Simula Research Laboratory, Technical Report 2015-3 Feb, 2016

Understanding Uncertainty in Cyber-Physical Systems: A Conceptual Model

Man Zhang[1], Bran Selic[1], Shaukat Ali[1], Tao Yue[1, 2], Oscar Okariz[3], Roland Norgren[4]

[1]Simula Research Laboratory, [2]University of Oslo, Oslo, Norway
[3]ULMA Handling Systems, Oñati, Spain; [4]Future Position X, Gävle, Sweden

{manzhang, bselic, shaukat, tao}@simula.no
ookariz@manutencion.ulma.es, roland.norgren@fpx.se

Abstract. Uncertainty is intrinsic in most technical systems, including Cyber-Physical Systems (CPS). Therefore, handling uncertainty in a graceful manner during the real operation of CPS is critical. Since designing, developing, and testing modern and highly sophisticated CPS is an expanding field, a step towards dealing with uncertainty is to identify, define, and classify uncertainties at various levels of CPS. This will help develop a systematic and comprehensive understanding of uncertainty. To that end, we propose a conceptual model for uncertainty specifically designed for CPS. Since the study of uncertainty in CPS development and testing is still irrelatively unexplored, this conceptual model was derived in a large part by reviewing existing work on uncertainty in other fields, including philosophy, physics, statistics, and healthcare. The conceptual model is mapped to the three logical levels of CPS: Application, Infrastructure, and Integration. It is captured using UML class diagrams, including relevant OCL constraints. To validate the conceptual model, we identified, classified, and specified uncertainties in two distinct industrial case studies.

Keywords. Uncertainty; Cyber-Physical Systems; Conceptual Model.

1 Introduction

Cyber-Physical Systems (CPS) are present in a variety of safety/mission critical domains [2-4]. Given the pervasiveness of CPS and their criticality to the daily functioning of society, it is vital for such systems to operate in a reliable manner. However, since they generally function in an inherently complex and unpredictable physical environment, a major difficulty with these systems is that they must be designed and operated in the presence of uncertainty. By *uncertainty* we mean here the lack of certainty (i.e., knowledge) about the timing and nature of inputs, the state of a system, a future outcome, as well as other relevant factors.

As a first crucial step in such an investigation, we feel that it is necessary to understand the phenomenon of uncertainty and all its relevant manifestations. This means to systematically identify, classify and specify uncertainties that might arise at any of the three levels of CPS: *Application*, *Infrastructure*, and *Integration*. Based on studying and analyzing existing uncertainty models developed in other fields, including philosophy, physics, statistics and healthcare [5-8], we have defined an uncertainty conceptual model for CPS (*U-Model*) with the following objectives: 1) provide a unified and comprehensive description of uncertainties to both researchers and practitioners, 2) classify uncertainties with the aim of identifying common representational patterns when modeling uncertain behaviors, 3) provide a reference model for systematically collecting uncertainty requirements, 4) serve as a methodological baseline for modeling uncertain behaviors in CPS, and, last but not least, 5) provide a basis for standardization of the conceptual model leading to its broader application in practice.

To verify the completeness and validity of the *U-Model*, we validated it using uncertainty requirements[1] collected from two industrial case studies from two different domains: 1) Automated Warehouses developed by ULMA Handling Systems (www.ulmahandling.com/en/), Spain, 2) GeoSports (fpx.se/geo-sports/) developed by Future Position X, Sweden. This empirical validation was systematically performed in several stages and, as a result, several revisions of the *U-Model* were obtained in addition to a refined set of uncertainty requirements. The version of the *U-Model* that emerged from this work is presented in this paper. Based on the results of this validation, we discovered 61.5% (averaged across the two case studies) additional uncertainties not identified in the initial specifications. The rest of this paper is organized as follows: Section 2 presents the background and a running example. Section 3 presents the *U-Model*. Section 4 presents evaluation and discussion. Section 5 discusses related work and we conclude the paper in Section 6.

2 Background and Running Example

This section first presents key definitions that are required to understand the rest of the paper in Section 2.1, followed by a running example that will be used to explain *U-Model* concepts (Section 2.2) throughout the paper.

2.1 Definitions

A CPS is defined in [1] as: *"A set of heterogeneous physical units (e.g., sensors, control modules) communicating via heterogeneous networks (using networking equip-*

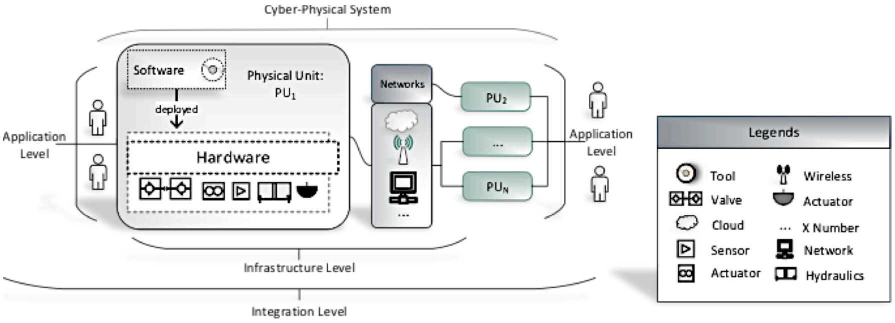


Fig. 1. Conceptual model of a Cyber-Physical System [1]

[1] Use cases containing scenarios having uncertainty.

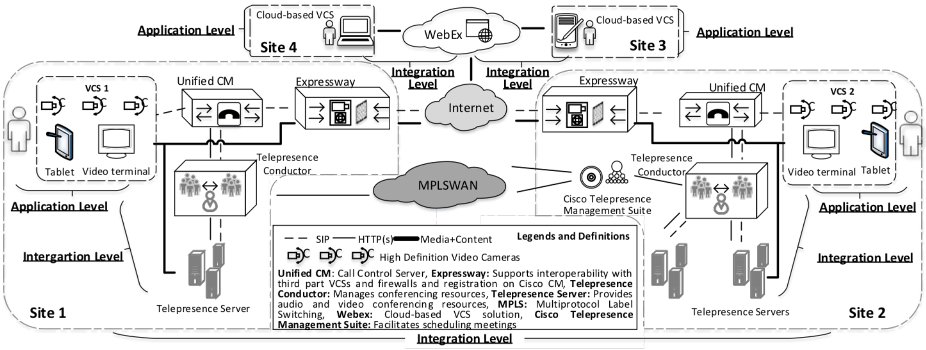
ment) and potentially interacting with applications deployed on cloud infrastructures and/or humans to achieve a common goal" and is conceptually shown in Fig. 1. As defined in [1], uncertainty can occur at the following three levels (Fig. 1): 1) *Application level:* Due to events/data originating from the application of the CPS; 2) *Infrastructure level*: Due to interactions including events/data among physical units, networking infrastructure, and/or cloud infrastructure, 3) *Integration level*: Due to either interaction among uncertainties at the first two levels or due to interactions between application and infrastructure levels.

2.2 Running Example

Due to confidentiality constraints, the actual industrial CPS case studies that we used to evaluate the *U-Model* (as reported in Section 4.2) cannot be described in detail. Instead, we chose a Videoconferencing Systems (VCS) developed by Cisco, Norway, as an example to illustrate the conceptual model that has been used in our previous projects.

A typical VCS sends and receives audio/video streams to other VCS in a videoconference including dedicated hardware-based VCS, software-based VCS for PCs, and cloud-based VCS solutions (e.g., WebEx) as shown in Fig. 2 (inspired from [9] and our existing collaboration with Cisco). To support videoconferences a complex infrastructure is provided by Cisco (Fig. 2) comprising of a variety of hardware such as gateways (e.g., Expressway) and dedicated servers (e.g., Telepresence and unified Call Management servers). In Fig. 2, we also show the various levels at which the uncertainties can occur in the context of our running example. For example, as shown in Fig. 2, at Site 2, the interactions of *Application* level uncertainties in VCS 2 and uncertainties in the Telepresence Servers are shown as *Integration* level uncertainties.

To facilitate the understanding of concepts, a VCS represents aspects of the physical world in a somewhat simplified form. Among other functions, the VCS controls the movement of a set of cameras that are directly attached to it via wired/wireless media. This can also be performed via a cloud-based VCS application (i.e., WebEx) in addition to dedicated hardware-based solutions. In the course of a videoconference, a number of different uncertainties exist due to the complex and heterogeneous collection of networks, cloud-based infrastructures, and VCS.



Unified CM: Call Control Server, **Expressway:** Supports interoperability with third part VCSs and firewalls and registration on Cisco CM, **Telepresence Conductor:** Manages conferencing resources, **Telepresence Server:** Provides audio and video conferencing resources, **MPLS:** Multiprotocol Label Switching, **Webex:** Cloud-based VCS solution, **Cisco Telepresence Management Suite:** Facilitates scheduling meetings

Fig. 2. Running Example – Videoconferencing System (VCS)

3 Uncertainty Conceptual Model

The *U-Model* as shown in Fig. 3 includes *Belief Model*, *Uncertainty Model* and *Measure Model*.

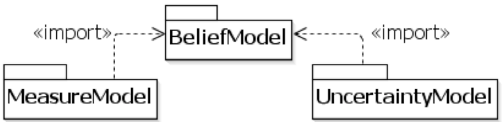


Fig. 3. The top-level model of *U-Model*

3.1 Belief Model

The *U-Model* takes a subjective approach to representing *uncertainty*. This means that uncertainty is modeled as a *state* (i.e., worldview) of some agent or agency – henceforth referred to as a BeliefAgent – that, for whatever reason, is incapable of possessing complete and fully accurate knowledge about some subject of interest. Since it lacks perfect knowledge, a BeliefAgent possesses a set of subjective Beliefs about the subject. These may be *valid*, if the beliefs accurately represent facts, or *invalid*, if they do not[2]. A Belief is an abstract concept, but can be expressed in concrete form via one or more explicit BeliefStatements. Different BeliefAgents may hold different views about a given subject, which is why each BeliefStatements is associated with a particular BeliefAgent. Note that a BeliefAgent does not necessarily represent a human individual; it could constitute a community of individuals, some non-human organism, or even some technological system, such as a computer system[3].

These and other core concepts of the *U-Model* are represented as a class diagram in Fig. 4, where *subjective* concepts are represented by the grey-filled boxes and *objective* concepts as the unfilled boxes in Fig. 4. Subjective concepts are manifestations of the imperfect knowledge of a BeliefAgent. Conversely, objective concepts reflect objective reality and are, therefore, independent of BeliefAgents and their imperfections. One significant characteristic of the subjective concepts is that they can vary over time, as might occur, e.g., when more information becomes available[4].

Uncertainty (lack of confidence) represents *a state of affairs whereby a* BeliefAgent *does not have full confidence in a* Belief *that it holds*. This may be due to various factors: lack of information, inherent variability in the subject matter, igno-

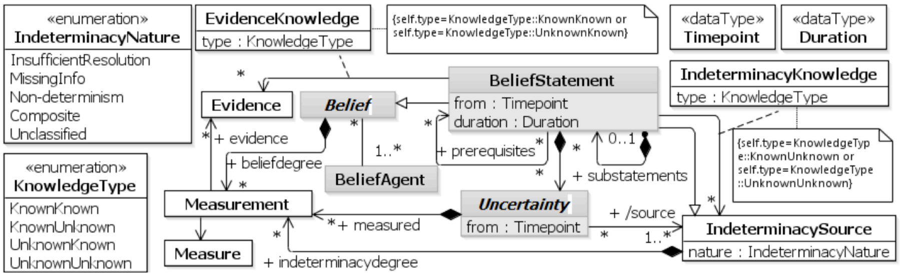


Fig. 4. The Core Belief Model

[2] Such a strictly binary categorization may not be always realistic, since *Belief*s could be characterized by degrees of validity. However, in this model, we choose to ignore such subtleties. Specifically, a BeliefStatement is deemed to be valid if it is a sufficient approximation of the truth for the purpose on hand.

[3] In this case, the Beliefs would be reflected in the rules that are programmed into the system.

[4] However, more information does not necessarily imply a decrease in uncertainty.

rance, or even due to physical phenomena, e.g., the Heisenberg uncertainty principle. While Uncertainty is an abstract concept, it can be represented by a corresponding Measurement expressing in some concrete form the *subjective degree* of uncertainty held by the agent to a BeliefStatement. Since the latter is a subjective notion, a Measurement should not be confused with the degree of validity of a BeliefStatement. Instead, it indicates the level of confidence that the agent has in a statement[5].

Finally, note that this model is intentionally made very general, which allows it to be extended and customized for a variety of purposes, e.g., uncertainty model-based testing of CPS in the context of our project (www.u-test.eu). Fig. 4 does not show the complete model, e.g., to reduce visual clutter, some of the OCL constraints have been removed. In the remainder of this section, we examine key concepts of the core model in more detail and illustrate some of them using the running VCS example.

Belief. A Belief is an *implicit* subjective explanation or description of some phenomena or notions[6] held by a BeliefAgent. This is an abstract concept whose only concrete manifestation is as a BeliefStatement.

In the context of our running example, a test engineer at Cisco may have his/her own Beliefs about how a VCS works. When coding test cases, he/she concretizes his/her Beliefs as executable test scripts that may or may not correspond to the actual implementation the VCS. A BeliefStatement in this context could be manifested as one executable test case file and in other contexts it may correspond to other artifacts, e.g., source code.

BeliefAgent. A BeliefAgent is a physical entity[7] owning one or more Beliefs about phenomena/notion. A BeliefAgent can take actions based on its Beliefs. In our example of CPS testing, BeliefAgents include: 1) Application level: software test engineers focusing on testing new versions of the VCS software, and 2) Infrastructure level: Network engineers focusing on testing a VCS under diverse network situations.

BeliefStatement. A BeliefStatement is a concrete and explicit specification of some Belief held by a BeliefAgent about possible phenomena or notions belonging to a given subject area. A BeliefStatement can be an aggregate of two or more component BeliefStatements, or it may require one or more prerequisite BeliefStatements.

The concrete form of a BeliefStatement can vary, and may represent informal pronouncements made by individuals or groups, documented textual specifications expressed in either natural or formal languages, formal or informal diagrams, etc.

Due to the complex nature of objective reality and our human and technical limitations, it may not always be possible to determine whether or not a BeliefStatement is valid. Furthermore, the validity of a statement may only be meaningfully defined

[5] E.g, many people in the past were absolutely certain that the Earth was flat.

[6] "Phenomena" here is intended to cover aspects of objective reality, whereas "notion" covers abstract concepts, such those encountered in mathematics or philosophy.

[7] We exclude here from this definition "virtual" BeliefAgents, such as those that might occur in virtual reality systems and computer games.

within a given context or purpose at a given point of time. Thus, the statement that "the Earth can be represented as a perfect sphere" may be perfectly valid for some purposes but invalid or only partly valid for others. For our needs, we are more interested in analyzing uncertainties in a BeliefStatement rather than studying its validity.

In our example, we define the following BeliefStatements: 1) *Application* level: The VCS will successfully connect to another VCS 70% of the time (see Table 1); 2) *Infrastructure* level: The Expressway gateway is successful 99% of the time in connecting a Cisco VCS with a third party VCS; and 3) *Integration* level: A VCS communicates with the Expressway gateway with a 90%-95% success rate.

Evidence. Evidence is either an observation or a record of a real-world event occurrence or, alternatively, the conclusion of some formalized chain of logical inference that provides information that can contribute to determining the validity (i.e., truthfulness) of a BeliefStatement. Evidence is inherently an objective phenomenon, representing *something that actually happened*. This means that we exclude here the possibility of counterfeit or invented evidence. Nevertheless, although Evidence represents objective reality, it needs not be conclusive in the sense that it removes all doubt (Uncertainty) about a BeliefStatement. In our example of an Application level BeliefStatement, i.e., "The VCS successfully dials to another VCS 70% of the time". The Evidence of the 70% of success rate of dial may be obtained from the execution of 100 test cases on the VCS in the past week (see Evidence Table 1).

EvidenceKnowledge. EvidenceKnowledge expresses an objective relationship between a BeliefStatement and relevant Evidence. It identifies whether the corresponding BeliefAgent is aware of the appropriate Evidence. Thus, an agent may be either aware that it knows something (KnownKnown), or it may be completely unaware of

Table 1. Running Example – *Dial* of VCS

Package	Concept	Explanation
Belief Model	Level	*Application*
	BeliefAgent	Software testing engineers
	BeliefStatement	The VCS successfully dials to another VCS 70% of the time.
	Indeterminacy Source	Improper human behavior where he/she enters an incomplete name/number of VCS to dial `IndeterminacyNature:: Non-determinism`, and `IndeterminacyKnowledge.type= KnowledgeType::KnownUnknown`
	Evidence	Execution of 100 test cases on the VCS in the past week involving the dial command `EvidenceKnowledge.type =KnowledgeType::KnownKnown`
	Uncertainty	Uncertainty in whether the dial to another VCS will be successful or not. This concept may depend on another uncertainty composed by another BeliefStatement specified by the network engineer, e.g. "The Expressway gateway is 99% of the time successful in connecting Cisco's VCS with third party VCS."
Uncertainty Model	Type	Occurrence
	Lifetime	Difference of time that the dial was initiated and response from the system was received
	Locality	Invocation of the dial API of VCS
	Pattern	Derived pattern from the collection of values of lifetime of the uncertainty
	Risk	Low or even can be ignored
Measure Model	Measurement	70% of the time, derived from Evidence based on test execution history
	Measure	Probability

Evidence (UnknownKnown). This is formally expressed by the two constraints attached to EvidenceKnowledge (see below). An example is provided in Table 1.

```
Context EvidenceKnowledge
Inv: self.type = KnowledgeType::KnownKnown or self.type = Knowledg-
  eType::UnknownKnown
```

Indeterminacy. Indeterminacy is a situation whereby the full knowledge necessary to determine the required factual state of some phenomena/notions is unavailable[8]. This is an abstract concept whose only concrete manifestation is in the form of an IndeterminacySource. As noted earlier, this may be due either to subjective reasons (e.g., agent ignorance) or to objective reasons (e.g., the Heisenberg uncertainty).

IndeterminacySource. It is also useful to explicitly identify factors that lead to Uncertainty referred to as IndeterminacySources. This represents a situation whereby the information required ascertaining the validity of a BeliefStatement is indeterminate in some way, resulting in Uncertainty being associated with that statement. One possible source of indeterminacy can be another BeliefStatement, which is why the latter is a specialization of IndeterminacySource (see Fig. 5).

BeliefStatement and Uncertainty when associated with IndeterminacySource represent sources of lack of confidence. And the source of Uncertainty is subset of the source of BeliefStatement, which own this Uncertainty (see below).

```
Context BeliefStatement
Inv: self.uncertainty->forAll(u:Uncertainty|self.indeterminacysource-> in-
  cludesAll(u.source))
```

For example, for the following BeliefStatement: “The VCS successfully dials to another VCS 70% of the time”, for which there might be several IndeterminacySources. A possibility is incorrect operator behavior, where an incomplete name of the target VCS specified (IndeterminacySource entry in Table 1).

IndeterminacyNature. IndeterminacyNature represents the specific kind of indeterminacy and can be one of the following: 1) InsufficientResolution – The information available about the phenomenon in question is not sufficiently precise; 2) MissingInfo – The full set of information about the phenomenon in question is unavailable at the time when the statement is made; 3) Non-determinism – The phenomenon in question is either practically or inherently non-determinism; 4) Composite – A combination of more than one kinds of indeterminacy; 5) Unclassified – Indeterminate indeterminacy.

IndeterminacyKnowledge. IndeterminacyKnowledge expresses an objective relationship between an IndeterminacySource and the awareness that the BeliefAgent has of that source. So, even though it is agent specific, it is still an objective concept since it does not represent something that is declared by the agent. For instance, an

[8] Care should be taken to distinguish between indeterminacy and non-determinism. The latter is only one possible source of indeterminacy.

agent may be aware that it does not know something about a possible source (KnownUnknown), or the agent may be completely unaware of a possible source of indeterminacy (UnknownUnknown).

This is formally expressed by constraints attached to IndeterminacyKnowledge (see below).

```
Context IndeterminacyKnowledge
Inv: self.type = KnowledgeType::KnownUnknown or self.type = Knowledg-
  eType::UnknownUnknown
```

KnowledgeType. KnowledgeType (represented as enumeration) has four values:

1) KnownKnown – Indicates that an associated BeliefAgent is consciously aware of some relevant aspect.

2) KnownUnknown (*Conscious Ignorance*) – Indicates that an associated BeliefAgent understands that it is ignorant of some aspect.

3) UnknownKnown (*Tacit Knowledge*) – Indicates that an associated BeliefAgent is not explicitly aware of some relevant aspect that it, nevertheless, may be able to exploit in some way

4) UnknownUnknown (*Meta Ignorance*) – Indicates that an associated BeliefAgent is unaware of some relevant aspect.

At a given point in time, a BeliefAgent always makes a statement based on a KnownKnown Evidence and a KnownUnknown IndeterminacySource. Splitting EvidenceKnowledge and IndeterminacyKnowledge provides the flexibility to enable transitions among different knowledge types (e.g., from UnknownKnown to KnownKnown), based on the evolution of EvidenceKnowledge and IndeterminacyKnowledge related to the associated BeliefAgent.

For the following BeliefStatement: "The VCS successfully dials to another VCS 70% of the time" and an IndeterminacySource is improper operator behavior, the KnowledgeType of IndeterminacyKnowledge is KnownUnknown.

Measurement. Measurement when associated with a given IndeterminacySource represents the optional quantification (or qualification) that specifies the degree of indeterminacy of the IndeterminacySource. For example, in the case of a non-determinism IndeterminacySource, its measurement could be expressed by a probability or a probability density function. For the example presented in Table 1, '70%' is the measurement of the IndeterminacySource improper operator behavior.

Measurement when associated with Uncertainty is a subjective concept representing the actual measured value of an uncertainty defined by a BeliefAgent. It may be possible to specify a Measurement that quantifies in some way (e.g., as a probability) the degree of the uncertainty that a BeliefAgent associates with a BeliefStatement.

Measurement when associated with Belief represents sets of measured values of all the uncertainties contained by a BeliefStatement defined by a BeliefAgent. Several constraints on Measurement ensure that each Measurement owned by either Belief, Uncertainty or IndeterminacySource has a unique Measure. Currently, we modeled three different measures, i.e., Probability, Ambiguity and Vagueness that are discussed in the *Measure model* (Section 3.3). In the future, we will provide UML model libraries for Measurement when implementing *U-Model* as a UML profile.

Measure. Measure is an objective concept specifying method of measuring uncertainty. More details are presented in Section 3.3.

3.2 Uncertainty Model

This model (Fig. 5) was inspired by concepts defined in the literature on uncertainty [10-14] and is an adjunct to the *Core Belief Model* (Section 3.1). The uncertainty model expands on Uncertainty from several different viewpoints and introduces related abstractions. Notice that Uncertainty has a self-association. This self-association facilitates: 1) relating different *Application* level uncertainties to each other, 2) relating different *Infrastructure* level uncertainties to each other, 3) relating *Application* level and *Infrastructure* level uncertainties to each other, 4) relating *Integration* level uncertainties to each other, and 5) relating *Application*, *Integration*, and *Infrastructure* level uncertainties. This self-association can be specialized into different types of relationships such as ordering and dependencies. Here, we intentionally did not specialize it to keep the model general, so that it can be specialized for various purposes and contexts. In the rest of the section, we discuss each subtype of Uncertainty and its associated concepts.

Uncertainty. As noted, Uncertainty represents a situation whereby a BeliefAgent lacks confidence in a BeliefStatement. Fig. 5 shows a conceptual model for different types of Uncertainty inspired from the concepts reported in [11, 13, 14]. Uncertainty is specialized into the following types:

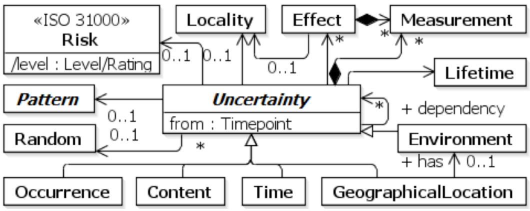


Fig. 5. The Core Uncertainty Model

1) Content – represents a situation, whereby a BeliefAgent lacks confidence in content existing in a BeliefStatement;

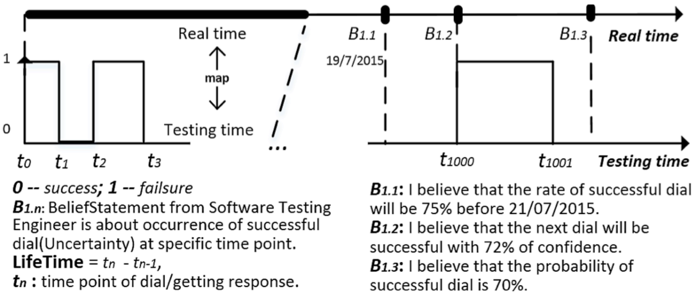
2) Environment – represents a situation whereby a BeliefAgent lacks confidence in the surroundings of a physical system existing in a BeliefStatement;

3) GeographicalLocation –represents a situation whereby a BeliefAgent lacks confidence in geographical location existing in a BeliefStatement;

4) Occurrence –represents a situation whereby a BeliefAgent lacks confidence in the occurrence of events existing in a BeliefStatement;

5) Time –represents a situation whereby a BeliefAgent lacks confidence in time existing in a BeliefStatement.

For example, for the BeliefStatement: “The VCS successfully calls another VCS 70% of the time”, the Uncertainty is whether the dialing to another VCS will be successful or not and classified as Occurrence uncertainty. In case of the BeliefStatement: “The Expressway gateway is successful 99% of the time in connecting a Cisco VCS with a third party VCS”, the Uncertainty is in the connection of the gateway with the third party VCS, and type of uncertainty is again Occurrence (see type of Uncertainty in Table 1).



0 -- success; 1 -- failsure
***B**1.n*: BeliefStatement from Software Testing Engineer is about occurrence of successful dial(Uncertainty) at specific time point.
LifeTime = t_n - t_{n-1},
t_n : time point of dial/getting response.

$B_{1.1}$: I believe that the rate of successful dial will be 75% before 21/07/2015.
$B_{1.2}$: I believe that the next dial will be successful with 72% of confidence.
$B_{1.3}$: I believe that the probability of successful dial is 70%.

Fig. 6. Example of Lifetime and Pattern of Uncertainty

Lifetime. Lifetime represents an interval of time, during which an Uncertainty exists. That is, an Uncertainty may appear temporarily and then disappear. On the other hand, an Uncertainty could be persistent, i.e., it remains until appropriate actions are taken to resolve it.

An example of Lifetime is shown in Table 1. We show two types of time in the diagram: 1) *Real Time* showing the actual passing of the time, 2) *Testing Time*, i.e., a time point in real time, where a testing activity was performed, e.g., a call attempt to establish a videoconference (stimulus to the system under test) or a response from the system was received about success or failure of the call (test result). Time points t_n are shown on *Testing Time* in Fig. 6. A BeliefStatement can be made at any point in the real time, for example, three versions of BeliefStatement B_1 ($B_{1.1}$, $B_{1.2}$, and $B_{1.3}$) can be made at different points of time as shown in Fig. 6. Lifetime of Uncertainty (the occurrence of successful dial) in BeliefStatement B_1 should be $t_n - t_{n-1}$: difference of time that the dial was initiated and response from the system was received for $B_{1.3}$.

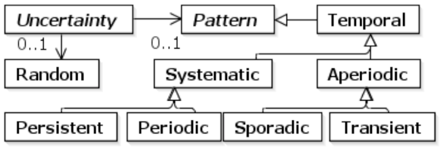


Fig. 7. The Patterns of Uncertainty

Pattern. Fig. 7 shows a conceptual model for the *occurrence Pattern of Uncertainty* inspired from concepts reported in [13, 15, 16]. Notice that in this section, patterns presented are by no means the representation of a complete set of patterns that may exist for an Uncertainty. Rather, we only present the most common patterns.

Periodic uncertainty occurs at regular intervals of time, whereas Persistent Uncertainty is the one that lasts forever. The definition of “forever” varies; e.g, an uncertainty may exist permanently until appropriate actions are taken. On the other hand, an uncertainty may not be resolvable and remains forever. Both Periodic and Persistent inherit from Systematic, which means that these types of patterns occur in some methodical manners, i.e., a pattern that can be described in a mathematical way.

An uncertainty with an Aperiodic pattern occurs at irregular intervals of time, which is further specialized into Sporadic and Transient. A Sporadic uncertainty occurs occasionally, whereas a Transient uncertainty occurs temporarily.

Systematic and Aperiodic uncertainty patterns inherit from Temporal, which means that they both inherently have the notion of time. If an uncertainty occurs without a definite method, purpose or conscious decision, the type of the pattern it follows is referred to as Random.

Uncertainty when associated with Systematic represents pattern and can be measured by Probability (see below).

```
Context Uncertainty
Inv:     self.pattern.oclIsKindOf(Systematic)     implies     self.measured-
   >select(m:Measurement |m.meaasure.oclIsKindOf(Probability))->size()>0
```

For example, when looking at Fig. 6, a pattern of the Uncertainty (Occurrence of a successful call attempt) can be derived after collecting values of Lifetime of the Uncertainty (see Pattern in Table 1).

Locality. Locality is a particular place or a position where an Uncertainty occurs in a BeliefStatement. For example, for the BeliefStatement: "The VCS successfully dials to another VCS 70% of the time", the Locality of the Uncertainty (whether the call attempt to another VCS will be successful or not) is in the invocation (position) of dial API of VCS (see Locality in Table 1).

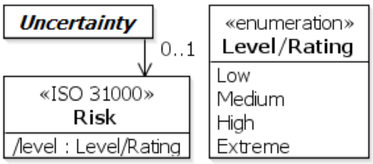


Fig. 8. The Risk of Uncertainty

Risk. An uncertainty may have an associated Risk and high-risk uncertainties deserve special attention. As shown in Fig. 8, an Uncertainty might or might not associated to Risk, whose level can be classified into four levels according to the ISO 31000 – Risk Management standard [17]. Level/Rating is derived from Measurement owned by Uncertainty (e.g., Probability of the Occurrence of an Uncertainty) and Measurement owned by Effect (e.g., high impact using the risk matrix in [18] or any other matrix). For example, for the BeliefStatement: "The VCS successfully calls another VCS 70% of the time", the Risk associated with the Uncertainty in this BeliefStatement is low or the risk could be even ignored (see Risk in Table 1).

3.3 Measure Model

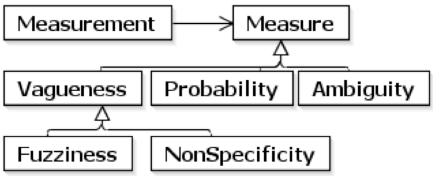


Fig. 9. Measure Model

Fig. 9 shows the *Measure Model* of the *U-Model*, inspired from concepts reported in [11-13] and by no means complete. Depending on the type of Uncertainty, a variety of measures could be applied and new ones can also be proposed when needed. We aim to give a high-level introduction to commonly known measures.

An uncertainty may be described ambiguously (Ambiguity). For example, in statement “The camera is down”, the ambiguity is in the measurement, i.e., the camera is either facing down or disconnected. Interested readers may consult [19] for various measures of Ambiguity. Another common way of measuring Uncertainty is in a vague manner (i.e., Vagueness), which can be further classified into Fuzziness and Non-Specificity. Regarding Fuzziness, an uncertainty may be measured using fuzzy methods. More details can be referred to the fuzzy logic literature such as [19]. In certain cases, it may not be possible to measure an uncertainty using quantitative measurements and instead qualitative measurements can be used. Such qualitative measurements are classified under Non-Specificity methods. Finally, a common way of measuring uncertainty is via Probability. For example, for the BeliefStatement: “The VCS successfully calls another VCS 70% of the time”, the Uncertainty is measured by Probability (see Measure in Table 1).

4 Evaluation

This section presents the results of the industrial case studies that we conducted to evaluate the *U-Model* and collect uncertainty requirements. First, we present a brief overview of the industrial case studies followed by the data collection and validation procedure, results, discussions and lessons learnt.

4.1 Industrial Case Studies

Two industrial case studies were conducted.

Automated Warehouse (AW) by ULMA Handling Systems ULMA. Handling Systems develops automated handling systems for worldwide warehouses of different natures such as Food and Beverages, Industrial, Textile and Storage. Each handling facility (e.g., cranes, conveyors, sorting systems, picking systems, rolling tables, lifts, and intermediate storage) forms a physical unit and together they are deployed to one handling system application (e.g., Storage). A handling system cloud supervision system (HSCS) generally interacts with diverse types of physical units, network equipment, and cloud services. Application-specific processes in HSCS are executed spanning clouds and CPS requiring different configurations.

Geo Sports (GS) by Future Position X. To improve the performance of an individual or a team, conditions of athletes must be measured accurately, routinely over a sustained period of time, and in the athletes’ real environment (e.g., soccer field). Due to the latest technology in positioning, it has become possible to measure the performance of athletes both outdoors and indoors. The measurement is made continuously and in real time using geo sensors during training and live games. In the context of U-Test, we use Bandy in Geo Sports, a kind of ice hockey predominantly played in

Northern Europe and Russia, as a case study. This project is the first in the world in terms of monitoring sports on ice using sensors.

4.2 Development and Validation of Uncertainty Requirements and U-Model

We collected uncertainty requirements from the two industrial case studies in the following ways. The uncertainty requirements were collected as part of an EU project on testing CPS under uncertainty (www.u-test.eu). An initial set of uncertainty requirements were collected by the industrial partners themselves and were later classified into the three CPS levels: Application, Infrastructure, and Integration. Later on, the researchers of Simula Research Laboratory conducted one workshop per partner to further refine the requirements. For AW, the onsite workshop took around three days, whereas in case of GS, a one-day onsite workshop was organized.

The validation procedure is summarized in Fig. 10 and comprises two parallel validation processes. The first validation process is related to the validation of the *U-Model* and was mainly conducted by the researchers. The second validation process focuses on the validation of uncertainty requirements and was mainly performed by the industrial partners.

The validation was developed incrementally (Activities A1 and A2 in Fig. 10),

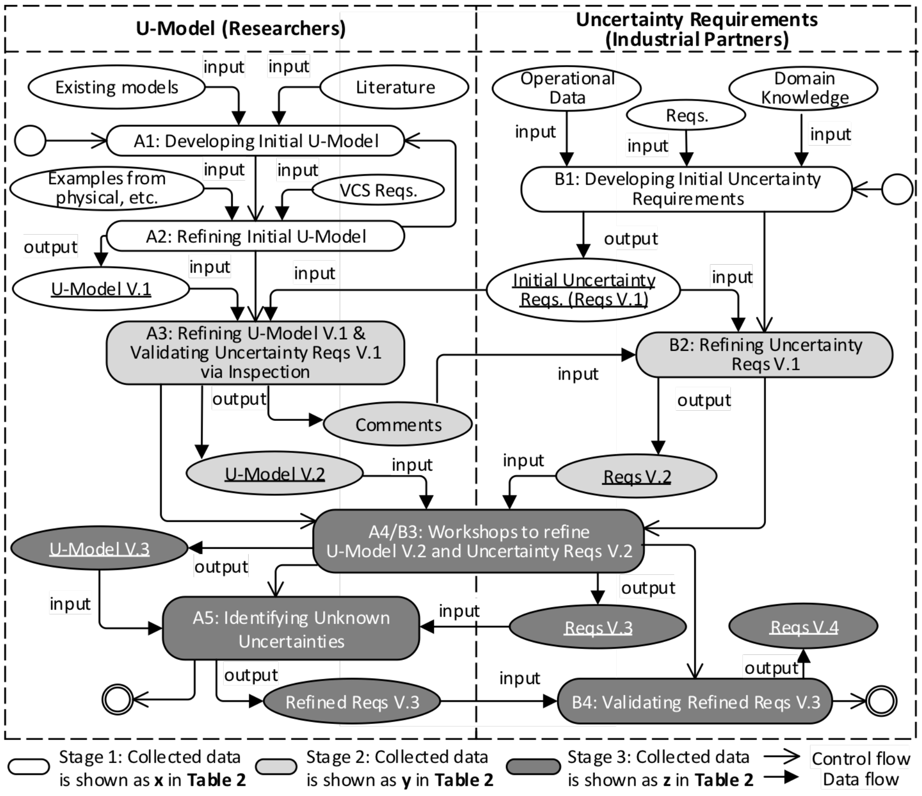

U-Model (Researchers)
Uncertainty Requirements (Industrial Partners)
Existing models
input
input
Literature
A1: Developing Initial U-Model
Examples from physical, etc.
input
input
VCS Reqs.
A2: Refining Initial U-Model
output
U-Model V.1
input
input
A3: Refining U-Model V.1 & Validating Uncertainty Reqs V.1 via Inspection
output
Comments
U-Model V.2
input
A4/B3: Workshops to refine U-Model V.2 and Uncertainty Reqs V.2
U-Model V.3
output
input
A5: Identifying Unknown Uncertainties
input
output
Refined Reqs V.3
Operational Data
Reqs.
Domain Knowledge
input
input
input
B1: Developing Initial Uncertainty Requirements
output
Initial Uncertainty Reqs. (Reqs V.1)
input
B2: Refining Uncertainty Reqs V.1
input
output
input
Reqs V.2
output
Reqs V.3
Reqs V.4
output
input
B4: Validating Refined Reqs V.3
Stage 1: Collected data is shown as x in Table 2
Stage 2: Collected data is shown as y in Table 2
Stage 3: Collected data is shown as z in Table 2
Control flow
Data flow

Fig. 10. Development and Validation of Uncertainty Requirement and *U-Model*

based on existing models in the literature and other related published works (see Section 4.4 for details). The Simula team validated the conceptual model using two types of examples shown as inputs to A2 in Fig. 10: 1) Examples of uncertainties from domains other than CPS, and 2) A subset of VCS requirements. As a result an initial version of the *U-Model* was produced referred as *U-Model* V.1 in Fig. 10.

In parallel, initial uncertainty requirements (Reqs V.1) were provided (Activity B1 in Fig. 10) by the industrial partners based on their domain knowledge, existing requirements of their CPS, and some information from the real operation of the CPS. These initial uncertainty requirements were used as input for A3, focusing on further refining the *U-Model*. In addition, the researchers inspected the collected uncertainty requirements using a requirements inspection checklist provided in [20] and provided a set of comments for the industrial partners on how to improve their requirements. There were two key outputs of the A3 activity: *U-Model* V.2 and comments to refine the requirements. These comments were used by the industrial partners to produce a second version of requirements (Reqs V.2) in B2.

Reqs V.2 and *U-Model* V.2 were then used as inputs for the onsite workshops conducted with the two industrial partners (FPX and ULMA) (A4/B3). These workshops aimed at understanding the CPS by observing them in their real operating environments, giving a presentation of *U-Model* to the industrial partners and collecting their feedback, and discussion on the uncertainty requirements. The outputs of the two workshops represent the current version of the *U-Model* (V.3) presented in this paper and another version of refined requirements (Reqs V.3). Problems that we encountered and eventually addressed during the revisions of *U-Model* are mostly related to incorrect associations between two concepts including e.g., wrong cardinalities and wrong source and target concepts, and redundant concepts.

Finally, the researchers identified a potential set of uncertainties that were not specified or not explicitly specified in the requirements (Reqs V.3) and produced a refined version of Reqs V.3 (activity A5). We followed these steps to systematically perform A5. First, we analyzed each Uncertainty to identify IndeterminacySources that were not explicitly specified in the requirements. Second, based on the identified IndeterminacySources, we analyzed the semantics of each sentence in the requirements to identify unknown uncertainties. For example, if the subject of an English sentence in the requirements is Operator (which has been identified as an IndeterminacySource in the first step), then one Uncertainty might be that the Operator does not perform the task (defined as the predicate of the English sentence) properly. Third, we carefully studied the possibilities of further identifying unknown uncertainties by combining already identified uncertainties that are associated to the same piece of system behavior (specified as e.g., a use case). Whether or not two or more uncertainties can be combined is dependent on the relationships among the uncertainties and their connections to system behavior. In the future, we will propose a methodology with tool support relying on natural language processing to facilitate the identification and specification of uncertainty requirements.

These refined requirements (Refined Reqs V.3) were provided to the industrial partners for validating whether the newly discovered uncertainties were realistic. As a result the final version of requirements (Reqs V.4) was produced.

4.3 Evaluation Results

For each of the industrial case studies, we mapped the three versions of uncertainty requirements (Reqs V.1, Reqs V.2, and Reqs V.4) to the three versions of *U-Model* (V.1 to V.3). The number of the instances of the concepts are shown in columns x (for mapping Reqs V.1 to *U-Model* V.1), y (for mapping Reqs. V.2 to *U-Model* V.2), and z (for mapping Reqs V.4 to *U-Model* V.3) of Table 2, respectively. Notice that Reqs V.3 was the result of the onsite workshops together with *U-Model* V.3 and thus these requirements are not mapped to the model since both the conceptual model and requirements were refined together. We analyzed in total 20 use cases for *AW* and 18 use cases for *GS*. Notice that, the number of use cases for each case study did not change during the requirements collection and the *U-Model* validation process. They were selected at the beginning of the process to capture and specify the key functionalities of the CPS.

Based on the final version of requirements, we can see from Table 2 that most common types of identified uncertainties are Content uncertainties having 91 instances (the last column in Table 2) and Occurrence uncertainties having 205 instances. On the other hand, a relatively lower number of *Time* uncertainties (50), *Environment* uncertainties (32), and GeographicalLocation uncertainties (31) were found in the case studies. Most of the time, uncertainties are due to InsufficientResolution (42 instances), MissingInfo (31 instances) or Non-determinism (89 instances). In terms of Measure, our analysis revealed that 76 of the uncertainties across the case studies may be measured with the Fuzziness measures, 119 with NonSpecificity, whereas 148 with Probability. Notice that in Table 2, we do not show the concepts that have no instances identified from any of the case studies.

In Table 2, the *R1 = y/x -1* column represents the increased percentage of mapping of concepts explicitly captured in Reqs V.2 as compared to Reqs V.1. The *R2* = z/y -1 column shows the increased percentage of mapping of concepts explicitly captured in Reqs V.4, i.e., including unknown uncertainties that weren't explicitly specified in

Table 2. Evaluation Results of Uncertainty Requirements and U-Model

Case Study		**AW**					**GS**					**Freq.**
		x	*y*	*z*	*R1**	*R2**	*x*	*y*	*z*	*R1*	*R2*	*Total*$^+$
Uncertainty	*Content*	14	36	55	1.57	0.53	16	20	36	0.25	0.80	91
	Time	6	16	28	1.67	0.75	5	11	22	1.20	1.00	50
	Occurrence	27	81	126	2.00	0.56	6	50	79	7.33	0.58	205
	Environment	13	15	22	0.15	0.47	4	6	10	0.50	0.67	32
	Geographical Location	4	11	14	1.75	0.27	3	11	17	2.67	0.55	31
Sum for x, y, z / Average for R1, R2		**64**	**159**	**245**	**1.43**	**0.51**	**34**	**98**	**164**	**2.39**	**0.72**	**409**
Indeterminacy	*Insufficient Resolution*	7	18	24	1.57	0.33	11	14	18	0.27	0.29	42
	Non-determinism	7	45	52	5.43	0.16	11	20	37	0.82	0.85	89
	MissingInfo	2	19	24	8.50	0.26	0	5	7	N/A	0.40	31
Sum for x, y, z / Average for R1, R2		**16**	**82**	**100**	**2.67**	**0.43**	**22**	**39**	**62**	**0.55**	**0.57**	**162**
Measure	*Fuzziness*	6	22	51	2.67	1.32	6	15	25	1.50	0.67	76
	NonSpecificity	16	40	73	1.50	0.83	12	26	46	1.17	0.77	119
	Probability	18	56	98	2.11	0.75	4	37	50	8.25	0.35	148
Sum for x, y, z / Average for R1, R2		**40**	**118**	**222**	**2.09**	**0.96**	**22**	**78**	**121**	**3.64**	**0.60**	**343**

**R1 = y/x – 1 *R2 = z/y – 1 +Total = AW(z)+GS(z) Feq.is Frequency*

Reqs V.2. As can be seen from Table 2, in case of *AW* for *R1*, on average, we identified an additional 1.43 of uncertainties and in *R2* we identified an additional 0.51 of uncertainties. For *GS*, these percentages are 2.39 in *R1*, and 0.72 in *R2*, respectively. In total, in *R1* on average we identified additional 1.91 of uncertainties, whereas in R2 we identified on average 0.615 of unknown uncertainties.

In Table 2, one can see that we didn't have exact data (e.g., probability) and risk information available at the moment. Such data will be collected using questionnaire-based surveys in the future to quantify the identified uncertainties. In addition, we didn't observe any pattern for the occurrences of the identified uncertainties. Moreover, the Belief part of the conceptual model (e.g., concepts Belief, BeliefAgent) was derived to understand Uncertainty and is not relevant for the validation.

4.4 Lessons Learned and Future Challenges

Lack of Systematic Way of Collecting Uncertainty Requirements: Based on our experience of the application of the *U-Model* for the three case studies, we observed that there is no systematic and structured way (methodology and tool) available for precisely collecting and specifying uncertainty requirements in CPS. For our industrial case studies AW and GS, the uncertainty requirements were collected based on the experience of domain experts involved in the projects with their own understanding about uncertainty. In the phase of evaluation and during the onsite workshops, we used the *U-Model* as a reference model to ask questions in a structured, precise, and systematic manner, which eventually led to the several rounds of the refinement of the collected requirements, as we reported in Section 4.2. Based on our experience, we can conclude that there is a need of a specialized methodology with tool support for collecting and specifying uncertainty requirements. We plan to propose such a methodology with tool support based on existing requirements specification methodologies, e.g., Zen-RUCM [21].

Challenges in Understanding Uncertainty: Due to the interdisciplinary nature of CPS, we felt that it was difficult to precisely understand uncertainties. This is mainly because uncertainties not only exist in software, but also in hardware, communications, humans, and the interactions among them. Comprehending an uncertainty requires a wide range of knowledge across different disciplines. Particularly we observed that, based on our experience of collecting requirements from the industrial partners, a large number of uncertainties exist due to the mismatch between software and physical worlds. For example, for the *AW* case study, according to the implementation of the warehouse management software, at a given point in time, for a specific customer order, a specific item must be on a particular conveyor. But in the physical world, the sensor installed on the conveyor might not be able to detect the item. Therefore, we can conclude that to be able to understand uncertainties in CPS, one needs to have a wide range of knowledge of various components of CPS, their interactions and the overall functionalities of CPS as a whole.

Challenges in Quantifying Uncertainty: Another challenge is about systematically collecting data to quantify identified uncertainties based on the *U-Model* using a particular measure such as Probability. Such data are very important to support MBT of uncertainty and they should be precisely captured as part of models, from which tests should be generated. The reason of collecting such data is that uncertainty requirements are at the high level and a gap between them and system designs (often captured as e.g., UML models) should be bridged. Another reason is that a methodology with tool support is needed to help the collection of such data in a systematic and structured manner. The third reason is that domain experts with different background and experience might come up with different data; therefore a rigorous method (e.g., questionnaire-based survey) should be applied when collecting such data. It is also interesting to mention that some of this type of data (especially related to the infrastructure level of uncertainty) may also be automatically obtained from historical data (e.g., an automated infrastructure monitoring system). As part of the U-Test project, we will propose a methodology with tool support to enable the quantification of uncertainties for the purpose of enabling MBT of uncertainty of CPS.

Common Uncertainties across Domains: Though we worked with two different case studies from different domains, we were able to identify common uncertainties across the domains such as uncertainties introduced by humans due to incorrect interactions with CPS and uncertainties in networks (e.g., uncertain percentage of packet loss). Such common uncertainties will be used in the future to define reusable models capturing uncertainties and their corresponding behaviors of CPS, which can be configured (by e.g., assigning various values to the uncertain percentage of packet loss) to support test case generation for CPS of various domains. Moreover, risk information associated with common uncertainties may differ and therefore must be configurable across domains. Note that such risk information can serve as key information in test strategies to guide the generation of test cases to discover high-risk uncertainties. In summary, identifying common and reusable uncertainties can potentially reduce the overall effort required to collect and specify uncertainties. Consequently, the same or similar kinds of test strategies may be applied to generate test cases for common uncertainties. We will investigate the possibilities of reusing uncertainties across CPS domains in the future.

Model-based Uncertainty Testing with the U-Model: One of the key objectives of the development of the *U-Model* is to eventually facilitate automated testing of uncertainty of CPS. The *U-Model* will be used as the starting point to define either a domain specific language or extensions of existing modeling languages (e.g., UML, SysML), with which one can capture uncertainties together with behaviors of CPS. The ultimate objective is to enable automated MBT of CPS. We will investigate if existing OMG standards (e.g., SysML, MARTE, OCL and the UML Testing Profile (UTP)) can be used with extensions on uncertainty to enable automated MBT. In addition, we will define novel test strategies based on captured uncertainty information to generate cost-effective test cases and test data to reveal uncertainty faults in

CPS before these are actually deployed for operation. Both modeling and testing strategies based on the *U-Model* are part of near future work in the context of the ongoing EU project.

5 Related Work

Uncertainty is a term that has been used in various fields such as philosophy, physics, statistics and engineering to describe a state of having limited knowledge where it is impossible to exactly tell the existing state, a future outcome or more than one possible outcome [17]. Various uncertainty models have been proposed in the literature from different perspectives for various domains. For instance, from an ethics perspective, uncertainties are classified as objective uncertainty and subjective uncertainty, both of which are further classified into subcategories to support decision-making [5]. In healthcare, uncertainty has often been defined as "the inability to determine the meaning of illness-related events" [6] and comprehensive domain-specific uncertainty models (e.g., [7]) have been proposed, as discussed in [8].

Uncertainty is receiving more and more attention in recent years in both system and software engineering, especially for CPS, which are required to be more and more context aware [22-24]. Moreover, CPS inherently involves tight interactions between various engineering disciplines, information technology, and computer science. This magnifies uncertainties. Therefore, adequate treatment of uncertainty becomes increasingly more relevant for any non-trivial CPS. However, to the best of our knowledge, there is no comprehensive uncertainty conceptual model existing in literature that focused specifically on CPS design or on system/software engineering in general. In the remainder of the section, we discuss how the concepts uncovered during the literature review align with our proposed conceptual model.

The *U-Model* concepts BeliefAgent, BeliefStatement, and Belief of the Belief model were adapted from [11]. The author of [11] postulates that uncertainty involves a statement whose truth is expected by a person, and therefore the truth might differ for different persons (defined as BeliefAgent in our model). However, as we discussed in Section 3.1, we assigned a broader meaning to BeliefAgent: which can be an individual, a community of individuals, or a technology. The *U-Model* concepts Environment and Locality were adapted from [11, 25-27], and we related them to the other *U-Model* concepts.

Our knowledge conceptual model aligns well with the model of knowledge reported in [28]. Here the authors looked at how to manage different types of known and unknown knowledge to distinguish what is known from what is not known. Knowledge is also classified from a different perspective: something that everyone knows, tacit knowledge, conscious ignorance and meta-ignorance. Their objective is to better understand ignorance. The author of [29] also studied unknowns and provided a taxonomy particularly focusing on ignorance (named as KnownUnknown and UnknownUnknown in our conceptual model). In our conceptual model, we further elaborate these concepts and captured them as KnowledgeType, which is associated to Evidence and IndeterminacySource via EvidenceKnowledge and IndeterminacyKnowledge.

We classified uncertainties into various types including Content, Time and Occurrence. In [11], a chapter was dedicated to the discussion of content uncertainty and its measurement. The other two types of uncertainties were mentioned in [11, 13, 14], with examples but with no clear definitions provided. We adopted the measurements in our conceptual model.

Different types of sources of uncertainty for various purposes have been identified in the literature. In [30], the authors captured sources of uncertainty by considering risk and reliability analyses, based on which they classified uncertainty. The authors of [14, 31] identified sources of uncertainty in active systems. In [23, 32], the authors described the sources of uncertainty in software engineering in general. We however proposed the *U-Model* concepts IndeterminacySource and IndeterminacyNature to capture sources of uncertainty.

Aleatory and *Epistemic* uncertainties are the two generic categories of uncertainties discussed in many works [30, 33]. According to the work reported in [30], Aleatory is due to the inherent randomness of phenomena, whereas the Epistemic uncertainty is mainly due to the lack of knowledge. These two types are also covered in the *U-Model*. For example, the Non-determinism (nature of indeterminacy in *U-Model*) represents the randomness as in Aleatory, and Epistemic is covered by MissingInfo—nature of indeterminacy.

In [34], the author noted that uncertainty can occur in a random or systematic manner. In the Pattern part of the *U-Model*, we further elaborated the "systematic" concept by introducing Pattern and its sub categories.

In literature, uncertainty is often related to Risk. The acquisition project team of the US Air Force Electronic System Center (ESC) has proposed a risk matrix for evaluating risks [18]. They introduced the concepts of Risk, Impact, Likelihood of Occurrence, and Rate of Risk and also identified their relations. We reused these concepts and linked them with Uncertainty.

6 Conclusion

Cyber-Physical Systems (CPS) often consist of heterogeneous physical units (e.g., sensors, control modules) communicating via various networking equipment, interacting with applications and humans. Thus, uncertainty is inherent in CPS due to tight interactions between hardware, software and humans, and the need for them to be increasingly context aware. To understand uncertainty in the context of CPS, unified and comprehensive uncertainty conceptual model should be derived. The *U-Model* is such a conceptual model developed in an EU project, based on a thorough literature review of existing uncertainty models from various domains (e.g., philosophy, healthcare), and refined and validated with two industrial CPS case studies of various domains. Based on the results of several stages validation, we obtained the current version of the conceptual model in addition to refined uncertainty requirements. On average, we managed to learn 61.5% of unknown uncertainties that weren't explicitly specified in the uncertainty requirements collected from the two case studies.

7 Acknowledgment

This research was supported by the EU Horizon 2020 funded project U-Test (Testing Cyber-Physical Systems under Uncertainty). Tao Yue and Shaukat Ali are also supported by RCN funded Zen-Configurator project, RFF Hovedstaden funded MBE-CR project, RCN funded MBT4CPS project, and RCN funded Certus SFI.

8 References

1. Ali, S., Yue, T.: U-Test: Evolving, Modelling and Testing Realistic Uncertain Behaviours of Cyber-Physical Systems. pp. 1-2. IEEE, (2015)
2. Broy, M.: Engineering Cyber-Physical Systems: Challenges and Foundations. In: Proceedings of the Third International Conference on Complex Systems Design & Management CSD&M, pp. 1-13. (2013)
3. Huang, H.-M., Tidwell, T., Gill, C., Lu, C., Gao, X., Dyke, S.: Cyber-Physical Systems for Real-Time Hybrid Structural Testing: A Case Study. In: Proceedings of the 1st ACM/IEEE International Conference on Cyber-Physical Systems, pp. 69-78. (2010)
4. Tidwell, T., Gao, X., Huang, H.-M., Lu, C., Dyke, S., Gil, C.: Towards Configurable Real-Time Hybrid Structural Testing: A Cyber Physical Systems Approach. In: ISORC '09 Proceedings of the 2009 IEEE International Symposium on Object/Component/Service-Oriented Real-Time Distributed Computing, pp. 37-44. (2009)
5. Tannert, C., Elvers, H.D., Jandrig, B.: The ethics of uncertainty. EMBO reports 8, 892-896 (2007)
6. Mishel, M.H.: Uncertainty in illness. Image: The Journal of Nursing Scholarship 20, 225-232 (1988)
7. Babrow, A.S., Kasch, C.R., Ford, L.A.: The many meanings of uncertainty in illness: Toward a systematic accounting. Health communication 10, 1-23 (1998)
8. Han, P.K., Klein, W.M., Arora, N.K.: Varieties of Uncertainty in Health Care A Conceptual Taxonomy. Medical Decision Making 31, 828-838 (2011)
9. Cisco: Cisco Preferred Architecture for Video - Design Overview. (2015)
10. Bammer, G., Smithson, M.: Uncertainty and risk: multidisciplinary perspectives. Routledge (2012)
11. Lindley, D.V.: Understanding uncertainty (revised edition). John Wiley & Sons (2014)
12. Potter, K., Rosen, P., Johnson, C.R.: From quantification to visualization: A taxonomy of uncertainty visualization approaches. Uncertainty Quantification in Scientific Computing, pp. 226-249. Springer (2012)
13. Taylor, B.N.: Guidelines for Evaluating and Expressing the Uncertainty of NIST Measurement Results (rev. DIANE Publishing (2009)

14. Wasserkrug, S., Gal, A., Etzion, O.: A taxonomy and representation of sources of uncertainty in active systems. Next Generation Information Technologies and Systems, pp. 174-185. Springer (2006)
15. Cimatti, A., Micheli, A., Roveri, M.: Timelines with Temporal Uncertainty. In: Aaai. (2013)
16. Sprunt, B., Sha, L., Lehoczky, J.: Scheduling sporadic and aperiodic events in a hard real-time system. DTIC Document (1989)
17. ISO: ISO 31000: Risk management. (2009)
18. Garvey, P.R., Lansdowne, Z.F.: Risk matrix: an approach for identifying, assessing, and ranking program risks. Air Force Journal of Logistics 22, 18-21 (1998)
19. Klir, G.: Facets of systems science. Springer Science & Business Media (2013)
20. Yue, T., Briand, L.C., Labiche, Y.: Facilitating the transition from use case models to analysis models: Approach and experiments. ACM Transactions on Software Engineering and Methodology (TOSEM) 22, 5 (2013)
21. Zhang, G., Yue, T., Wu, J., Ali, S.: Zen-RUCM: A Tool for Supporting a Comprehensive and Extensible Use Case Modeling Framework. In: Demos/Posters/StudentResearch@ MoDELS, pp. 41-45. Citeseer, (2013)
22. Rajkumar, R.R., Lee, I., Sha, L., Stankovic, J.: Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th Design Automation Conference, pp. 731-736. ACM, (2010)
23. Conti, M., Das, S.K., Bisdikian, C., Kumar, M., Ni, L.M., Passarella, A., Roussos, G., Tröster, G., Tsudik, G., Zambonelli, F.: Looking ahead in pervasive computing: Challenges and opportunities in the era of cyber–physical convergence. Pervasive and Mobile Computing 8, 2-21 (2012)
24. Garlan, D.: Software engineering in an uncertain world. In: Proceedings of the FSE/SDP workshop on Future of software engineering research, pp. 125-128. ACM, (2010)
25. Hu, F.: Cyber-Physical Systems: Integrated Computing and Engineering Design. CRC Press (2013)
26. Cheng, B.H.C., Sawyer, P., Bencomo, N., Whittle, J.: A goal-based modeling approach to develop requirements of an adaptive system with environmental uncertainty. Model Driven Engineering Languages and Systems, pp. 468-483 %@ 3642044247. Springer (2009)
27. Wan, K., Man, K.L., Hughes, D.: Specification, analyzing challenges and approaches for cyber-physical systems (CPS). Engineering Letters 18, 308 %@ 1816-1093X (2010)
28. Kerwin, A.: None Too Solid Medical Ignorance. Science Communication 15, 166-185 (1993)
29. Smithson, M.: Ignorance and uncertainty: Emerging paradigms. Springer-Verlag Publishing (1989)
30. Der Kiureghian, A., Ditlevsen, O.: Aleatory or epistemic? Does it matter? Structural Safety 31, 105-112 %@ 0167-4730 (2009)
31. de Lemos, R., Giese, H., Müller, H.A., Shaw, M., Andersson, J., Baresi, L., Becker, B.: Software engineering for self-adaptive systems. Springer, (2013)
32. Ziv, H., Richardson, D., Klösch, R.: The uncertainty principle in software engineering. (1997)

33. Matthies, H.G.: Quantifying uncertainty: modern computational representation of probability and applications. Extreme man-made and natural hazards in dynamics of structures, pp. 105-135 %@ 1402056540. Springer (2007)

34. Bell, S.: A beginner's guide to uncertainty of measurement. National Physical Laboratory Teddington, Middlesex (2001)