Testing Cyber-Physical Systems under Uncertainty: Systematic, Extensible, and Configurable Model-based and Search-based Testing Methodologies

D 2.1 - Report on Uncertainty Modelling Framework V.1

Project Acronym	U-TEST	**Grant Agreement Number**		H2020-ICT-2014-1. 645463	
Document Version	0.4	**Date**	2016-06-30	**Deliverable No.**	2.1
Contact Person	Christian Hein	**Organisation**		Fraunhofer FOKUS	
Phone	+49 30 3463 7375	**E-Mail**		christian.hein@fokus.fraunhofer.de	

Document Version History

Version No.	Date	Change	Author(s)
0.1	2015-11-05	Initial document outline	Christian Hein
0.2	2016-06-206	Integration of contributions from partner	All
0.3	2016-06-21	Internal review version	All
0.4	2016-06-27	Addressed internal review comments – Final Version	All

Contributors

Name	Partner	Part Affected	Date
Christian Hein	Fraunhofer FOKUS	All	2015-11-05
Georgiana Copil	TUW	Section 3 -5, Appendix	2016-15-06
Daniel Moldovan	TUW	Section 3-5, Appendix	2016-15-06
Marc-Florian Wendland	Fraunhofer FOKUS	All	2016-17-06
Max Bureck	Fraunhofer FOKUS	All	2016-17-06
Shaukat Ali	SRL	Section 3-5, Appendix	2016-17-06
Man Zhang	SRL	Section 3- 5, Appendix	2016-17-06
Tao Yue	SRL	Section 3- 5, Appendix	2016-17-06
Phu Hong Nguyen	SRL	Section 3- 5, Appendix	2016-17-06

Table of Contents

1 Executive Summary

This deliverable describes the Uncertainty Modelling Framework UMF. The UMF provides a specification of uncertainty concepts as Unified Modelling Language (UML) profile. In addition to the profile a modelling methodology has been defined in order to guide test engineers through the UMF components. Furthermore, this deliverable defines the test-ready models of the U-Test pilot cases. The deliverable contains two report documents (this report and the uncertainty profile specification) as well as the source files of the produced models.

2 Introduction

This report describes the Uncertainty Modelling Framework (UMF). The overall idea of UMF is to provide an approach to create and specify test-ready models based on existing modeling and testing standards. In the context of the U-Test project these models are enriched by a specific uncertainty profile, which provides all relevant semantic to describe uncertainty at the three levels: application, infrastructure and integration. This report provides the first version of the UMF. It is planned to provide further iteration and refinements in future U-Test reports.

2.1 U-Test Workflow

The UMF is embedded in a general U-Test workflow, which is shown in Figure 1.

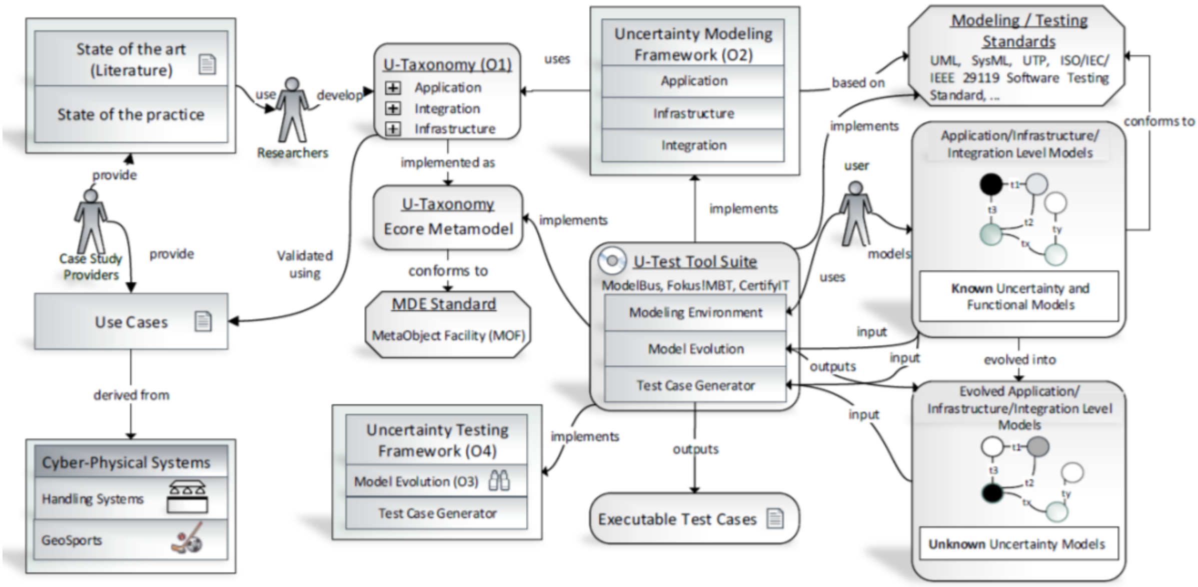

Figure 1 U-Test workflow

The overall workflow starts with a top-down and bottom up approach. The analysis of the state of the art and state of practice are part of top-down approach in order to acquire knowledge about uncertainties, which is not reflected in the U-Test use cases. The analysis of the use cases is part of the bottom up approach to detect the case study specific uncertainty requirements. As a result of this approach the U-Taxonomy was defined. The UMF uses the concepts specified in the U-Taxonomy in order to define a modelling approach based on existing standards and in particular on an uncertainty profile. The output of the UMF is test-ready models based on this uncertainty profile. These models are the input to the Uncertainty Testing Framework UTF (WP4). The UTF offers corresponding uncertainty test case generators to generate and execute adequate test cases for the U-Test use cases. Finally, the test execution results are used to evolve unknown uncertainty information (before the U-Test workflow) into known uncertainty functional models (after a walk-through to the U-Test workflow). This means that formerly unknown uncertainty behavior went into known (uncertainty) behavior.

2.2 Objectives of the Deliverable

This report is a deliverable of work package "Developing Modeling Methodologies for Uncertainty Testing" (WP2) of the U-Test project. The overall objective is to provide a systematic way of standard-based holistic modelling of uncertainties and uncertain behaviors in CPS at three levels application (task 2.1), infrastructure (task 2.2) and integration (task 2.3).

Therefore, this deliverable will report the results of the modelling support (uncertainty profile and methodology) developed in all levels. Moreover, this deliverable will provide a description of the defined test-ready models of the two U-Test uses cases at the application, infrastructure, and integration level.

2.3 Structure of the Deliverable

As mentioned in previous section the UMF has relationships with other U-Test deliverables and work packages. In particular, the specification of the uncertainty requirements from both U-Test use cases (D1.1) [2] as well as the semantic concepts summary in the uncertainty taxonomy report (D1.2) [3] are major input to the UMF. In addition to that, the output of the UMF will be test-ready models of the use cases that will be further processed in the UTF. Figure 2 depicts the relationships and the structure of this deliverable.

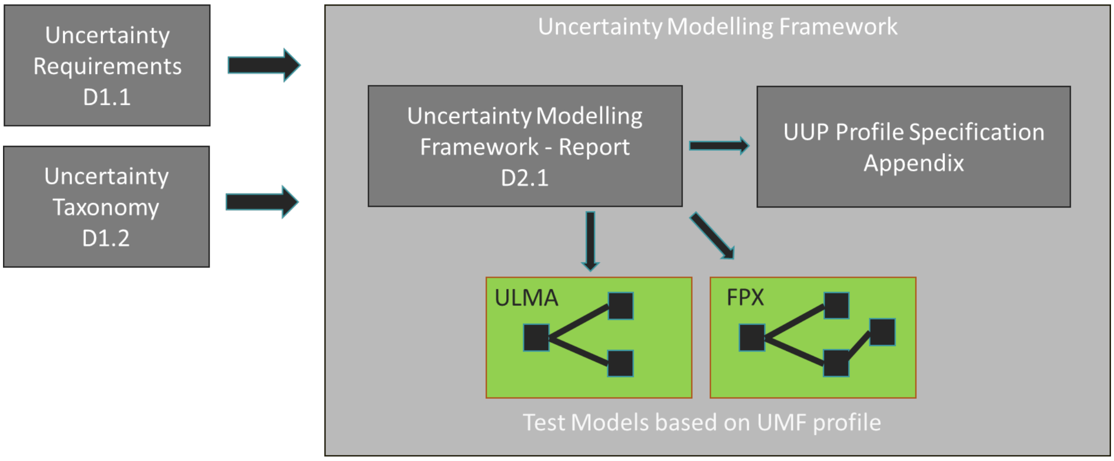

Figure 2 UMF structure and relationships

The deliverable consists of two documents on the one hand, this report describes the Uncertainty Modelling Framework architecture and it provides a detailed description of the defined test-ready models. On the other hand, a detailed description of the uncertainty profile is given in separate document (appendix) [4]. The reason for this is the nature of the uncertainty profile, because its intention is to provide a detailed UML profile specification. Whereas, the purpose of D2.1 is to provide a report on the modelling support work. In addition to that, the deliverable also provides the source files of the RSA/Papyrus models. Thus, the deliverable is separated as follows.

D1.2 – Report on Uncertainty Modelling Framework	This report document describing the UMF overview and the test-ready models
D1.2 – Report on Uncertainty Modelling Framework - PART A [4]	The detailed description of the uncertainty profile and the modelling methodology
D1.2 – Report on Uncertainty Modelling Framework – PART B [5]	The source files from Rational Software Architect (RSA) and Papyrus containing the model specification and diagrams

Therefore, this report gives an overview of the requirements to the UMF. It describes the UMF architecture and the overall uncertainty profile design. Furthermore, it detailed the test-ready models based on the uncertainty profile for both pilot cases in the different abstraction levels.

3 Requirements to Uncertainty Modelling Framework

The UMF realizes several requirements, which are presented in this section. In general, the UMF shall provide a modelling approach and modelling support in order to increase the knowledge of uncertainties. More knowledge mitigates the risk of uncertainty behavior which may have a serious impact on the cyber physical systems and its environment.

3.1 Purpose of UMF

The general purpose of UMF is to deal with known and unknown uncertainties in the context of testing of cyber-physical systems. Therefore, we identified the following purposes of the UMF.

- Establishing awareness of uncertainty
- Consideration of relevant uncertainty taxonomies (uncertainty depends on knowledge)
- Discovery of different types of uncertainty
- Traceability of uncertainty concepts with design/system and test subjects
- Quantification of uncertainties and their risk / impact to the CPS
- Management of uncertainties
- Quantification of system quality

3.2 Intended Application of UMF

Basically, the UMF is designed to be used also outside the U-Test project. The idea is to provide a generic framework which intention is to be applied easily in model-based testing approaches and processes. For this reason, we identified the following objectives for an application of UMF in research and industrial contexts:

- Lightweight integration into existing engineering environments and processes
- Traceability for uncertainty aspects
- Modelling support for quality engineers in order to manage and mitigate uncertainties
- Design recommendation and methodology for uncertainty modelling
- Support to provide test-ready models for test case generation to validate specific uncertainty behaviors

3.3 General Requirements

Besides the UMF purpose and usage in other environments, we identified also some general requirements with respect to the UMF. These requirements are listed below.

- The UMF shall base on existing standard-based approach such as UML Testing Profile, ISO/IEC/IEEE 29119 Software Testing standard, UML, SysML, MARTE, MOF, OSLC, TDL, ISO's Risk Management (ISO 31000) and Assessment (ISO/IEC 31010:2009), IEEE Standard Anomalies with minimal extensions.
- The UMF shall support a holistic modeling approach that support modelling in all aspects of the CPS System Under Test (SUT) in application, infrastructure, and integration level.
- The UMF shall be extensible. It shall provide extension points to add new libraries. For instance, it shall be possible to add new model libraries for risk calculation and uncertainty measurements based on various theories.
- The UMF shall be configurable for different applications and purposes. This means that UMF shall provide specific model components (e.g., Class diagrams or State Machines) which are pre-defined for common behaviors of CPS at infrastructures level. These pre-defined common behaviors shall be configured within the execution of the U-Test workflow for specific use cases and infrastructure environments.

4 Uncertainty Modeling Framework

The following sections give an overview of the UMF architecture and its components. The uncertainty profile is an essential component of the UMF. A detailed specification of this profile is given in the separate document [4].

4.1 Overview

The following Figure 3 depicts an architectural overview of the UMF as well as the inputs and outputs of it.

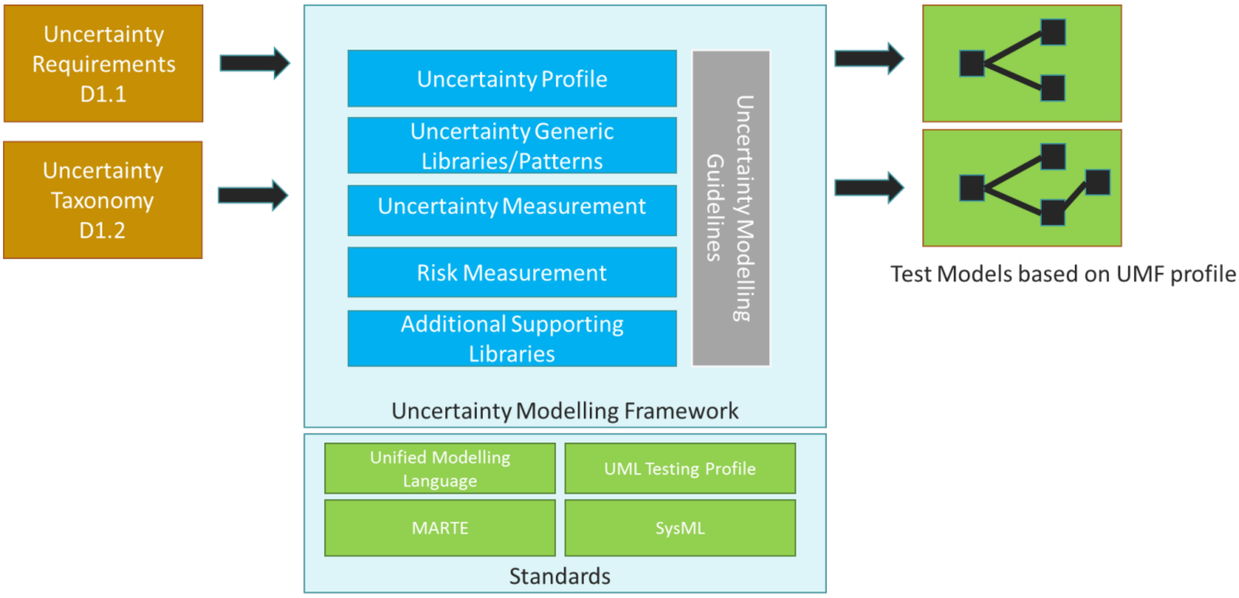

Figure 3 UMF Architecture

The input to the UMF is shown on the left hand side. In general, the uncertainty requirements defined by analyzing specific use cases is a major input the UMF. In addition to that, generic uncertainty concepts are also input to the UMF architecture. Furthermore, the UMF is built on well-established and widely accepted modelling standards such as UML, UTP, MARTE and SysML, rather than defining own proprietary concepts. Therefore, the UMF defines an UML profile that deals with the definition, management und specification of uncertainties. The profile also provides guidance for engineers in order to ease the usage and application of the uncertainty profile. The output of the UMF are test-ready models based on UML, UTP and the defined uncertainty profile. The test-ready models can be further processed by the uncertainty testing framework (UTF).

4.2 UMF components

This sections gives a summary of the individual UMF components. The profiles and the modelling methodology are described in detail in the appendix document to this report.

4.2.1 UML Uncertainty Profile

The UML Uncertainty Profile (UUP) is the general profile for all U-Test profile. It contains the core profile, the belief profile as well as the profile for application and infrastructure level. The core profile is reused by all other U-Test profile. Figure 4 shows the UUP and its relations to existing standards and model libraries.

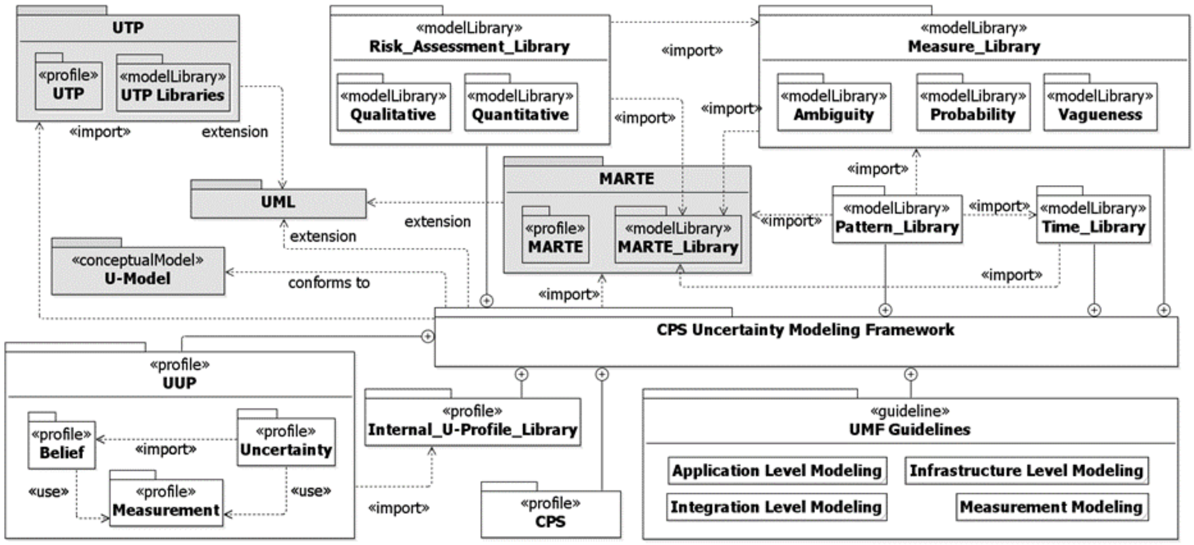

Figure 4 UU Profile

4.2.1.1 *Core UUP Profile*

The uncertainty profile is the core profile. It defines the core concepts such as Uncertainty, Cause or Effect. The core profile is shown in Figure 5. For the complete documentation of Uncertainty Modeling Framework at the Integration Level, please check the attached technical report or online one [6].

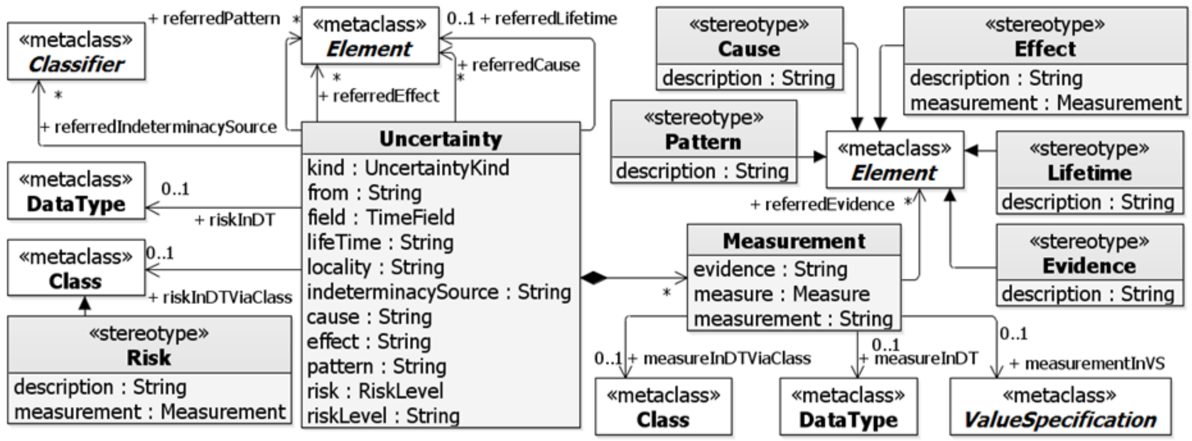

Figure 5 Uncertainty Profile

4.2.1.2 *Application Level*

The Application Level Uncertainty concepts are depicted in Figure 6. The core element is the ApplicationLevelUncertainty stereotype. It details the uncertainty concept from core profile.

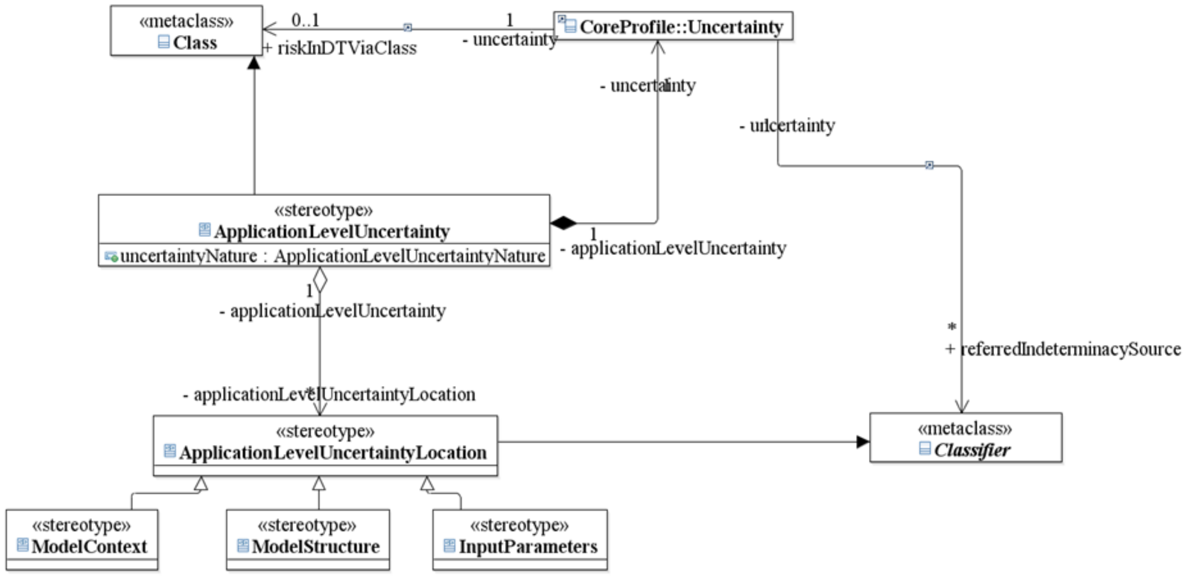

Figure 6 Application Level Profile

In general, the application level is targeting uncertainties in the environment of the CPS. Therefore, it provides concepts to specify the environment and location of uncertainties. For instance, a test modeler can indicate that an uncertainty of the CPS is caused by a specific entity in the physical environment of the SUT. In addition to that, the application level profile extends also the core concept of “Effect” in order to detail the potential impact of a certain uncertainty effect.

4.2.1.3 Infrastructure Level

The Infrastructure Level Uncertainty profile depicted in Figure 7 captures the infrastructure-related uncertainty concepts initially defined in the Infrastructure Level Uncertainty Domain Model. The Infrastructure Level Uncertainty profile relies on concepts from the Uncertainty Core Profile. Namely, the Uncertainty class from the Core Profile is extended by the Infrastructure Level Uncertainty stereotype, and the Effect stereotype from the Core profile is extended by an Infrastructure Level Effect. Each Infrastructure Level Uncertainty can belong to one or more Uncertainty Family instances. It has one or more Infrastructure Level Effects, a Nonfunctional or Functional Dimensionality to which the effect applies. Multiple Cause specifications can be associated to an uncertainty, along Locality information, Temporal Manifestation specification, and Observation Time. An uncertainty can further affect the Ingress/Egress of a system. The supported Uncertainty Families and specific types used in the profile are described in the Infrastructure Internal Profile Library. The Infrastructure Level Uncertainty profile is described in detail in the annex [4].

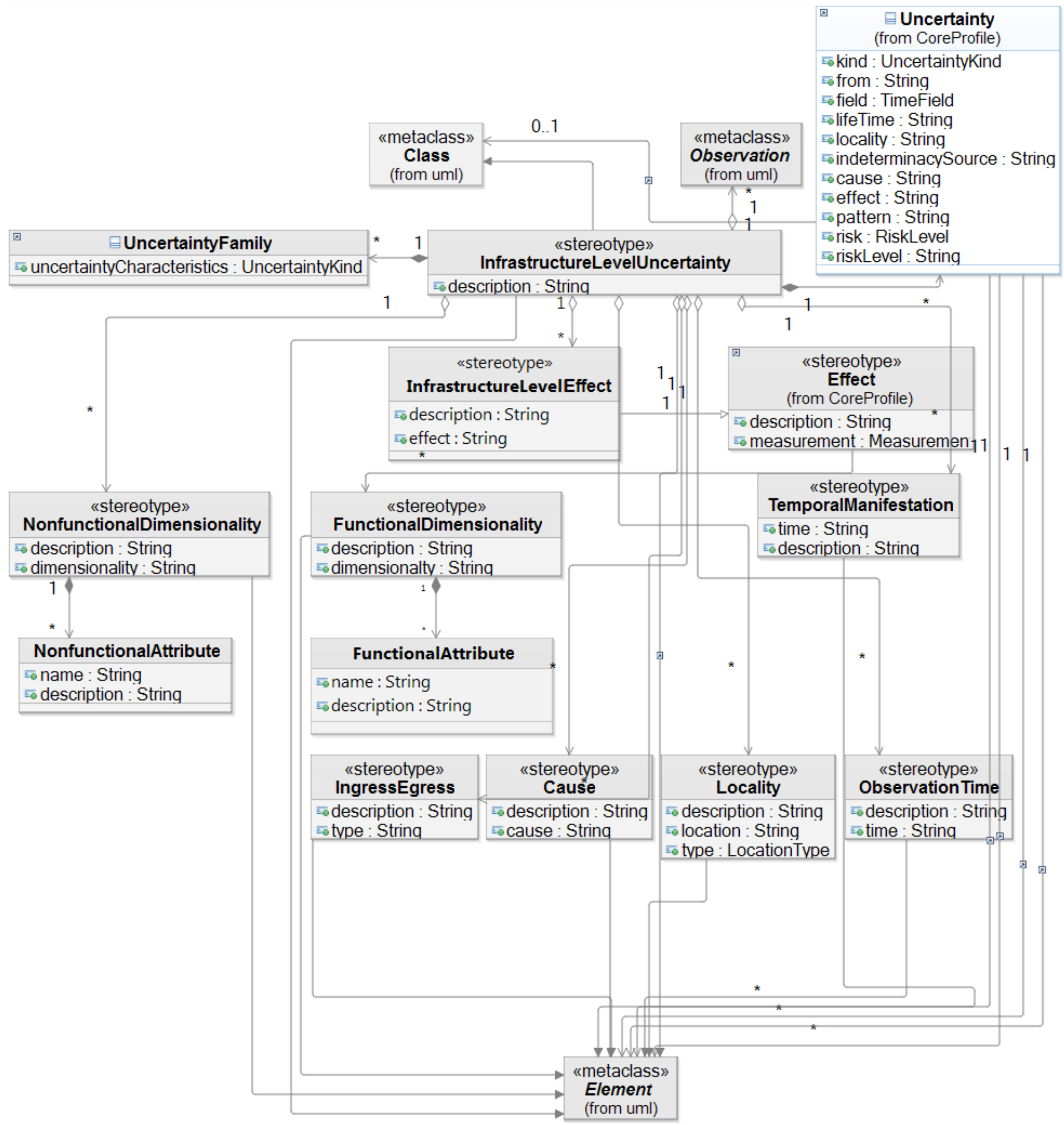

Figure 7 Infrastructure Level Uncertainty profile

4.2.2 Model Libraries

Besides the definition of core concepts, the UMF provides also model libraries which contain generic concepts for uncertainty measurement or risk analysis. Some of these libraries are explained in this section.

4.2.2.1 CPS Profile

The Cyber-Physical Systems (CPS) profile centers on the CPS class. The physical and virtual components of the CPS are modelled symmetrically, for being able to uniformly analyze them.

The CPS is composed of multiple `PhysicalUnit` and `VirtualUnit`. Each of them are complex and can be subsequently composed of other `PhysicalUnits` and respectively `VirtualUnits.` The `PhysicalUnit` has a `location`, `description` and `id` associated. It extends the `InfrastructureElement` stereotype, the metaclass `Class` and the metaclass `Device`. The `VirtualUnit` has a `location`, `description` and `id` associated. It extends the metaclass `Class` and the metaclass `Component`.

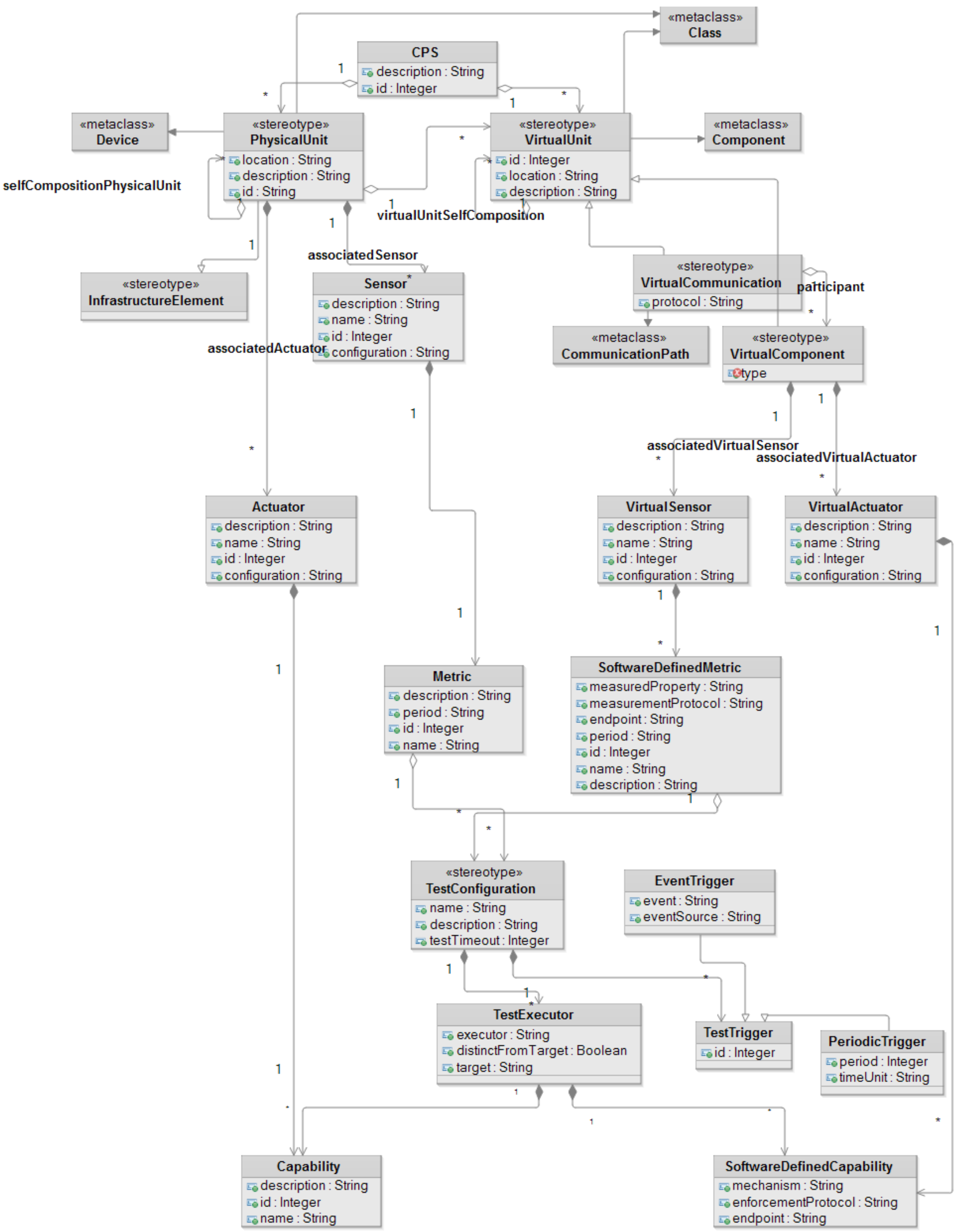

Figure 8 CPS Infrastructure Profile

Physical Unit

The PhysicalUnit has associated an Actuator and a Sensor.

The Actuator describes the component through which the physical unit changes its environment (e.g., light switch, or locks, rig controller). It has an id, name, description, and configuration. The configuration represents the current settings associated to the physical unit. Each actuator has associated a Capability, describing various capabilities of the actuator. One actuator can have multiple capabilities, for instance, a thermostat could have the capability of setting the temperature and the capability of setting the humidity.

The Sensor describes the component through which the physical unit monitors its environment (e.g., location tracker, temperature sensor, humidity sensor). It has an id, name, description, period and configuration. The configuration represents the current settings associated to the physical unit. Each sensor has associated one or more metrics, which it is collecting. The Metric consists of the id, name, description and a period. The period represents the rate at which the sensor sends data for the respective metric.

Each Physical Unit has associated one or more virtual units, that run on top of it.

We consider that the communication devices, at the physical level, can still be modelled as physical units (e.g., routers, or cables). For the virtual layer of the communication, we can associate virtual units, more specifically, VirtualCommunication which details the software-related aspects of the communication devices.

Virtual Unit

The Virtual Unit can be either simple, or composite, composed of other Virtual Units. The Virtual Unit is composed of the Virtual Communication and Virtual Component stereotypes. The VirtualCommunication stereotype extends the metaclass CommunicationPath, and the VirtualUnit stereotype. Each VirtualCommunication has associated at least two participants of the type VirtualComponent.

Each VirtualComponent has associated a VirtualSensor and a VirtualActuator.

The Virtual Actuator, is, similarly with the physical actuator, the mean through which change can be inflicted. As opposed to the physical actuator, the Virtual Actuator is a software component. The Virtual Actuator has associated a description, name and configuration. The configuration represents the current settings associated with the virtual actuator. Each virtual actuator has associated a SoftwareDefinedCapability. The capability contains the needed information to manage the environment through the virtual unit. The attributes characterizing it are the enforcementProtocol, the endpoint, and the mechanism.

The Virtual Sensor is a software component through which data concerning the environment is collected. The Virtual Sensor has associated a description, name, id and configuration. The configuration represents the current settings associated with the virtual sensor. Each virtual sensor has associated a number of SoftwareDefinedMetrics that it is measuring. The SoftwareDefinedMetric consists of the id, name, description, endpoint, period, measuredProperty, and measurementProtocol. These are the attributes necessary for accessing the sensor information.

Test configurations for cyber-physical systems

For testing cyber-physical systems, we associate to Metric and to SoftwareDefinedMetric a TestConfiguration stereotype, being an extension point for further types of tests. The TestConfiguration has a name, description and a testTimeout. The testTimeout gives the maximum amount of time in which the associated TestExecutor should answer the test. The TestExecutor has an association to Capability or to SoftwareDefinedCapability, for the case in which certain settings need to be done on the CPS before the test is executed. The TestExecutor has String descriptor of the executor, a Boolean saying whether the unit executing this test is distinct from the target of the test, and a target attribute describing the target of the test.

Each TestConfiguration has associated a TestTrigger, which describes when the test should be executed. The TestTrigger is of two types, either EventTrigger or PeriodicTrigger. The EventTrigger has two attributes: the description of the event, and the eventSource. The EventTrigger is used for event-based testing (e.g., when, during system runtime, the quality is too low). The PeriodicTrigger has two attributes: the period and the timeUnit. The PeriodicTrigger is used for tests executed in specific periods (i.e., the period attribute), described under various units of time (i.e., the timeUnit attribute).

4.2.2.2 Infrastructure Internal U-Profile Library

This section presents the internal library that defines the various types described in D1.2 [3] and used in the CPS Profile. An overview of the diagram is shown in Figure 9 followed by a short description of each type.

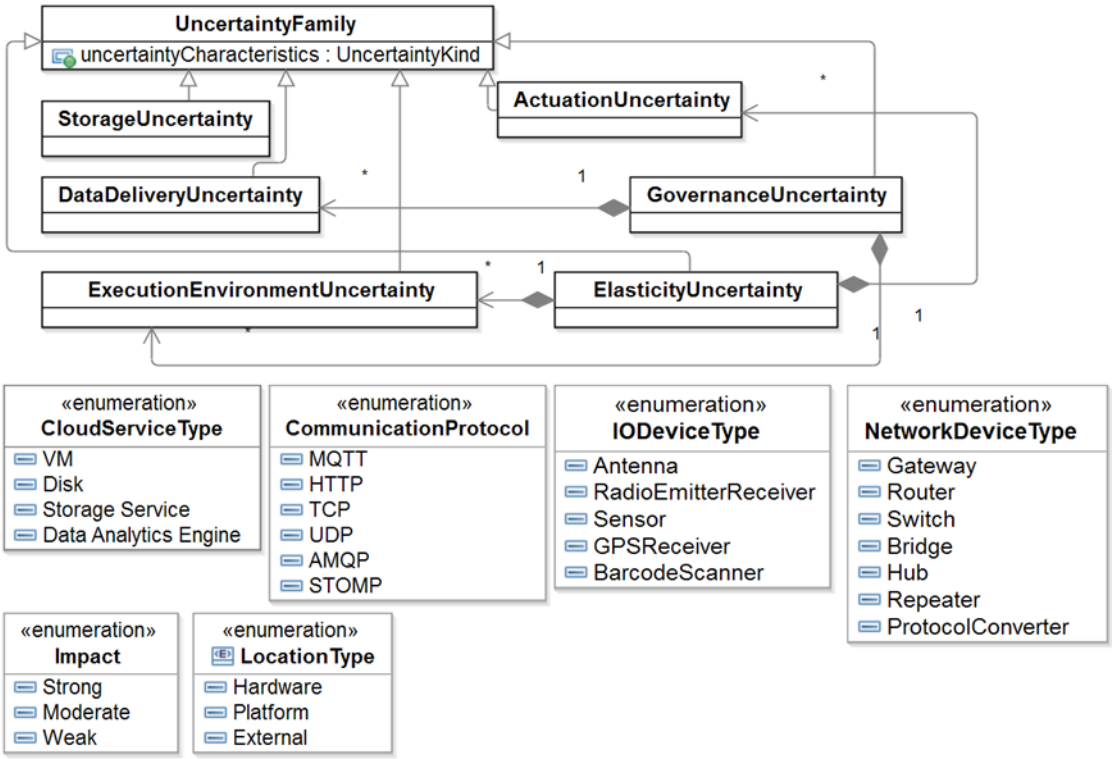

Figure 9 Internal Infrastructure Profile Library

UncertaintyFamily

This element defines the type of uncertainty family according to the families defined in the previous section. The element is extended by 6 uncertainty families: DataDeliveryUncertainty, ActuationUncertainty, ExecutionEnvironmentUncertainty, GovernanceUncertainty, ElasticityUncertainty, StorageUncertainty. A new family StorageUncertainty is added to to capture uncertainties that occur regarding the storage infrastructure.

LocationType

This enumeration defines an initial set of location types where an uncertainty can occur, and has as values: Hardware, Software, and External. A Hardware and Software location indicates that the uncertainty appears in the hardware or software of the CPS, while External indicates an uncertainty appearing external to the CPS, but still affecting the CPS.

CloudServiceType

This enumeration defines an initial set of cloud service types which abstract the plethora of offered cloud services, enabling us to model CPSs using cloud offerings. The enumeration values are VM, Disk, StorageService, and DataAnalyticsEngine.

CommunicationProtocol

This enumeration defines an initial set of communication protocols between different infrastructure elements. The supported protocol values are MQTT, HTTP, TCP, UDP, AMQP, and STOMP.

IODeviceType

This enumeration defines a set of Input/Output (I/O) devices which can be found in cyber-physical systems. The enumeration values are: Gateway, Router, Switch, Hub, and ProtocolConverter.

4.2.2.3 Additional Supporting Library

The aim of the additional supporting library is to ease the management of uncertainties. The libraries defines the concept "ManagedUncertainty" which extends the core uncertainty concept. In general, the specification of test-ready model is an iterative processes in which the models will be enhanced. Probably, these enhancements cause changes in the model and in particular the application of the uncertainty profile stereotypes will be updated based on new acquire knowledge. The management concepts are shown in Figure 10.

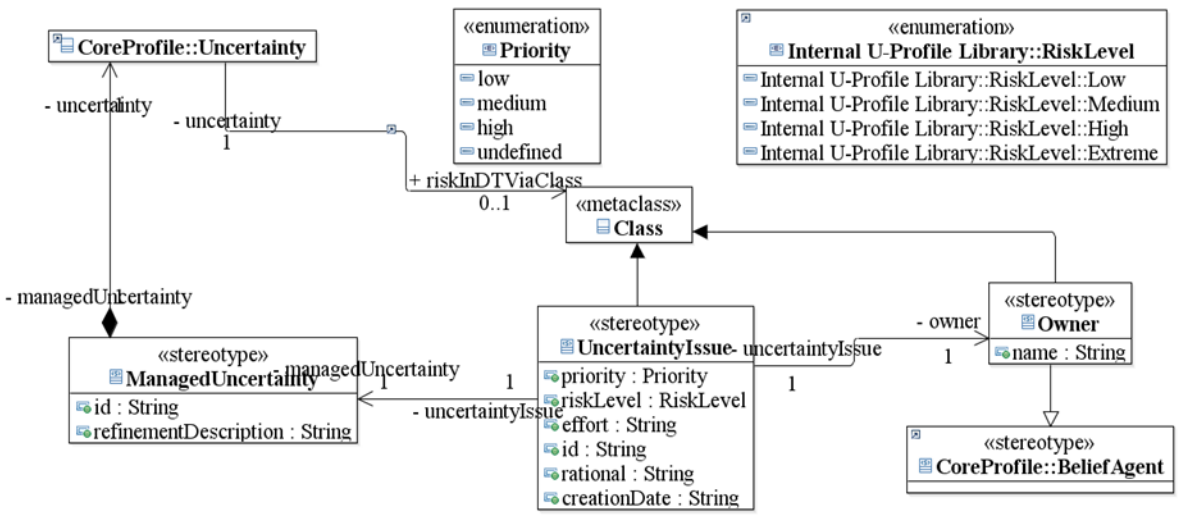

Figure 10 Management of Uncertainties

Basically, test modelers can assign specific uncertainty issues to managed uncertainties. For each issue priority and risk level can be defined. Furthermore, a potential effort to solve the issue can be indicated. The Owner of the issue is based on the BeliefAgent concepts from the core profile.

Uncertainties are not isolate. They can have relationships with each other. Thus this library defines also specific kind of uncertainty relationships to describe dependencies between uncertainties in more detail. The main relationship concepts are depicted by Figure 11.

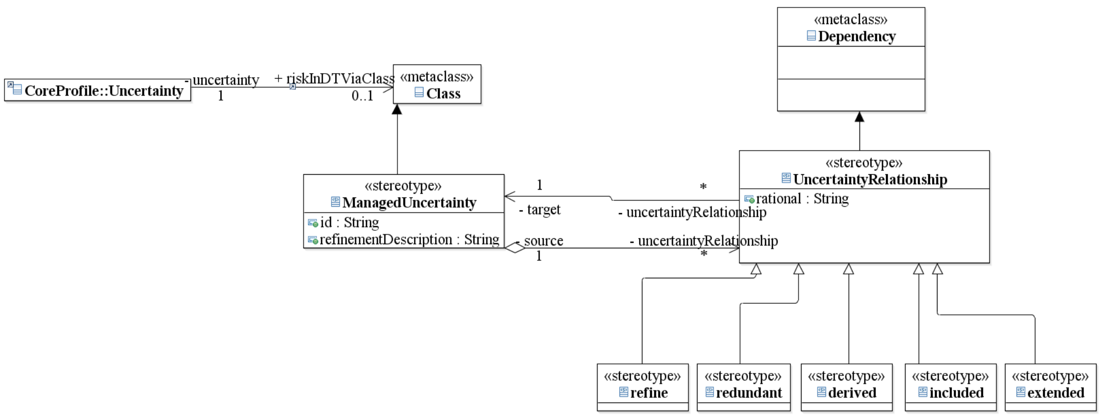

Figure 11 Uncertainty Relationships

The uncertainty relationship class extends the UML dependency class. The source and targets of uncertainty relations are uncertainties. The library specifies five relationship types between uncertainties, namely refine, redundant, derived, included and extended.

4.2.3 Modelling Methodology

In addition to the uncertainty profile definition, the UMF provides also a modelling methodology to guide test modeler through the U-Test workflow. The modelling methodology is detailed described in the appendix document. Figure 12 depicts the overall flow of this UMF guidance.

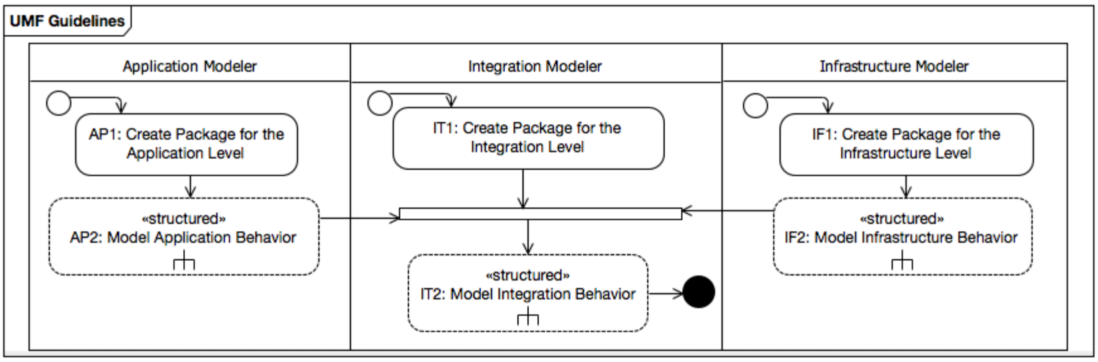

Figure 12 Uncertainty Modelling Guidelines

The modelling methodology is expressed by UML activity diagrams. The methodology distinguishes between the three U-Test CPS abstraction levels, in fact Application, Integration, and Infrastructure. For each abstraction level a detailed list of action points is provided. For instance, the modeler is guided and instructed to create specific state machines to describe the behavior or to reuse existing generic state machines. Finally, it provides information on which level specific UUP profile could be applied.

4.2.4 Test-Ready Models for Pilots

The output of the UMF is test ready models, which are based on UML, UTP and the UUP profile. In general, test modeler should be guided by the provided modeling methodology in order to create these test ready models. The goal is to produce models that are defined at sufficient level of detail to generate adequate test cases. The test-ready models of the both pilots are described in the next section.

5 Pilot Modelling

This chapter describes the defined test-ready models based on the U-Test workflow and the modelling methodology. In this version of the report the models describes specific use cases for both pilots in order to validate the maturity of the uncertainty profile concepts. It is planned to update this section in future versions of this report.

5.1 Design Consideration

This section summarizes the design consideration, which is relevant to the modelling of the pilot use cases. The set of relevant design considerations comprises assumptions and dependencies, general constraints, design methodology as well as goals and guidelines. The latter are described in this document in detail. The other design considerations are illustrated with examples and references to their source are given.

5.1.1 ASSUMPTIONS AND DEPENDENCIES

The following models are defined based on a number of assumptions. These assumptions have been captured in the following list.

- Models are not complete. They depict only subsets of the real existing CPS described within Pilot Use Cases. U-Test scope is not on focusing of the modelling of the CPS rather the definition of uncertainties and generation of test cases. However, an existing model of the CPS is mandatory to apply the developed U-Test methodology.

5.1.2 GENERAL CONSTRAINTS

In addition to the above assumptions and dependencies, there are also a number of more general constraints, which have an impact on the modelling of the pilot use cases. These general constraints are documented as design constraints in the requirements specification.

5.1.3 GOALS AND GUIDELINES

The goal of the pilot use case modelling, presented in this document, is to describe the pilot CPSs in a way that enables all stakeholders to discuss requirements and functionality of U-Test methodology and to enable U-Test partner to apply the developed uncertainty modelling approach as well as test case generation and execution based on uncertainty models.

For achieving this goal each pilot use case model is following the IEEE guidelines as outlined in [1]. This means in particular that the system high-level design is organized according to the IEEE design viewpoints.

5.1.4 DESIGN METHODOLOGY

For the development of the models the UML has been used. Based on the model-driven development approach, Rational Software Architect Version 9.1 (RSA) and Eclipse Papyrus MDT have been used.

5.2 FPX

The following section describes the test-ready model of the Geo Sports demonstrator of FPX (see U-Test deliverable D1.1.) [2]. The model description is separated into the three levels application, infrastructure and integration.

5.2.1 Application Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

5.2.2 Infrastructure Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

5.2.3 Integration Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

5.3 ULMA

The following section describes the test-ready model of the handling system demonstrator of ULMA (see U-Test deliverable D1.1) [2]. The model description is separated into the three levels application, infrastructure and integration.

5.3.1 Application Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

5.3.2 Infrastructure Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

5.3.3 Integration Level

<<The details have been removed from this document due to the agreement for confidentiality with the pilot companies>>

6 Summary and Conclusion

6.1 Application Level

Achievement of M2 – FF

FF specified and implemented the UUP for application level modelling for RSA and Papyrus. Furthermore, FF designed the UTEST application level modelling methodology which is part of modelling guidelines of D2.1

FF has achieved the agreed percentage of test-ready models of the pilots. In absolute numbers, one use case of each pilot has been modelled. The use case UC1_APP_9 has been developed from the FPX pilot, and the use case UC2_APP_1.2 has been developed from the ULMA case study.

Plan for achieving M3 – FF

As agreed in the milestone planning document, FF will develop test-ready models for at least 50% of the use cases of the ULMA pilot and at least 66% of the uses cases of the FPX pilot.

6.2 Infrastructure Level

Achievement of M2 – TUW

The second milestone was achieved with having the profile stable for both the infrastructure-level uncertainty, and for the cyber-physical systems profile. For both the Geo-sports case (FPX) and the Handling Systems Demonstrator (ULMA) we have modelled infrastructure-related information, and we have modelled one use-case for each, UC1_INFR_1.1 and UC2_INFR_1.1, which represents the 25% of the use-cases promised for the 2nd Milestone.

Plan for achieving M3 – TUW

Based on our experience and results with milestone 2, we will increase the number of use-cases covered, and will deliver 60% of the infrastructure-related test-ready models.

6.3 Integration Level

Achievement of M2 – SRL

As agreed in the revised plan of WP2, we have devised an initial version of the UMF at the integration level and successfully modeled at least 33% of use cases for each case study. We modeled 2 out of 4 use cases for ULMA and 1 out of 3 use cases for FPX. This demonstrates the achievement of M2 for WP2 for the integration level.

Plan for achieving M3 – SRL

In terms of the achievement of M3, we already have a plan to further refine UMF at the integration level. In addition, we have already started to model additional 33% of use cases for both case studies to achieve the target of 66% by M3 as planned for M3.

7 Bibliography

[1] 1471-2000 - IEEE Recommended Practice for Architectural Description for Software-Intensive Systems, 2000.

[2] U-Test Deliverable Report on Requirements Collection D 1. 1

[3] U-Test Deliverable Report on Taxonomy D 1. 2

[4] U-Test Deliverable Report on Uncertainty Modelling Framework V1 part A (Uncertainty Profile description

[5] U-Test Deliverable Report on Uncertainty Modelling Framework V1 part B (model files for RSA/Papyrus)

[6] M. Zhang, S. Ali, T. Yue, and P. H. Nguyen, "Uncertainty Modeling Framework for the Integration Level V.1," Simula Research Laboratory, Technical Report 2016-01, 2016, Available: https://www.simula.no/publications/uncertainty-modeling-framework-integration-level-v1.