFI, MDH, INT, CSW	INT: metamodel support UNIFI: better support for back-annotation of results from state-based analysis, improve consistency with properties specified at design time. Extend support for back-annotation of results from failure logic analysis, improve consistency with properties specified at design time.
	R25.1	SHALL	The modelling language shall allow defining hardware devices like sensors, mobile local hubs	INT	INT: metamodel support
	R25.2	SHALL	The modelling language shall allow defining network services to represent, for example, centralized decision support systems.	INT	INT: metamodel support
	R25.3	SHALL	The modelling language shall be extendable in order to provide means to define vendor specific details.	INT	INT: metamodel support
	R25.4	SHALL	The modelling language shall support dynamic	INT, BME	INT: metamodel support BME: case study and metamodel

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			reconfiguration of allocations for the telecare domain.		extension
	R51.1	SHALL	The modelling language should provide constructs for associating safety/risk properties with all types of components, including components that represent physical equipment or human/organizational elements.	UNIFI, MDH, SINTEF, INT	INT: metamodel support SINTEF: identification and definition of relevant safety/risk properties for the petroleum domain (which are likely of relevance also in other domains). UNIFI/MDH: investigate safety properties for these kinds of components.
	R44.1	SHALL	CONCERTO shall support the detailed specification of dependability attributes (with particular focus on safety/reliability/availability), such as failure rates, failure modes, etc.	UNIFI, MDH, INT, CSW	UNIFI/MDH: investigate the dependability properties to be attached to components.
	R44.2	SHALL	CONCERTO should ensure the consistency between dependability properties (with particular focus on safety/reliability/availability) specified at different levels of detail	UNIFI, MDH, INT, CSW	UNIFI/MDH: identify possible sources of inconsistency in dependability information attached to components in the CONCERTO ML
	R44.3	SHALL	CONCERTO should ensure the consistency between the dependability properties (with particular focus on safety/reliability/availability) back-annotated in the model as a result of the analysis, and properties which are added by the modeller.	UNIFI, MDH, INT, CSW	UNIFI/MDH: identify possible sources of inconsistency in dependability information attached to components and information back-annotated from analyses methods.
	R39.1	SHALL	CONCERTO shall support the association of ASILs (Automotive Safety Integrity Levels) on model elements	UNIFI, MDH, INT, CSW, ISEP	UNIFI: Investigate the possibility to integrate SIL/ASIL specification with state-based analysis. MDH: Investigate the possibility to integrate SIL/ASIL specification with failure logic analysis.
	R39.2	SHALL	CONCERTO shall support the derivation of	MDH, INT	MDH: according to ISO 26262 requirements, an ASIL value should

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			the ASIL of a model element from already specified information, when possible		be derived from information related to severity, controllability, exposure a function will be implemented.
	R53.1	SHALL	CONCERTO shall allow the dynamic properties of barriers to be specified through parameterization in the sense that safety properties of system components can depend on external data	SINTEF, INT, CSW	INT: metamodel support SINTEF: definition of suitable techniques/formats for parameterization
	R61.1	SHALL	CONCERTO modelling language shall allow to represent argumentation of independence for the redundant elements, e.g. in the form of independency requirements	INT	
	R60.1	SHALL	Autosar component model shall be mapped to the CONCERTO component model	INT	
	R62.1	SHALL	CONCERTO shall be able to map Simulink functional blocks to components behaviour	INT, TCS	
	R9.1	SHOULD	CONCERTO should support the specification of data flow in the system architecture	UNIFI, INT, TCS	INT: metamodel support TCS: expertise on the modelling
	R9.2	SHOULD	CONCERTO should support the specification of control flow in the system architecture	UNIFI, INT	INT: metamodel support
	R9.3	SHOULD	CONCERTO should support the specification of faults for fault propagation analysis.	UNIFI, MDH, INT	To some extent, faults can already be specified in the CHESS language. INT: metamodel support
	R43.1	SHOULD	Autosar specification for diagnostic attributes specification should be checked and mapped to CONCERTO.	INT, CSW	
	R42.1	SHOULD	CONCERTO should give the possibility to	INT, ISEP	ISEP is interested in using those information for the schedulability

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			specify a Network Communication matrix in order to detail how functional communication (e.g. communication between the Body Computer and the Engine Control ECUs) and extra functional communication (e.g. error diagnostic messages) should be delivered (Autosar specification should be checked here).		analysis developed in WP4
	R40.1	SHOULD	CONCERTO should support the import of DBC format specifying the CAN networks and messages infrastructure.	INT, ISEP	ISEP is interested in using those information for the schedulability analysis developed in WP4
	R52.3	SHOULD	The modelling language should allow safety properties to be associated with (and retrieved from) components at all nesting levels.	INT, SINTEF	SINTEF: identification and definition of properties for the petroleum domain (which are likely of relevance also in other domains).
	R85.1	SHOULD	CONCERTO PSM should support CAN, LIN, FlexRay, 1553 bus	INT, ISEP	
Extend the CONCERTO Dependability View capabilities	R2.3	SHALL	CONCERTO shall allow error propagation information between system components to be specified.	UNIFI, MDH, INT	INT: metamodel support UNIFI/MDH: extensions to the error model and propagation properties at system-level
	R2.4	SHALL	CONCERTO shall allow to specify/derive dependability information based on information specified on subcomponents.	UNIFI, MDH	UNIFI/MDH: extend the dependability view with support for automated presentation of dependability properties that can be automatically derived from subcomponents.
	R44.2	SHALL	CONCERTO should ensure the consistency between dependability properties (with particular focus on safety/reliability/availability) specified at different	UNIFI, MDH, INT	UNIFI/MDH: Investigate which properties need to be checked on the model in order to verify consistency of dependability information

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			levels of detail		
	R44.3	SHALL	CONCERTO should ensure the consistency between the dependability properties (with particular focus on safety/reliability/availability) back-annotated in the model as a result of the analysis, and properties which are added by the modeller.	UNIFI, MDH, INT	UNIFI/MDH: Investigate which properties need to be checked on the model in order to verify consistency of dependability information
	R39.2	SHALL	CONCERTO shall support the derivation of the ASIL of a model element from already specified information, when possible	MDH, INT	MDH: according to ISO 26262 requirements, an ASIL value should be derived from information related to severity, controllability, exposure a function will be implemented.
	R61.3	SHALL	Patterns for redundancy could be automatically provided, allowing (semi) automatic ASILs assignment rules.	INT, UNIFI	
Extend the CONCERTO Dependability Analysis capabilities	R2.5	SHALL	CONCERTO shall allow the evaluation of system-level availability	UNIFI, BME, AEN	UNIFI: Extend/refine state-based analysis to perform availability evaluation considering the new domains and properties
	R50.1	SHALL	The analysis framework should provide a notion of safety/risk model encapsulation to support compositional safety/risk analysis from which computational models can be established.	UNIFI, MDH, SINTEF	SINTEF: identification /definition of a suitable encapsulation approach. UNIFI: Extend the state-based analysis to support the evaluation of safety/risk properties MDH: Extend the failure logic analysis to support the evaluation of safety/risk properties
	R50.2	SHALL	The analysis framework should allow analysts to configure/define a concrete computational model/formula at design time for aggregation of safety properties specifically tailored to the particular system under analysis.	UNIFI, SINTEF	UNIFI: Investigate the application of the state-based analysis technique, possibly with ad-hoc extensions, as an aggregation model, and identification of its aspects that can be parameterized for different systems. SINTEF: identification/definition of aggregation models that can be parameterized.
	R9.4	SHOULD	CONCERTO should	UNIFI,	UNIFI: Software FMEA through

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			provide analysis techniques to perform fault propagation analysis based on data flow and control flow information	MDH	model execution and model-level fault-injection (as a research opportunity). MDH: Failure logic analysis
Extend the CONCERTO User Modeling Space	R14.1	SHALL	CONCERTO shall define views where safety analysis results are shown	UNIFI, MDH, INT	UNIFI/MDH: Identify how results of safety analysis can be presented to the user
	R23.1	SHALL	CONCERTO shall capture domain specific configuration information necessary to generate code for domain-specific platforms	INT	INT: metamodel support, OSEK code generation
	R61.2	SHALL	CONCERTO modelling environment shall be able to check ISO 26262 compliant ASIL decomposition rules	INT	
	R62.1	SHALL	CONCERTO shall be able to map Simulink functional blocks to components behaviour	INT, MDH	
	R68.2	SHALL	CONCERTO shall provide a modelling language to precisely define the PIM-to-PSM transformation chain as a workflow.	BME, MDH	BME metamodel definition
Extend the CONCERTO back-propagation capabilities	R14.3	SHALL	CONCERTO shall provide back-annotation of properties evaluated by dependability analysis techniques with particular focus on availability and safety	UNIFI, MDH, BME	UNIFI: Improve the back-annotation of results obtained from state-based analysis. Extend the back-annotation of results obtained from failure logic analysis. BME: definition of query-based traceability support for back-annotation
	R44.3	SHALL	CONCERTO should ensure the consistency between the dependability properties (with particular focus on safety/reliability/availability) back-annotated in the model as a result of	UNIFI, MDH, INT	UNIFI: Investigate the impact of properties back-annotated from state-based analysis on other dependability information which may be already present in the model MDH: Investigate the impact of properties back-annotated from

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			the analysis, and properties which are added by the modeller.		failure logic analysis on other dependability information which may be already present in the model
	R11.1	SHOULD	Availability analysis results should be propagated in the user modelling space	UNIFI, MDH	UNIFI: Extend back-annotation of results from the state-based analysis technique UNIFI: Extend back-annotation of results from the failure logic analysis technique (when omission failures are involved)
Extend the CONCERTO Deployment View	R23.2	SHALL	Domain-specific configuration information shall be assigned to the nodes of the platform architecture in the Deployment View	INT	INT: support for OSEK code generation
	R25.4	SHALL	The modelling language shall support dynamic reconfiguration of allocations for the telecare domain.	INT, BME	INT: metamodel support BME case study and metamodel evaluation
Extend the CONCERTO Model Valida-tion support	R23.3	SHALL	Domain-specific configuration information shall be validated prior any code generation step is attempted	INT	INT: support for OSEK code generation
	R44.2	SHALL	CONCERTO should ensure the consistency between dependability properties (with particular focus on safety/reliability/availabi lity) specified at different levels of detail	UNIFI, MDH, INT	UNIFI/MDH: Investigate which properties need to be checked on the model in order to verify consistency of dependability information
	R44.3	SHALL	CONCERTO should ensure the consistency between the dependability properties (with particular focus on safety/reliability/availabi lity) back-annotated in the model as a result of the analysis, and properties which are added by the modeller.	UNIFI, MDH, INT	UNIFI/MDH: Investigate which properties need to be checked on the model in order to verify consistency of dependability information

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
Extend the CONCERTO PSM	R86.1	SHOULD	The compatibility of CONCERTO PSM should be investigated with respect to OSEK	INT	
	R85.1	SHOULD	CONCERTO PSM should support CAN, LIN, FlexRay	INT	
Extend the CONCERTO PIM to PSM transformations	R17.1	SHALL	Telecare domain-specific configuration shall be taken into account when deriving intermediate models for code generation	BME, AEN	BME&AEN: metamodel support
	R18.1	SHALL	Mobile devices domain-specific configuration shall be taken into account when deriving intermediate models for code generation	BME, AEN	BME&AEN: metamodel support
	R68.1	SHALL	The transformation chain definition language shall support: fully automated, semi-automated and completely user-driven transformation steps.	BME, MDH	BME metamodel definition
	R68.2	SHALL	CONCERTO shall provide a modelling language to precisely define the PIM-to-PSM transformation chain as a workflow.	BME, MDH	BME metamodel definition
	R61.4	SHALL	Automotive domain-specific configuration shall be taken into account when deriving intermediate models for code generation	INT	
Extend the CONCERTO PSM to Code transformations	R20.1	SHOULD	CONCERTO should support the definition of M2C transformation chains	BME	The same language will be used as for R68.2
	R75.1	SHOULD	An Autosar compliant Basic Software should be used in order to demonstrate the use of CONCERTO in an Autosar project, as	INT	

Coarse-Grained Action	Req. No.	Overall Priority	Derived Requirement	Involved Partners	Comments
			alternative an Autosar model should be generated starting from the CONCERTO PSM (the latter in case an Autosar compliant Basic Software will not be available in the project)		

Table 3. Analysis of derived requirements and mapping with actions

5. CONCLUSION

This deliverable illustrated the needs and corresponding solution directions to cover WP3 tasks. In particular, user requirements pertaining to WP3, i.e. the work package addressing non-functional and system properties support, have been deeply revisited (as coming from WP1) and decomposed in more specific derived requirements. In some cases, requirements have been reallocated to other work packages in order to make WP tasks more homogeneous. Eventually, derived requirements have been mapped to corresponding concrete actions, each of which addressing a specific extension of the CHESS project heritage. The list of actions and corresponding involved partners serves as a plan for realization of the user needs, which will be covered in deliverables D3.2 and D3.3.

6. REFERENCES

1. B. Gallina, M. Atif Javed, F. Ul Muram and S. Punnekkat. Model-driven Dependability Analysis Method for Component-based Architectures. In proceedings of the Euromicro-SEAA Conference, IEEE Computer Society, ISBN 978-1-4673-2451-9, Cesme, Izmir, Turkey, September, 2012.

2. B. Gallina, S. Punnekkat. FI^{4}FA: A Formalism for Incompletion, Inconsistency, Interference and Impermanence Failures Analysis. International workshop on Distributed Architecture modeling for Novel Component based Embedded systems (DANCE) at Euromicro-SEAA, pp. 493-500, IEEE Computer Society, ISBN 978-1-4577-1027-8, Oulu, Finland, September, 2011.