Classifying Uncertainties in an Industrial Elevator with the Cynefin Framework

LIPING HAN, Simula Research Laboratory, Norway
TAO YUE, Simula Research Laboratory, Norway
SHAUKAT ALI, Simula Research Laboratory, Norway
AITOR ARRIETA, Mondragon University, Spain

Abstract Industrial elevators are complex Cyber-Physical Systems (CPSs) that operate in uncertain environments, including unpredictable passenger traffic, uncertain passenger behaviors, hardware delays, and software errors. Identifying, understanding, and classifying such uncertainties in industrial elevators is, thus, essential to enable system designers to think and reason about uncertainties, and develop specific approaches to empower such systems to deal with uncertainties systematically. To this end, we present an approach, which is based on the Cynefin framework, to classify uncertainties in industrial elevators provided by our industrial partner, Orona – a world-renowned industrial elevators developer. First, we developed a conceptual model for the Cynefin framework to enable systematic thinking of uncertainty in the context of CPSs, in general. Second, using the conceptual model, we developed a novel classification algorithm to identify the Cynefin contexts with varying degrees of uncertainties for industrial elevators. Third, we conducted an (industrial elevator) case study provided by Orona to classify various uncertain situations defined with a set of uncertain factors and their t-way interactions, based on our Cynefin conceptual model and the classification algorithm. Our results provide a mechanism for elevator designers to think, reason, and handle uncertainties in elevators in a systematic and fine-grained way.
Keywords Cynefin Framework, Uncertainty, Elevators, Quality of Service

1 INTRODUCTION

Industrial elevators are software-based Cyber-Physical Systems (CPSs) with characteristics including: 1) constantly operating in a dynamic and uncertain environment, 2) continuously evolving throughout their life-cycles, 3) elegantly handling various levels of internal uncertainties, e.g., due to implemented algorithms and external uncertainties from human interactions, environment, and uncertain information networks, and 4) gracefully dealing with situations and issues that are unknown at the design time, but may emerge during operations.

Hence, software engineering methodologies need to be revolutionized to develop industrial elevators that can operate in dynamic and uncertain environments. However, existing methodologies do not inherently handle uncertainties. For instance, a common practice of Software in the Loop (SiL) testing of industrial elevators is to simulate passengers' weights with fixed values (e.g., 75 KG in Europe) from guidelines/standards [2]. However, during any real operation, it is very difficult (if possible) to know beforehand the exact weight of a passenger, at which floor the passenger will take the elevator and which floor her/his destination will be. Thus, several open issues, such as systematically handling various extents of complexity and various levels of uncertainty inherent in industrial elevators and their environments, must be solved to develop dependable elevators. Systems designed with new methodologies that handle these issues will ensure their robustness to known and unknown eventualities inherent in themselves and their operating environments.

Before designing such methodologies, we advocate that a systematic understanding of the *complexity* and *uncertainty* is necessary. Therefore, in this paper, we propose to apply the Cynefin framework [16] (Section 3.1) to achieve this purpose. We **first** develop a conceptual model of the Cynefin framework to provide a precise understanding of various concepts, including Cynefin

Authors' addresses: Liping Han, liping@simula.no, Simula Research Laboratory, Oslo, Norway; Tao Yue, taoyue@ieee.org, Simula Research Laboratory, Oslo, Norway; Shaukat Ali, shaukat@simula.no, Simula Research Laboratory, Oslo, Norway; Aitor Arrieta, aarrieta@mondragon.edu, Mondragon University, Mondragon, Spain.

contexts and relationships among them for CPSs, in general. **Second**, we take the conceptual framework as the guidance to develop a classification algorithm for industrial elevators developed by our industrial partner, Orona – one of the largest vertical transportation builders in Europe. Given many dimensions of uncertainties, we focus exclusively on passengers' uncertainty in the SiL setup. With the algorithm, we aim to identify the Cynefin contexts, which consequently lead to systematic understanding of various extents of uncertainties. **Third**, we evaluate the algorithm with an industrial *dispatcher* from Orona to identify various contexts under various uncertain factors and their t-way interactions in traffic profiles of an elevator. Based on results, we provide lessons learned, including research implications for researchers and practical use of our tool for elevator designers.

The rest of the paper is organized as follows. Section 2 describes the industrial context. Section 3 presents the conceptual model of the Cynefin framework. Section 4 outlines our classification approach. Section 5 evaluates our algorithm with an industrial elevator case study. Section 6 presents the related work, whereas we conclude the paper in Section 7.

2 INDUSTRIAL CONTEXT

In this section, we first introduce a typical industrial elevator system developed by Orona in Section 2.1, followed by introducing the SiL practice at Orona in Section 2.2. Finally, we present a running example (Section 2.3) for illustrating our approach and algorithm in the rest of the paper.

2.1 Industrial Elevator Systems by Orona

Orona[1] designs, manufactures, installs, and maintains various types of vertical transportation systems such as elevators and escalators that are deployed in many types of buildings (e.g., hospitals, airports, supermarkets) [10]. This paper focuses only on elevator systems, which transport passengers between floors of a building efficiently, while maintaining reliability and comfortableness of the passengers. With the increase in the number of floors, the number of passengers, and the diversity of passenger traffic, the design, development, and testing of elevator systems become complicated.

Figure 1 presents a high-level view of a typical elevator system, which consists of three elevators serving N floors (*Floor 1 - Floor N*) of a building. Each floor is equipped with a dedicated control panel to collect passenger calls. Corresponding to each elevator, there is a designated controller that controls movements of the elevator between floors. All the control panels and controllers receive instructions from the *Traffic Master* via the Controller Area Network (CAN) bus. The *Traffic Master* is a software component that has an essential module called *Dispatcher*, which is responsible for handling all passenger calls. Depending on configuration and security requirements of a building, the *Traffic Master* can also be connected to a computer system via Ethernet that implements specialized access control mechanisms, e.g., restricting the access of a passenger with no security clearance to certain floors.

[1] https://www.orona-group.com/

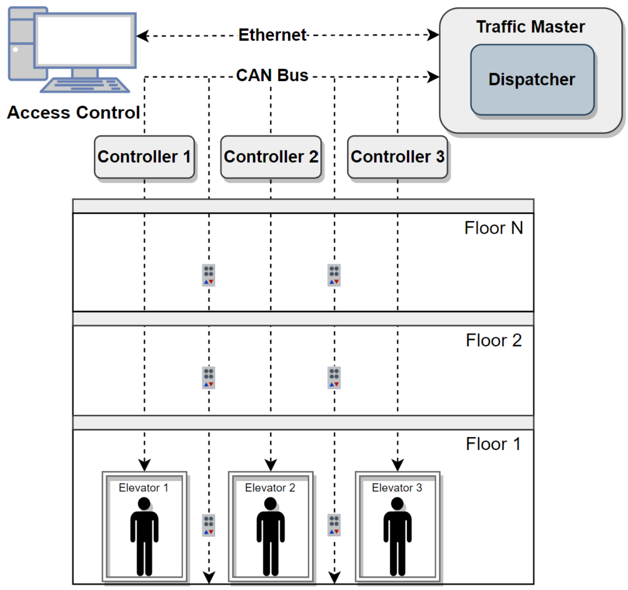

Fig. 1. A simplified elevator system

We used the *dispatcher* developed by Orona in this paper. A *dispatcher* responds to passenger calls and schedule elevators optimally at the acceptable level of Quality of Service (*QoS*). Various *QoS* attributes (Table 1) are calculated based on time list, including: Waiting Time (*WT*), Transit Time (*TT*) and Time to Destination (*TD*) defined below and also shown in Figure 2:

Definition 1 (WT). *WT is the time between a passenger registering a call on a certain floor until an elevator arrives and its door begins to open. If the elevator door is already opened when the passenger arrives, then WT for this passenger is 0.*

Definition 2 (TT). *TT is the time between the elevator door begins to open until the elevator reaches the passenger's destination floor and begins to open elevator door again.*

Definition 3 (TD). *TD is the sum of the passenger's waiting time and transit time.*

2.2 Software in the Loop (SiL) Simulations

Orona supports various types of simulations such as SiL and Hardware in the Loop (HiL). In this paper, we focus only on SiL testing of one *dispatcher* of Orona, with the commercial Elevate[2] simulator. Elevate is a software emulating the hardware of an elevator system, i.e., the *dispatcher* from Orona can be deployed and configured with various settings such as building types, passenger information (e.g., how many passengers using the elevator and what are the attributes of each passenger), and elevator information (e.g., car area, capacity).

As shown in Figure 3, Elevate requires the following inputs: 1) Analysis data including simulation settings (e.g., time slice between simulation calculations) and the selection of a *dispatcher*, either from Elevate or your own (e.g., Orona's *dispatcher*, in our context); 2) Building data including building configurations such as the floor height and the number of floors; 3) Elevator data including various elevator system design parameters, such as the number of elevators, capacity, speed and

[2]https://elevate.helpdocsonline.com/home

Table 1. Definitions of typical Quality of Service (*QoS*) attributes

QoS	Definition	Formula
AWT	Average *WT* (Definition 1) of all passengers whose calls have been answered within a certain period of time	$\left(\sum_{m=1}^{NP} wt_m\right)/NP$
LWT	The longest *WT* (Definition 1) experienced by a passenger within a certain period of time	$max\{wt_m \| m = 1, 2, ..., NP\}$
ATT	Average *TT* (Definition 2) of all passengers who completed their journeys within a certain period of time	$\left(\sum_{m=1}^{NP} tt_m\right)/NP$
LTT	The longest *TT* (Definition 2) experienced by a passenger who completed her/his journey within a certain period of time	$max\{tt_m \| m = 1, 2, ..., NP\}$
ATD	Average *TD* (Definition 3) of all passengers within a certain period of time	$\left(\sum_{m=1}^{NP} td_m\right)/NP$
LTD	The longest *TD* (Definition 3) experienced by a passenger within a certain period of time	$max\{td_m \| m = 1, 2, ..., NP\}$

wt_m, tt_m and td_m are the waiting time, transit time and time to destination of the mth passenger respectively; NP is the number of total passengers used to analyze the *QoS* of the elevator system.

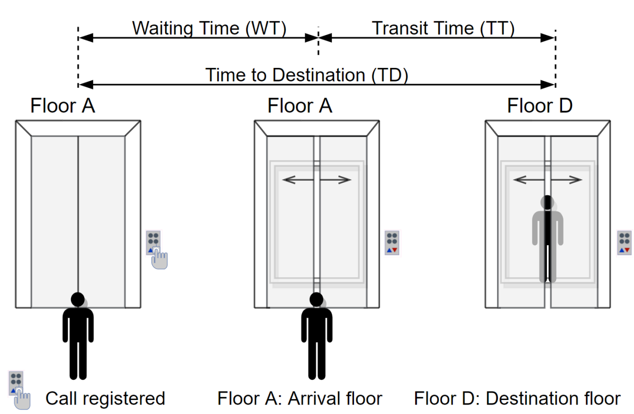

Fig. 2. Time list for a single passenger

car area; and 4) Passenger data including information about the passengers using the elevators, which contributes to the passenger traffic that the *dispatcher* needs to respond to. In addition to these necessary inputs, we can also optionally specify ways (e.g., graphs) of displaying context on simulation reports through the Report Options function. After performing all these settings, a simulation can be run. Once the simulation is completed, a set of reports are generated, including analysis reports including information such as *QoS* values, and detailed reports such as *WT* and *TT* values of each passenger. A *dispatcher* is usually tested with various traffic profiles to ensure

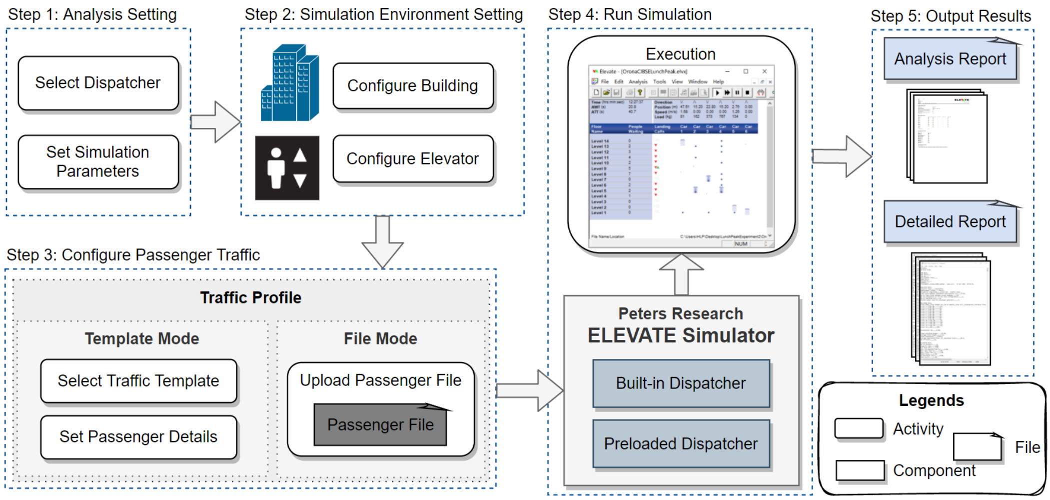

Fig. 3. Simulation Overview

that it can handle different passenger traffic with an acceptable level of *QoS*. A traffic profile is an input file to Elevate, specifies a passenger traffic of the elevator system. It contains information about a set of passengers using the elevators. For each passenger, the following information are specified: 1) *Arrival Time* - the time at which a passenger arrives on the floor and calls the elevator. It is measured in seconds and starts from the past midnight; 2) *Arrival Floor* - the floor number of the passenger's arrival; 3) *Destination Floor* - the floor number that the passenger aims to go; 4) *Mass* - the passenger's weight in kilograms; 5) *Capacity Factor* - a percentage value based on which a passenger decides whether to enter in an elevator; 6) *Loading Time* and 7) *Unloading Time* - the time required for a passenger to enter or exit the elevator, in seconds.

As shown in Figure 3, a traffic profile can either be generated by a selected traffic template in the template mode, or uploaded by a user in the file mode. The traffic template is a required configuration option of Elevate when the template mode is chosen to generate the passenger traffic. Based on a selected template and the entered parameters of passenger details, Elevate generates a list of passengers of the same values for (*Mass*, *Capacity Factor*, *Loading Time* and *Unloading Time*), and a fixed set of values for (*Arrival Time*, *Arrival Floor* and *Destination Floor*).

2.3 Running Example

We present a running example, including a SiL process, to illustrate our approach in the rest of the paper. Table 2 shows an excerpt of the *Up Peak*[3] traffic profile with 10 passengers provided by Elevate. This means that most passengers with different destinations arrive at the first floor, e.g., in the morning during a weekday in an office. The arrival time of the first passenger is 30691, which is also represented as 08:31:31 in the morning. The conversion is: 8×60×60+31×60+31.

Once the simulation is completed, Elevate outputs simulation reports. Table 3 and Table 4 show key results corresponding to the *Up Peak* traffic profile in Table 2. Table 3 presents details of each passenger's journey in simulation reports. Results including: (1) which elevator responded the call (e.g., the 2nd elevator responded to the first passenger's call, shown in the first row of the *EU* column), (2) times at which an elevator arrived and reached the destination (the *TEA*

[3]https://elevate.helpdocsonline.com/peters-research-cibse-modern-office-up-peak

Table 2. Running Example: Passenger data of 10 passengers during *Up Peak*

AT	AF	D	M	C	L	U
30691 (8:31:31)	1	2	75	70	1.2	1.2
30703 (8:31:43)	6	1	75	70	1.2	1.2
30705 (8:31:45)	1	13	75	70	1.2	1.2
30712 (8:31:52)	1	3	75	70	1.2	1.2
30727 (8:32:07)	1	12	75	70	1.2	1.2
30727 (8:32:07)	1	2	75	70	1.2	1.2
30735 (8:32:15)	1	6	75	70	1.2	1.2
30737 (8:32:17)	1	14	75	70	1.2	1.2
30738 (8:32:18)	1	5	75	70	1.2	1.2
30742 (8:32:22)	2	1	75	70	1.2	1.2

AT: *Arrival Time*; *AF*: *Arrival Floor*; *D*: *Destination Floor*; *M*: *Mass*; *C*: *Capacity Factor*; *L*: *Loading Time*; *U*: *Unloading Time*.

and *TRD* columns). For instance, for the first passenger, the 2nd elevator arrived at 8:31:31 and the passenger reached the destination floor at 8:31:45; (3) time information that determine the passengers' satisfaction: *WT* and *TT*. For instance, for the first passenger, the *WT* and *TT* are 0 and 14.1 respectively.

In addition to *WT* and *TT*, *TD* is also important. It is simply calculated by adding *WT* and *TT*. Table 4 shows the *QoS* metrics with six *QoS* attributes corresponding to Table 2 and Table 3, which are calculated based on the formulas provided in Table 1.

Table 3. Running Example: Partial simulation results corresponding to Table 2

PA	EU	TEA	TRD	WT	TT
-	2	8:31:31	8:31:45	0	14.1
-	3	8:31:55	8:32:16	12.2	20.4
-	4	8:31:45	8:32:36	0.1	50.9
-	4	8:31:52	8:32:08	0.1	15.9
-	5	8:32:07	8:33:11	0.1	64.2
-	5	8:32:07	8:32:31	0.1	23.4
-	3	8:32:15	8:32:49	0.1	34.3
-	3	8:32:15	8:33:27	0	70.2
-	3	8:32:16	8:32:36	0	17.8
-	2	8:32:22	8:32:36	0.1	14.1

PA: Passenger Attributes, the same as the seven attributes in Table 2; *EU*: Elevator Used; *TEA*: Time Elevator Arrived; *TRD*: Time Reached Destination.

Table 4. Running Example: values of *QoS* metrics corresponding to Table 2 and Table 3

AWT	LWT	ATT	LTT	ATD	LTD
1.3	12.2	32.5	70.2	33.8	70.2

3 CONCEPTUAL MODEL OF THE CYNEFIN FRAMEWORK

3.1 Cynefin Contexts

The Cynefin framework [16], originated in knowledge management, helps decision makers to make sense of, e.g., (unspecified) problems, uncertain situations, their evolution such that they can eventually come to consensuses when making decisions under uncertainty, named *sense making* [16].

In Figure 4, we present a conceptual model (as the UML class diagram notations), which summarizes the key concepts of the Cynefin framework, and its relationships with systems and their environments. Cynefin defines five contexts (also named as situations and domains in the literature): *Simple*, *Complicated*, *Complex*, *Chaotic*, and *Disorder*. Each context is associated with various extents and types of *Uncertainty*, which is characterized with two attributes: *state of Knowing* and *topic* (defining what an uncertainty is about). In the context of Cynefin, an uncertainty can be about the system and its operating environment, cause and effect relationships, and/or the problem under study, all together modeled as enumeration *Uncertainty Target*. Various extents of uncertainty about these targets determine which Cynefin context a system is situated at a given point of time.

One key aim of Cynefin is to help systematically understand the complexity of a Cynefin context/situation/domain, especially uncertainty entailed by it. The process of understanding itself is complex as well. In [15], these two types of complexity are named as objective complexity (considering as a property of a Cynefin context) and cognitive complexity (about a relation between a Cynefin context and an agent who perceives its complexity). In Cynefin, a context is mainly formed by the environment in which a system is situated. Therefore, complexity exists both in *System* and *Environment*.

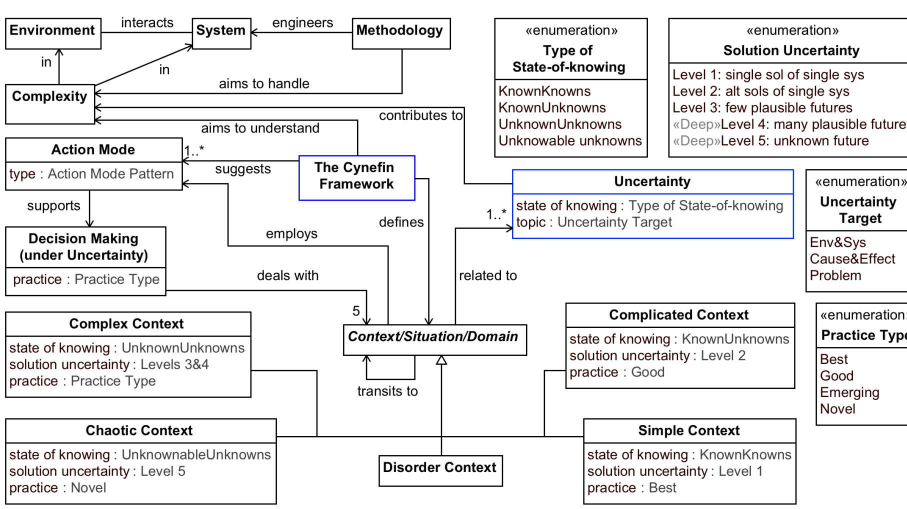

Fig. 4. Enriching the Cynefin Framework with Uncertainty

As shown in Figure 4, we characterize each Cynefin context (except for *Disorder*[4]) from four aspects: knowledge type (categorized into the four literals of enumeration *Type of State-of-knowing*),

[4]In a "disorder" situation, it is needed to gather more information, try to move to one of the other Cynefin contexts and then take the appropriate action from there. In our context, there is no such situations.

solution uncertainty (typed with enumeration *Solution Uncertainty*), and a type of practice summarizing the essence of each context (typed with enumeration *Practice Type*).

As shown in Figure 4, enumeration *Solution Uncertainty* defines five levels of uncertainties, details of which will be discussed in Section 3.2. This classification of uncertainty was borrowed from [23], where deep uncertainty was intensively discussed. We think that it is useful way to characterize solutions (also named as outcomes or futures) of a Cynefin context based on this classification.

Kurtz and Snowden [16] considered that simple and complicated contexts are ordered, while complex and chaotic contexts are un-ordered. In addition to these four, there is one disorder context. An ordered context implies that there exist known or knowable causal links (relationship between causes and effects) in the past behavior allows us to define best or good practices for future behavior. For example, in a simple context, relationships of causes and effects are known, while they are knowable (but for a given group of persons, not fully known, i.e., *KnownUnknowns*) in a complicated context. It is important to notice that in Cynefin when talking about *state of knowing*, it is from the perspective of an organization or a set of stakeholders as a collective identity, instead of referring to the knowledge of individuals.

The term of *un-order* is intentionally used in Cynefin to distinguish it from order and disorder, representing that in a dynamic and constantly-changing environment there either exist perceivable but not predicable (also named as *emergent*) cause and effect relationships (in a complex context) or not at all perceivable such relationships (in a chaotic context). On the other hand, a disorder context describes a state of disagreements among decision makers.

Kurtz and Snowden [16] also discussed in details connections and transitions among the Cynefin contexts. Theoretically it is possible to make a transition between any two contexts. However, in practice, we often observe the following transitions: 1) moving between *Simple* and *Complicated* due to, for instance, technological growth, which is the most common and incremental movement among the all; 2) moving at the complicated and complex boundary: from the knowable to the complex (e.g., exploratory moves to engine new ideas), or in a reverse direction such as selectively representing identified patterns - repeatable and predicable regularities in the world in the *Complex* context in an ordered manner. Kurtz and Snowden proposed in total 10 types of transitions among the Cynefin contexts, details of which are discussed in [16].

3.2 Uncertainty

The notion of uncertainty can be traced back to the philosophical question about the certainty of knowledge, debated by the ancient Greek philosophers, including Aristotle. Uncertainty has been given various meanings in different fields, such as philosophy, finance, and engineering. In the last decade, uncertainty, especially operational uncertainty (i.e. external uncertainty in the open and operational environment) in self-adaptive systems and CPSs, has attracted significant attention due to the fact that uncertainty is gradually being recognised as an inevitable characteristic of such systems. For instance, in the research agenda for CPSs [11], one of the key challenges is to enable CPSs to operate in increasingly uncertain, unpredictable, open, and networked environments in a robust manner.

Though the Cynefin framework explicitly links its contexts to the state of knowing and dividing them into the ordered and un-ordered domains, it does not sufficiently, which perhaps was not one of the original aims of Kurtz and Snowden when proposing Cynefin, discuss how to deal with uncertainties at each context. Below, we make effort of explicitly associating uncertainty to the Cynefin contexts.

To determine ways of dealing with uncertainty, in [23] uncertainty is classified into two extreme levels (complete certainty and total ignorance) and five intermediate levels: a clear enough future of a single-system model, alternative futures with probabilities (with a single-system model), a few plausible futures with a few alternative system models, a multiplicity of (many) plausible futures with many system models, and an unknown future (i.e. recognised ignorance). After aligning this classification with the Cynefin framework, as shown in Figure 4, we captured the various uncertainty levels as five literals of the *Solution Uncertainty* enumeration, which are referenced in the *Simple*, *Complicated*, *Complex* and *Chaotic* contexts of the Cynefin framework, respectively. Please also note that Level 4 and Level 5 are deep uncertainty (as indicated with stereotype <<*Deep*>> applied on the two literals).

When we assume that the state of knowing solutions (or future) is of known (*Level 1* uncertainty; *Simple Context*), or knowable (*Level 2* uncertainty; *Complicated Context*), existing methods (e.g., sensitivity and robustness analyses [1], statistical models that quantify uncertainties with probability [8]) are often applied to deal with such shallow uncertainties. In model-based engineering, a framework was proposed by Zhang et al. [25] to model and test CPSs under environmental uncertainty of Level 2, and Camilli et al. [4] also proposed an online model-based testing approach, based on Bayesian reasoning, to manage and mitigate Level 2 uncertainties of a system under test. For uncertainties at Level 3, scenario planning (e.g., [3]) has been proposed to support decision making. However, as stated in [9] by French, "some uncertainties about future events are too deep to agree on probabilities" (i.e., beyond Level 2 and even Level 3), and conventional analyses are not sufficient to support decision making under deep uncertainties. Walker also discussed in [23] that when facing deep uncertain situations (i,e., when relevant actors such as experts do not know or cannot agree on plausible future states or outcomes), adaptive policies (containing, e.g., emerging or novel actions as formulated for the *Complex* and *Chaotic* contexts in Cynefin) are needed to systematically discover scenarios in an exploratory manner.

4 CLASSIFYING UNCERTAIN SITUATIONS IN INDUSTRIAL ELEVATOR SYSTEMS WITH THE CYNEFIN FRAMEWORK

Figure 5 shows the overall process and key components of our approach for the classification of uncertain situations in an industrial elevator, which will be described in this section. We first present the terminologies that will be used to explain various concepts (Section 4.1), followed by the modeling of uncertain factors of passengers and the generation of uncertain traffic profiles in Section 4.2. We then present the classification algorithm for industrial elevators in Section 4.3.

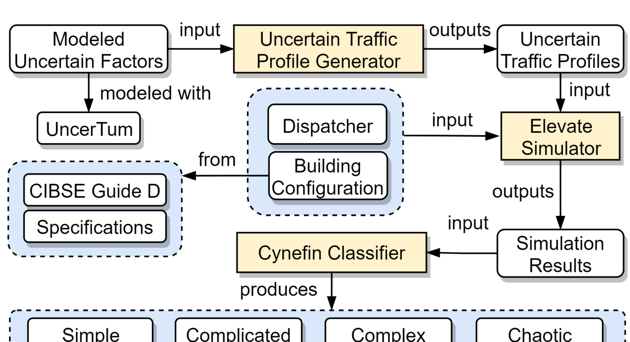

Fig. 5. Overview of uncertainty classification with the Cynefin framework in the elevator system domain

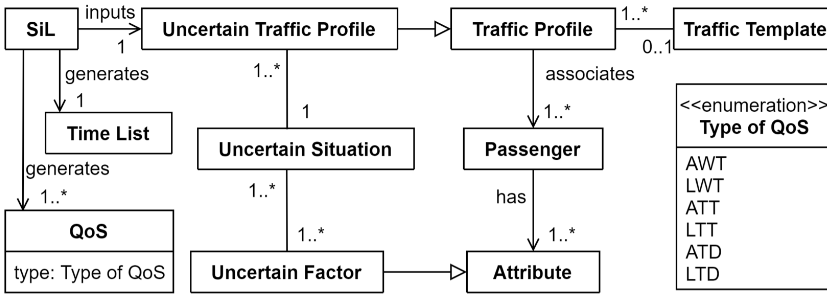

Fig. 6. Elevator system and passenger's uncertainty

4.1 Terminologies

The uncertainty in an elevator system can be studied in different development settings. Examples include uncertainties in software (e.g., uncertainties caused by *dispatcher*) where the rest of the system is simulated, uncertainties in hardware (e.g., start delay[5] due to a hardware error), and uncertainties in operation (e.g., unpredictable behaviors of passengers). In this paper we focus on uncertainties related to passengers in the SiL setting. Below, we provide definitions of key terminologies:

Definition 4 (Uncertain Factor). *All the attributes defining a traffic profile are uncertain factors (Section 2.2). For example, it is unpredictable at which time a passenger with what mass will arrive on which floor.*

Definition 5 (Uncertain Traffic Profile). *An uncertain traffic profile is a traffic profile with uncertainty inputted into Elevate to simulate an uncertain situation in SiL. Uncertainty in a traffic profile comes from unpredictable values of uncertain factor(s).*

Definition 6 (Uncertain Situation). *An uncertain situation denotes a case in SiL, in which uncertain traffic profiles are due to a single uncertain factor or by an interaction of two or more uncertain factors. Assuming an uncertain factor set U with un ($un > 1$) number of uncertain factors,* ***t-way interactions*** *among all the uncertain factors in U means t ($1 < t \leq un$) number of uncertain factors in U will be used to simulate C_{un}^{t} number of uncertain situations.*

Figure 6 shows the relationship between passengers' uncertainty and an elevator system configured in the SiL setting with Elevate. In this setting, *SiL* takes as input an *Uncertain Traffic Profile* (see the example of it in Table 2), as described by the association between *SiL* and *Uncertain Traffic Profile*, which itself is a specialization of *Traffic Profile*. A traffic profile can be generated with a specific traffic template (e.g., the excerpt for the traffic profile with the *Up Peak* traffic template in Table 2) or created by a user. A traffic template can be configured with different parameters each time then produces different traffic profiles.

A *dispatcher* configured in SiL takes input an uncertain traffic profile and generates a *Time List* containing time information of all passengers (see data reported in the *WT* and *TT* columns in Table 3) and *QoS* attributes (see the example in Table 4) as shown in Figure 6. The *QoS* was characterized with attribute *type* and enumeration *Type of QoS* defines six types of *QoS* (see the definitions in Table 4).

[5]Start delay measures the delay on the duration of the time when the elevator door is fully closed until the elevator actually starts moving.

A traffic profile contains information about several passengers as shown by the association between *Traffic Profile* and *Passenger* in Figure 6 . An example traffic profile is shown in Table 2 consisting of 10 passengers. Each passenger has several attributes (e.g., seven attributes of each passenger in Table 2) as shown in Figure 6 by an association between *Passenger* and *Attribute*. The attribute which leads to passenger's uncertainty is called passenger's *Uncertain Factor* (see Definition 4), such as the weight of a passenger (*Mass* in Table 2), time taking by a passenger to enter and exit the elevator (*Loading Time* and *Unloading Time* in Table 2). Thus, we show *Uncertain Factor* as a specialization of *Attribute* in Figure 6.

Uncertain Factor in Figure 6 shows an association with *Uncertain Situation* (see Definition 6). One uncertain factor can lead to several types of uncertain situations, by itself or due to interactions with other uncertain factors. For example, uncertain factor *Loading Time* in Table 2 can lead to different types of uncertain situations. Note that in this paper, we use *us* followed by an uncertain factor or several uncertain factors, such as *usM* and *usM-C*, to denote an uncertain situation caused by uncertain factor(s). One uncertain situation example is that all the 10 passengers' *Loading Time* in Table 2 are uncertain (*usL*); Another uncertain situation can be caused by the interaction between *Loading Time* and *Unloading Time* (*usL-U*), that is all the 10 passengers' *Loading Time* and *Unloading Time* in Table 2 are uncertain.

As shown in Figure 6, *Uncertain Situation* has an association with *Uncertain Traffic Profile* (see Definition 5). One type of uncertain situation can be simulated with several uncertain traffic profile(s) and one uncertain traffic profile can only simulate one type of uncertain situation. For example, to simulate uncertain situation *usM* based on the traffic profile without uncertainty in Table 2, we can randomize *Mass* values of 10 passengers within a reasonable range while keeping the other six attributes unchanged. For instance, for all the 10 passengers in the traffic profile shown in Table 2, we can generate one uncertain traffic profile corresponding to *usM*. The process can be repeated several times so as to generate several uncertain traffic profiles for *usM*.

4.2 Uncertain Traffic Profile Generation

To simulate uncertain situations, we implemented an *Uncertain Traffic Profile Generator* (Figure 5), which generates uncertain traffic profiles based on modeled uncertain factors.

Modeling Uncertain Factors. We study uncertain factors in the traffic profiles, to model uncertainties caused by uncertain factors, we employ UncerTum [25], which is a UML based approach for modeling/measuring uncertainties with probabilities, vagueness and ambiguity. Figure 7 presents an example of measuring *Mass* as an interval with the min and max values being 70KG and 80KG, respectively. In addition, we implemented a simple mechanism by setting an interval around standard values of these factors of a certain region (e.g., Europe). For instance, the European standards assume each passenger has a *Mass* of 75KG [2] and this average is typically used for simulations.

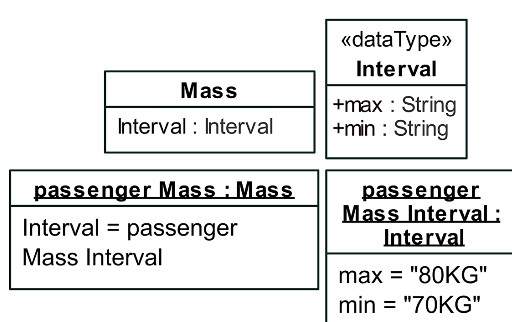

Fig. 7. Modeling uncertain factor *Mass*

Generating Uncertain Traffic Profiles. To simulate uncertain situations, we generate uncertain traffic profiles with Algorithm 1. Assuming a set of uncertain factors $U = \{u_i | i = 1, 2, ..., un\}$, where u_i is the ith uncertain factor (e.g., uncertain *Mass*), un is the number of uncertain factors, the uncertain situations caused by uncertain factor set U can be represented as $US = \{us_{i,j} | i =$

$1, 2, ..., un; j = 1, 2, ..., C^i_{un}\}$, where $us_{i,j}$ is the jth uncertain situation caused by the i number of uncertain factors. The uncertain situations cover those caused by a single uncertain factor ($i = 1$) and t-way interactions among uncertain factors (see Definition 6) among all the uncertain factors ($1 < i \leq un, t = i$).

The number of all possible uncertain situations generated under the uncertain factor set $U = \{u_i | i = 1, 2, ..., un\}$ is $C^1_{un} + C^2_{un} + ... + C^{un}_{un} = 2^{un} - 1$. Consequently, US in total will have $2^{un} - 1$ types of uncertain traffic profiles covering all the uncertain situations caused by the single uncertain factor and t-way interactions between several uncertain factors in U. For instance, for studying three uncertain factors in Table 2, e.g., *Mass*, *Loading* and *Unloading*, the number of uncertain situations caused by one single uncertain factor, 2-way interactions and 3-way interactions are: 3 (C^1_3, $\{us_{1,1} = usM, us_{1,2} = usL, us_{1,3} = usU\}$), 3 ($C^2_3$, $\{us_{2,1} = usM-L, us_{2,2} = usM-U, us_{2,3} = usL-U\}$), and 1 ($C^3_3$, $\{us_{3,1} = usM - L - U\}$), respectively. Consequently the uncertain situation set US totally has 7 types of ($C^1_3 + C^2_3 + C^3_3$) uncertain situations.

Algorithm 1: Uncertain Traffic Profiles Generation

Input: Uncertain factors U, Interval I, number of repetitions per generation NR, baseline profile P_b
Output: Uncertain traffic profiles P

for $us_{i,j}$ *in* US **do**
  for $k = 1, ..., NR$ **do**
    $P_{i,j,k} = P_b$
    for $p_{i,j,k,m}$ *in* $P_{i,j,k}$ **do**
      for u_i *in* $us_{i,j}$ **do**
        randomize u_i of $p_{i,j,k,m}$ using Eq. 1
    if *average value of* u_i *in* $P_{i,j,k}$ *equals to the value of* u_i *in* P_b **then**
      save $P_{i,j,k}$

Uncertain traffic profiles of an uncertain situation are generated by randomly generating a value for each of its uncertain factors within corresponding predefined intervals based on a baseline profile P_b (i.e., the original traffic profile without uncertainties). For example, to generate uncertain traffic profiles for uncertain situations caused by uncertain factors *Mass* and *Loading Time* (*usM-L*), we need to randomize values of *Mass* and *Loading Time* of each passenger based on a baseline traffic profile. The values of uncertain factors in the baseline traffic profile are the recommended values based on the CIBSE Guide D [2], e.g., the *Mass* of all passengers in Table 2 is 75KG, which is the recommended value for *Mass* in Europe. With these recommended values we can define interval set $I = \{I_i | i = 1, 2, ..., un\}$, where, $I_i = [a_i, b_i]$ is the interval for uncertain factor u_i, which can be defined based on the value of u_i in the baseline profile, the minimum and maximum values of the interval are chosen based on $\pm x$. For example, in Figure 7, the recommended value for *Mass* in Europe is 75KG, whereas by choosing $x = 5$, we obtain the minimum $a_i = 70$ KG and the maximum $b_i = 80$ KG for the interval.

Since we randomly generate values for each uncertain factor from its defined interval, we repeat the generation NR times for each uncertain situation. The kth uncertain traffic profile generated under uncertain situation $us_{i,j}$ can be represented as $P_{i,j,k} = \{p_{i,j,k,m} | i = 1, 2, ..., un; j = 1, 2, ..., C^i_{un}; k = 1, 2, ..., NR; m = 1, 2, ..., NP\}$, where $p_{i,j,k,m}$ is the mth passenger of uncertain traffic profile $P_{i,j,k}$.

As shown in Algorithm 1, we first initialize the uncertain traffic profile $P_{i,j,k}$ with the baseline profile P_b (Line 3), then we randomize the uncertain factor's value of each passenger in uncertain traffic profile $P_{i,j,k}$ within the interval (Line 6). The value of uncertain factor u_i of the mth passenger in uncertain traffic profile $P_{i,j,k}$ can be randomized as:

$$vu_{i,m} = RAND() * (b_i - a_i) + a_i \tag{1}$$

To avoid the generated uncertain traffic profiles deviating too much from the baseline traffic profile, we only keep those profiles whose average values of each uncertain factor are equal to their corresponding values in the baseline profile (Lines 7-8). For example, suppose we want to generate uncertain traffic profiles of the baseline profile in Table 2 based on uncertain factor *Mass*, then we randomize 10 passengers' *Mass* and only keep profiles that have the average value of *Mass* of the 10 passengers equal to 75KG.

4.3 Uncertain Situation Classification

An execution/simulation, with the *dispatcher* and Elevate, under an uncertain situation $us_{i,j}$ simulated by a specific traffic profile $P_{i,j,k}$ (see an example in Table 2), produces time list of all passengers $T_{i,j,k}$ (e.g., data reported in the *WT* and *TT* columns in Table 3) and a set of values for the *QoS* metrics (e.g., *AWT*, see the example in Table 4). After running simulations with all generated uncertain traffic profiles P of all uncertain situations US, we classify, with Algorithm 2, each uncertain situation $us_{i,j}$ into the Cynefin contexts.

As shown in Algorithm 2, output $C = \{c_{i,j,l} | i = 1, 2, ..., un; j = 1, 2, ..., C^i_{un}; l = 1, 2, ..., nq\}$, $c_{i,j,l}$ is the classified Cynefin context of uncertain situation $us_{i,j}$ for the lth *QoS*, where nq is the total number of *QoS* attributes (Table 4).

Q (Line 1) is the set of values for *QoS* metrics represented as $Q = \{Q_{i,j,k} | i = 1, 2, ..., un; j = 1, 2, ..., C^i_{un}; k = 1, 2, ..., NR\}$, $Q_{i,j,k} = \{q_{i,j,k,l} | l = 1, 2, ..., nq\}$, $Q_{i,j,k}$ is the kth *QoS* metrics under the jth uncertain situation that changes i uncertain factor(s) in U, $q_{i,j,k,l}$ is a value of the lth *QoS* attribute in $Q_{i,j,k}$. T (Line 1) is the set of time lists, which can be further represented as $T = \{T_{i,j,k} | i = 1, 2, ..., nu; j = 1, 2, ..., C^i_{un}; k = 1, 2, ..., NR\}$, $T_{i,j,k}$ means the kth time list generated under the jth uncertain situation that changes i number of uncertain factor(s) in U.

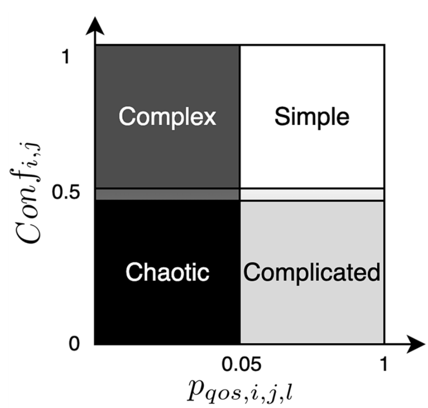

Fig. 8. Classification Rules

Lines 2-4 classify the Cynefin contexts based on collected *QoS* metrics, in a coarse-grained manner. Simulating one uncertain situation NR times produces NR values for each *QoS* attribute. For each *QoS* attribute, these values are compared with one value of the baseline traffic profile using the one sample Wilcoxon signed rank test at the significance level of 0.05. If a resulting p-value is less than 0.05 then it means that the uncertain situation has significant impact on a specific *QoS* attribute. The calculation is as follows:

$$p_{qos,i,j,l} = OSW((q_{i,j,1,l}, ..., q_{i,j,NR,l}), q_{bas,l}) \tag{2}$$

Lines 5-7 is a fine-grained way of classification; the NR generated time lists are compared with the one for the baseline traffic profile, using the two-samples Wilcoxon signed rank test at a significance level of 0.05, which results NR p-values:

$$p_{dis,i,j,k} = TSW(T_{i,j,k}, T_{bas}), \tag{3}$$

Algorithm 2: Cynefin Classification Algorithm

Input: Simulation results SR
Output: Classification results C
1 Extract Q and T from SR
2 **for** $Q_{i,j,k}$ *in* Q **do**
3   **for** $q_{i,j,k,l}$ *in* $Q_{i,j,k}$ **do**
4   get $p_{qos,i,j,l}$ using Eq. 2
5 **for** $T_{i,j,k}$ *in* T **do**
6   **for** $k = 1, ..., NR$ **do**
7   get $p_{dis,i,j,k}$ using Eq. 3
8   get $Conf_{i,j}$ using Eq. 4
9 **for** $us_{i,j}$ *in* US **do**
10   **for** $l = 1, ..., nq$ **do**
11   **if** $p_{qos,i,j,l} \geq 0.05$ **then**
12   **if** $Conf_{i,j} > 0.5$ **then**
13                 $c_{i,j,l}$ = Simple
14   **else if** $Conf_{i,j} < 0.5$ **then**
15                 $c_{i,j,l}$ = Complicated
16   **else**
17                 $c_{i,j,l}$ = Border(Simple, Complicated)
18   **else**
19   **if** $Conf_{i,j} > 0.5$ **then**
20                 $c_{i,j,l}$ = Complex
21   **else if** $Conf_{i,j} < 0.5$ **then**
22                 $c_{i,j,l}$ = Chaotic
23   **else**
24                 $c_{i,j,l}$ = Border(Complex, Chaotic)

In Equations 2 and 3, $OSW(x, y)$ and $TSW(x, y)$ are one sample and two-samples Wilcoxon signed rank tests, respectively, both of which are one-tailed tests for determining whether an uncertain situation leads to significantly longer time; $p_{qos,i,j,l}$ is the p-value of the lth QoS attribute under uncertain situation $us_{i,j}$; $q_{bas,l}$ is the value of the lth QoS attribute generated by the baseline profile; $(q_{i,j,1,l}, ..., q_{i,j,NR,l})$ is an array of all the values of the lth QoS attribute generated by NR uncertain traffic profiles of $us_{i,j}$; $p_{dis,i,j,k}$ is the p-value of the time list generated by uncertain traffic profile $P_{i,j,k}$; and T_{bas} is the time list generated by the baseline profile.

Line 8 obtains a value denoting the confidence of an observation about an uncertain situation $us_{i,j}$ not leading to a significant impact on generated time lists. Simply put, a confidence is the number of p-values that are great than or equal to 0.05 out of the total comparisons. For example, if $NR = 10$, and among all 10 p-values, eight of them is greater than or equal to 0.05, we then say with the 80% confidence that the 10 time lists are not significantly worse than the baseline one, as calculated below:

$$Conf_{i,j} = (NR - IC_{i,j})/NR \tag{4}$$

$$IC_{i,j} = \sum_{k=1}^{NR} Sym(p_{dis,i,j,k}), \tag{5}$$

Where $IC_{i,j}$ is the number of uncertain traffic profiles generated under the jth situation changing i number of uncertain factor(s), which have led to a significant influence on time list(s); $Sym(x)$ is a symbolic function defined as:

$$Sym\,(x) = \begin{cases} 0 & x \geq 0.05 \\ 1 & x < 0.05 \end{cases} \tag{6}$$

The other lines of Algorithm 2 perform classifications (also visualized in Figure 8). For example, if no significant impacts are observed from all comparisons of *QoS* attributes (e.g., *AWT*) and time lists (e.g., *WT*), it is a *Simple* context, as $p_{qos,i,j,l} \geq 0.05$ and $Conf_{i,j} = 1$ (which is greater than 0.5, satisfying the condition in Line 12 of Algorithm 2). Figure 8 also shows the border classifications, e.g., between *Simple* and *Complicated* for cases of $p_{qos,i,j,l} \geq 0.05$ and $Conf_{i,j} = 0.5$.

5 INDUSTRIAL EVALUATION

In this section, we present details of our industrial case study (Section 5.1) followed by results in Section 5.2. Finally, we discuss lessons learned in Section 5.3.

5.1 Research Questions, Settings and Execution

Research Questions (RQs). RQ1: Can we identify all the Cynefin contexts by simulating various uncertain factors? This RQ studies the completeness of the Cynefin framework for classifying the contexts in the elevator system. **RQ2:** How are *QoS* attributes impacted by uncertain situations across two types of traffic profiles? This RQ investigates whether uncertain situations of different traffic templates (e.g., *Up Peak*, *Lunch Peak*) have divergent influence on the *QoS* attributes. **RQ3:** Do interactions among various uncertain factors affect the classification? This RQ aims to study whether the interactions of different uncertain factors have different degrees of impacts on the *QoS* attributes.

Settings. We used an office building configuration from Section 4.8 of CIBSE Guide D [2] having 6 elevators, 14 floors, and 1120 persons (80 persons per floor). For each elevator, it has the following settings: Capacity (KG): 1600; Car area (m^2): 3.56; Speed (m/s): 2.5; Acceleration (m/s^2): 0.8 and Jerk (m/s^3): 1.0. We also selected two traffic templates for generating two baseline traffic profiles: modern office up peak[6] (*Up Peak*) and modern office lunch peak[7] (*Lunch Peak*). *Up Peak* models the up peak traffic between 08:30 and 09:30, comprising of 1150 passengers. *Lunch Peak* models the lunch peak traffic between 12:15 and 13:15, comprising 1409 passengers. Figure 9 and Figure 10 present the overall passenger activity of *Up Peak* and *Lunch Peak* respectively. These plots are generated from the Elevate software. Obviously, *Up Peak* has a majority of incoming traffic, *Lunch Peak* has nearly half incoming and half outgoing (with a minimum inter-floors) traffic. This is because most people came to the office for work around 9:00am, and went out for lunch, then came back after lunch during the lunch break. After selecting the traffic template, the passenger details need to be set uniformly. We set *Mass* to 75KG (i.e., average weight in Europe [2]), and maximum *Capacity Factor* which is usually 80% [2], but was set to 70% based on examples from [2]. *Loading* and *Unloading Times* are set to 1.2 seconds, which is the average time to enter and exit the elevator

[6]https://elevate.helpdocsonline.com/peters-research-cibse-modern-office-up-peak

[7]https://elevate.helpdocsonline.com/peters-research-cibse-modern-office-lunch-peak

according to [2]. When all the settings are completed, we run Elevate and obtained the baseline profiles for further generation of uncertain traffic profiles.

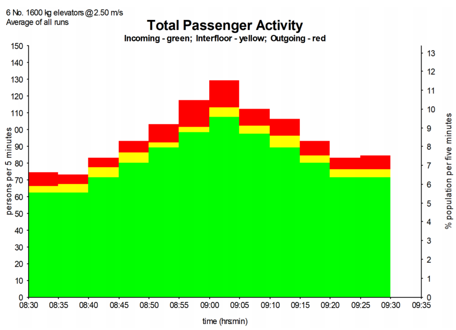

Fig. 9. *Up Peak* Passenger Activity

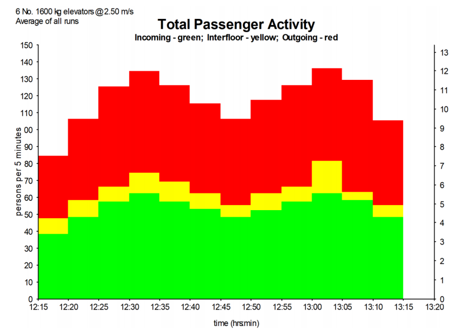

Fig. 10. *Lunch Peak* Passenger Activity

Execution. We study uncertain factors related to *Mass*, *Capacity Factor*, *Loading Time*, and *Unloading Time* of passengers, since changing the rest of the factors (e.g., *Arrival Time*, *Arrival Floor*) will change the traffic template provided by Elevate making them unrealistic. To generate uncertain traffic profiles, we randomize values of uncertain factors (i.e., *Mass*, *Capacity Factor*, *Loading Time*, and *Unloading Time*) with intervals. The intervals for all uncertain factors are shown in Table 5. We set $x = 20\%$ for each uncertain factor except for *Capacity Factor*. This is because the maximum value of *Capacity Factor* is usually 80% [2] and the value of the baseline is 70%, so we use ±10, instead of ±20%, to avoid generating values over 80%. The four uncertain factors lead to 15 uncertain situations (the size of US being 15) corresponding to 4 single way and 11 t-way interactions. We generated $NR = 10$ traffic profiles for each uncertain situation, resulting into 150 uncertain traffic profiles for each type of traffic template (i.e., *Up Peak* and *Lunch Peak*). As a result, we obtained 300 (150×2) uncertain traffic profiles, in total.

Table 5. Intervals of Uncertain Factors

Uncertain Factors	Baseline	Interval
Mass (KG)	75	[60, 90]
Capacity Factor (%)	70	[60, 80]
Loading Time (s)	1.2	[0.96, 1.44]
Unloading Time (s)	1.2	[0.96, 1.44]

5.2 Results and Analyses

RQ1. Table 6 summarizes the classification results for *Up Peak* and *Lunch Peak* traffic templates in terms of the percentage of uncertain situations (combined with different *QoS* attributes) classified into a Cynefin context or being on the border of two adjacent contexts. First, we see that all the four contexts were identified in both types of traffic templates. Second, for both types of traffic templates, the majority of the uncertain situations were classified into *Simple* (68.89% for *Up Peak*, 44.44% for *Lunch Peak* and 56.67% in total), implying that most of the uncertain situations caused by four uncertain factors have no significant influence on the performance of the *dispatcher*. We can also see that a few uncertain situations were on the borders between *Simple* and *Complicated*, and between *Complex* and *Chaotic*, considering collected data does not allow us to make a clear distinguishing of two adjacent contexts, which are transferable when more data collected.

Concluding Remarks. Our classification algorithm can identify all types of Cynefin contexts, and the distributions of the contexts across the two types of traffic profiles are comparable.

Table 6. Overall Results of Classification–RQ1

Cynefin context	UpPeak(%)	LunchPeak(%)	Total(%)
Simple	68.89	44.44	**56.67**
Border(Simple,Complicated)	3.33	5.56	4.44
Complicated	11.11	4.44	7.78
Complex	6.67	24.44	15.56
Border(Complex,Chaotic)	1.11	3.33	2.22
Chaotic	8.89	17.78	13.33

RQ2. Figures 11 and 12 show the classification results of *Up Peak* and *Lunch Peak*. Each row and each column of the figures are the studied *QoS* attribute and uncertain situation, respectively. For example, the column *usC-L-U* studies the impact of the 3-way interactions of *Capacity Factor*, *Loading time*, and *Unloading Time* on the six different *QoS* attributes.

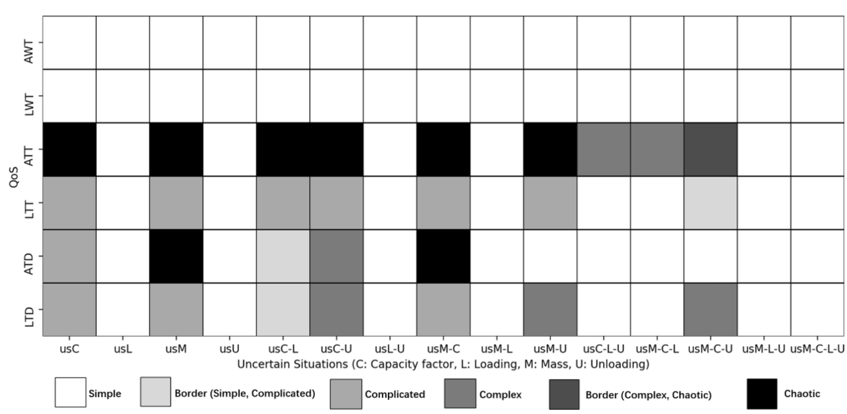

Fig. 11. Classification of Contexts for Up Peak–RQ2 and RQ3

Up Peak. From Figure 11, one can observe that, for *AWT* and *LWT*, we observed the same results that all uncertain situations are classified as *Simple*. These results suggest that *AWT* and *LWT* weren't affected by the uncertain situations. For *ATT*, 6, 2, 1 and 6 (out of 15) are *Simple*, *Complex*, *Border(Complex, Chaotic)* and *Chaotic*, respectively, which shows that most of the uncertain situations had significant impact on *ATT*. For *ATD*, 10, 1, 1, 1 and 2 of the 15 uncertain situations are classified into *Simple*, *Border(Simple, Complicated)*, *Complicated*, *Complex* and *Chaotic* respectively, which shows that most of the uncertain situations had various extents of impact on *ATD*. For *LTT*, 8, 1 and 6 are classified as *Simple*, *Border(Simple, Complicated)* and *Complicated*. For *LTD*, 8, 1, 3 and 3 are classified as *Simple*, *Border(Simple, Complicated)*, *Complicated* and *Complex*. Thus, we can conclude that uncertain situations have relatively small impact on *LTT* and *LTD*. Overall, we can observe that 75 out of 90 (6×15) cases, for the *Up Peak* profile, have been classified as *Simple*,

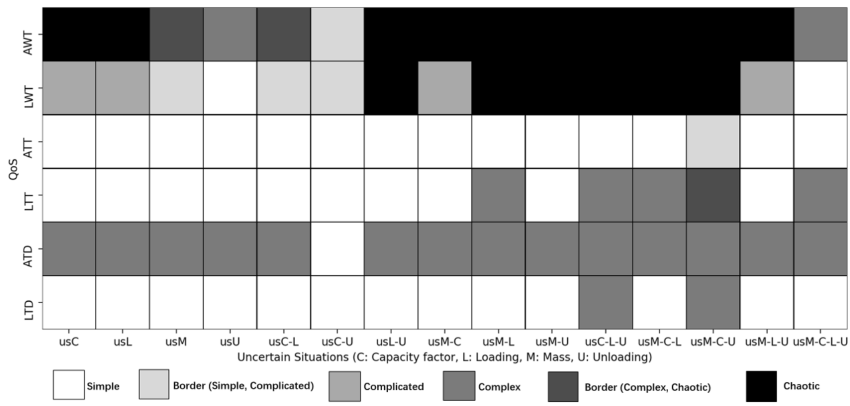

Fig. 12. Classification of Contexts for Lunch Peak–RQ2 and RQ3

Complicated, or the border between them, indicating that most of the situations for *Up Peak* are not deep uncertainty.

Lunch Peak. From Figure 12, we can observe that, for *AWT*, 1, 2, 2 and 10 (out of 15) are on *Border(Simple, Complicated)*, *Complex*, *Border(Complex, Chaotic)* and *Chaotic*, respectively, suggesting that the uncertain situations have large impact on *AWT*. For *LWT*, 2, 3, 4, and 6 (out of 15) are *Simple*, Border(Simple, Complicated), *Complicated* and *Chaotic*, also showing that *LWT* is heavily affected by the uncertain situations. For *ATT*, 14 out of 15 are *Simple* and one is on *Border(Simple, Complicated)*, whereas for *LTT*, 10 out of 15 are *Simple*, only 4 are *Complex* and 1 is on *Border(Complex, Chaotic)*. These results tell that the uncertain situations nearly have no impact on *ATT* and have very small influence on *LTT*. For *ATD*, 14 out of 15 are *Complex* and only 1 is *Simple*, which means the uncertain situations also have large impact on *ATD*. For *LTD*, 13 are *Simple* and only 2 are *Complex*, which means most of the uncertain situations have no influence on *LTD*. Overall, we can observe that 41 out of 90 (6×15) cases, for *Lunch Peak*, were classified as deep uncertainty, i.e., *Complex*, *Chaotic*, or the border between them.

When comparing *Up Peak* and *Lunch Peak*, one can note that more uncertain situations have been classified into the deep uncertainty domain for *Lunch Peak* than for *Up Peak*. Moreover, such differences are reflected on different *QoSs*. For instance, more chaotic situations were identified in terms of *ATT*, for *Up Peak*, than for *Lunch Peak*. *Lunch Peak*, instead, reveals significantly more chaotic situations in terms of *AWT* and *LWT*. One plausible explanation is that, for *Up Peak*, most of requests are for going up from the ground floor, which consequently leads to long transit times (measured with *ATT* and *LTT*) as elevators need to frequently come down to the ground floor to pick up customers. For *Lunch Peak*, there are lots of both up and down requests, which subsequently leads to long waiting time, but not necessarily long transit time as elevators do not need to frequently come down to the ground floor as for *Up Peak*.

Concluding Remarks. The impact of uncertain situations on *QoS* attributes is dependent on types of traffic templates. As shown by our experiment, the uncertain situations have significant impact on *AWT* and *LWT* in *Lunch Peak*, but no significant influence on *AWT* and *LWT* in *Up Peak*, and more deep uncertainty observed for *Lunch Peak* than *Up Peak*. Hence, we suggest applying our classification algorithm for each type of traffic templates to understand the impact of uncertain situations on *QoS* attributes, separately.

RQ3. When looking at the columns of Figures 11 and 12, we can see that interactions among various uncertain factors have various degrees of impacts on the classification. This means that different uncertain factors and their interactions affect the *QoS* attributes differently. For example, for *Up Peak* (Figure 11) uncertain factor *Mass* and interactions of *Mass* and *Capacity factor* easily lead to situations with the high degree of uncertainty (see columns *usM* and *usM-C* in Figure 11), i.e., 2 *Chaotic* contexts. *usL*, *usU*, *usL-U*, *usM-L*, *usM-L-U* and *usM-C-L-U* shows no significant impact on *QoS*. In contrast, for *Lunch Peak*, *usC-L-U* have 2 *Chaotic* and 3 *Complex*, *usM-C-U* and *usM-C-L* both have 2 *Chaotic* and 2 *Complex* contexts. This indicates that interactions of uncertain factors in *Lunch Peak* are more likely to lead to complex uncertain *QoS*. Also, uncertain situation *usC-U* has little impact on *QoS*.

Concluding Remarks. There is no clear cut answer on whether a particular type of uncertain situations (e.g., 2-way vs. 3-way, *usM* vs. *usM-C-U*) plays a prominent role on affecting *QoS*. Thus, we recommend studying each t-way interaction depending on the available time budget to better understand the influence of uncertain situations on *QoS*.

5.3 Lessons Learned

Lesson 1 - Research Implications. We studied the impact of various uncertain factors and their interactions related to individual passengers on the performance of elevators, as compared to a typical way of using the fixed values for all passengers. Thus, we provide a new way of systematically studying uncertainties in elevators that can be used by other researchers. Moreover, our work is a novel application of the Cynefin framework. Researchers can further study and understand uncertain situations in the vertical transportation domain and beyond (e.g., autonomous driving) and devise novel system engineering methods to deal with such uncertain situations automatically and systematically.

Lesson 2 - Practical Use for Industrial Elevator Developers. Our work provides a tool for practitioners, especially elevator designers, to identify various Cynefin contexts, aiming to study how various uncertain factors and their interactions can lead to various degrees of uncertainty in terms of a diverse set of *QoS* attributes in a particular installation of an elevator. Identifying such contexts will help practitioners to systematically understand uncertain factors. Consequently, they can use classification results to improve the performance of their *dispatcher* algorithms, e.g., by defining specialized uncertainty handling mechanisms corresponding to each context. Moreover, new methods can be designed to mitigate and reduce uncertainties in *QoS* by moving from a high uncertain context (e.g., *Complex* to a low one e.g., *Complicated*).

Lesson 3 - Having the Cynefin framework as the Backbone of Uncertainty-wise SiL. Our classification method has the potential to be part of the SiL simulation software (e.g., Elevate) for industrial elevators that can consider uncertainties explicitly during the SiL configuration. More specifically, for instance, Elevate, with the help of our solution, can generate uncertain traffic profiles based on uncertain factors interested by elevator designers, execute them, classify simulation results into the Cynefin contexts, and returns the classification results to the elevator designers who can then be informed to pay special attention to complex and chaotic uncertain situations. Doing so can facilitate the early detection of uncertain situations, consequently ensuring the development of high quality elevator systems.

Lesson 4 - Focused and Prioritized Testing of *dispatcher* with Uncertain Traffic Profiles. The results showed that some uncertain traffic profiles lead to the contexts with high complexity and uncertainty (e.g., Complex and Chaotic contexts) meaning that the uncertain factors in the traffic profiles have significant impact on the *QoS*. Consequently, the classification results can help elevator testers to prioritize test cases for *dispatcher* under test (e.g., for different releases) by focusing exclusively on uncertain scenarios with high degree of uncertainty (e.g., Complex, or Chaotic) instead of focusing on extensive testing on all the contexts. This will result in reduction in testing cost.

Lesson 5 - Relations of Traffic templates with Uncertain Situations. We experimented with *Up Peak* and *Lunch Peak* profiles and concluded that the impact of uncertain situations depends on the traffic templates. For example, the uncertain situation *usC-L-U* has a very small impact on *QoS* attributes during *Up Peak*, but has big influence on *QoS* attributes during *Lunch Peak*. Overall, we also find that the uncertainty in traffic profiles has more impact on *QoS* attributes during *Lunch Peak* than *Up Peak*. Both of these observations imply that the uncertain factors and traffic templates contribute to the degrees of influence on *QoS* attributes of the elevator system. Therefore, to systemically and comprehensively understand the impacts of diverse contexts on *QoS* attributes, several traffic templates (e.g., *Down Peak*, *Full Day*) are recommended to be considered during the uncertain traffic profile generation.

Lesson 6 - Unclear Boundaries at the Adjacent Cynefin Contexts. When we analyzed the classification results, we discovered that a few uncertain situations fell on the borders between adjacent contexts (e.g., at the border of *Simple* and *Complicated*, and *Complex* and *Chaotic* as shown in Table 6). This might be due to the reason that there wasn't sufficient data available to make a clear distinction between two adjacent contexts in some cases. Once more data become available, the boundaries may become more clearer.

Lesson 7 - Moving Uncertain Situations from one Context to Another. We observe that going from one Cynefin context to another is possible as it can be seen that some of our uncertain situations lie on the borders of the contexts (e.g., uncertain situation *usC-L* lies on the border between *Simple* and *Complicated* contexts in terms of *QoS* attribute *ATD* during *Up Peak*). The movement of uncertain situations from one context to another gives us insights into handling and reducing uncertainties and improving *QoS* of elevators. More specifically, it will help to develop *dispatcher* that can automatically make suitable scheduling decisions (also called *take actions* in Cynefin framework) according to the Cynefin contexts. Such decisions will be aimed at moving an uncertain situation from a context with high degree of uncertainty, e.g., *Chaotic*, to the adjacent context, i.e., *Complex* resulting in reduced uncertainty.

5.4 Threats to Validity

We used only one *dispatcher* algorithm, two traffic templates (*Up Peak* and *Lunch Peak*), four uncertain factors, six *QoS* attributes, and one building configuration. However, studying uncertainty in this setting already resulted in a complex case study. Nonetheless, we will improve our experiments with additional *dispatchers*, traffic templates, uncertain factors, *QoS* attributes, and building configurations. To deal with randomness in the uncertainty generation, for each uncertain situation, we repeated simulations 10 times. However, more repetitions will bring more confidence in the results. To classify the Cynefin contexts, we used one-sample and two-samples Wilcoxon tests as they fit our purpose. However, we will investigate other relevant tests in the future.

6 RELATED WORK

Since the Cynefin framework has been proposed [16], it has been practiced in very diverse contexts to support decision-making and strategy development in constantly-changing situations, such as improving decision making in the domain of economics [12], facilitating systems thinking in organization management [5], and understanding the complexity of issues and identifying appropriate strategies in health promotion [22]. According to the Cynefin framework, uncertainty is reduced, e.g., with more knowledge from *Chaotic* to *Complex* to *Complicated* to *Simple*. In medical field, Cynefin was applied in the practice of osteopathy in [17], which can help better understanding clinical reasoning so as to support the decision-making process. In [14], Cynefin was used to classify management approaches based on uncertainty criterion, so as to select appropriate project management tools.

In software engineering, it has been explored in software process management to detect important factors that have direct and significant impact on software development processes and therefore be robust to handle complex situations [19]. Moreover, McLeod and Childs [18], in the domain of information science, conducted an exploratory evaluation on using Cynefin as a research tool for identifying issues and practical strategies for managing electronic records. [13] proposed a vision for applying the Cynefin framework to assist AI bots in performing tasks. To increase the security of IoT systems, [20] applied Cynefin in DevOps to help find situations that can be optimized, e.g., those under which a system falls into the Chaotic context, in the DevOps process during the design phase. In contrast to these works, our work is a novel application of Cynefin framework in the context of CPSs to identify various contexts with the eventual aim of handling uncertainty in each context systematically.

Uncertainties in elevator systems are common, especially due to inherent uncertain behavior of passengers. In addition, the passenger traffic in elevators is also unpredictable, e.g., when a passenger will arrive to which floor, where to go, and her/his weight. To this end, some works have focused on generating uncertain traffics to study the performance of elevators. For instance, the authors of [21] presented an approach to generate traffics with Poisson and geometric Poisson distributions to study the robustness of elevators. We see the opportunity to introduce this work to our classification framework to determine the Cynefin contexts when a traffic is generated according to Poisson and geometric distributions. We will investigate this in future. Fuzzy logic has been used, e.g., to measure passengers' satisfaction [24], optimize energy [6], minimize waiting time [7]. In contrast, we focus on automated classification of the Cynefin contexts to identify various degrees of uncertainty and support efficient handling of uncertainties in the future. Nonetheless, these work could provide us additional benchmarks for experimentation in the future.

7 CONCLUSION AND FUTURE WORK

We presented an industrial application of the Cynefin framework for industrial elevators developed by Orona – an industrial elevator developer in the Europe. In particular, we made the following contributions: 1) formalized the Cynefin framework as a conceptual model for cyber-physical systems, 2) defined a classification algorithm for industrial elevators, 4) evaluated it in an industrial elevator context of Orona, and 5) provided lessons learned.

In the future, we plan to implement the Cynefin framework for a set of model-based engineering methodologies to demonstrate their cost-effectiveness to develop software systems that can deal with various uncertainties. Also, in the context of elevators, we will study additional *dispatcher* algorithms, more traffic templates, *QoS* attributes, and uncertain factors.

ACKNOWLEDGMENT

We thank Orona for providing us with a *dispatcher* to validate our ideas in the Adeptness EU project (https://adeptness.eu/). The results and discussions do not, in any way, represent the quality of Orona's elevators. Also, the results are from the authors' perspective when validating their research ideas.

REFERENCES

[1] Mohammed Abdellaoui and John Denis Hey. 2008. *Advances in decision making under risk and uncertainty.* Springer.
[2] Gina Barney. 2010. *Transportation systems in buildings : CIBSE Guide D: 2010.* London: Chartered Institution of Building Services Engineers.
[3] George Burt. 2011. Towards the integration of system modelling with scenario planning to support strategy: the case of the UK energy industry. *Journal of the Operational Research Society* 62, 5 (2011), 830–839.
[4] Matteo Camilli, Carlo Bellettini, Angelo Gargantini, and Patrizia Scandurra. 2018. Online model-based testing under uncertainty. In *2018 IEEE 29th International Symposium on Software Reliability Engineering (ISSRE).* IEEE, 36–46.

[5] H William Dettmer. 2011. Systems thinking and the cynefin framework-A strategic approach to managing complex systems. *Port Angeles, WA: Goal Systems International* (2011).

[6] Joaquín Fernández, Pablo Cortés, José Guadix, and Jesús Muñuzuri. 2013. Dynamic fuzzy logic elevator group control system for energy optimization. *International Journal of Information Technology & Decision Making* 12, 03 (2013), 591–617.

[7] J Fernandez, Pablo Cortés, Jesús Muñuzuri, and José Guadix. 2013. Dynamic fuzzy logic elevator group control system with relative waiting time consideration. *IEEE transactions on industrial electronics* 61, 9 (2013), 4912–4919.

[8] Jeffrey T Fong, James J Filliben, N Alan Heckert, Dennis D Leber, Paul A Berkman, and Robert E Chapman. 2019. Uncertainty Quantification of Failure Probability and a Dynamic Risk Analysis of Decision-Making for Maintenance of Aging Infrastructure. In *Risk Based Technologies*. Springer, 65–80.

[9] Simon French. 2014. Axiomatising the Bayesian paradigm in parallel small worlds. *Bayesian Analysis* (2014).

[10] Aitor Gartziandia, Jon Ayerdi, Aitor Arrieta, Shaukat Ali, Tao Yue, Aitor Agirre, Goiuria Sagardui, and Maite Arratibel. 2021. Microservices for Continuous Deployment, Monitoring and Validation in Cyber-Physical Systems: an Industrial Case Study for Elevators Systems. In *2021 IEEE 18th International Conference on Software Architecture Companion (ICSA-C)*. IEEE, 46–53.

[11] Eva Geisberger and Manfred Broy (Eds.). [n.d.]. *Living in a Networked World: Integrated Research Agenda Cyber-Physical Systems (agendaCPS)*. Utz, München.

[12] Iwona Gorzeń-Mitka and Małgorzata Okręglicka. 2014. Improving decision making in complexity environment. *Procedia Economics and Finance* 16 (2014), 402–409.

[13] Roumiana Ilieva, Kiril Anguelov, and Mario Nikolov. 2018. A Cynefin Framework for Agile Decision Making of AI BOTS. In *2018 International Conference on High Technology for Sustainable Development (HiTech)*. IEEE, 1–4.

[14] Vladimir I Khabarov and Vladimir V Volodin. 2020. UNCERTAINTY AS THE KEY FACTOR TO SELECT PROJECT MANAGEMENT TOOLS. *PalArch's Journal of Archaeology of Egypt/Egyptology* 17, 10 (2020), 2397–2407.

[15] Hermann Kopetz. 2019. *Simplicity is Complex*. Springer.

[16] Cynthia F Kurtz and David J Snowden. 2003. The new dynamics of strategy: Sense-making in a complex and complicated world. *IBM systems journal* 42, 3 (2003), 462–483.

[17] Christian Lunghi and Francesca Baroni. 2019. Cynefin framework for evidence-informed clinical reasoning and decision-making. *J Am Osteopath Assoc* 119, 5 (2019), 312–321.

[18] Julie McLeod and Sue Childs. 2013. The Cynefin framework: A tool for analyzing qualitative data in information science? *Library & information science research* 35, 4 (2013), 299–309.

[19] Rory V O'Connor and Marion Lepmets. 2015. Exploring the use of the cynefin framework to inform software development approach decisions. In *Proceedings of the 2015 International Conference on Software and System Process*. 97–101.

[20] Franklin Selgert. 2020. Cynefin framework, devops and secure IoT. In *International Conference on Computer Safety, Reliability, and Security*. Springer, 255–265.

[21] Janne Sorsa, Harri Ehtamo, Juha-Matti Kuusinen, Mirko Ruokokoski, and Marja-Liisa Siikonen. 2018. Modeling uncertain passenger arrivals in the elevator dispatching problem with destination control. *Optimization Letters* 12, 1 (2018), 171–185.

[22] Eric K Van Beurden, Annie M Kia, Avigdor Zask, Uta Dietrich, and Lauren Rose. 2013. Making sense in a complex landscape: how the Cynefin Framework from Complex Adaptive Systems Theory can inform health promotion practice. *Health promotion international* 28, 1 (2013), 73–83.

[23] Warren Walker, Robert Lempert, and Jan Kwakkel. 2013. *Deep Uncertainty*. 395–402.

[24] Dong Mei Yan and Li Liu. 2014. Research on elevator group control uncertainty. In *Applied Mechanics and Materials*, Vol. 602. Trans Tech Publ, 874–877.

[25] Man Zhang, Shaukat Ali, Tao Yue, Roland Norgren, and Oscar Okariz. 2019. Uncertainty-wise cyber-physical system test modeling. *Software & Systems Modeling* 18, 2 (2019), 1379–1418.