Towards an ontological framework for validity frames

Reference:

Mittal Rakshit, Eslampanah Raheleh, Albertins de Lima Lucas, Vangheluwe Hans, Blouin Dominique.- Towards an ontological framework for validity frames
2023 ACM/IEEE International Conference on Model Driven Engineering Languages and Systems Companion (MODELS-C), 01-06 October 2023, Västerås, Sweden - ISBN 979-83-503-2498-3 - 2023, p. 801-805
Full text (Publisher's DOI): https://doi.org/10.1109/MODELS-C59198.2023.00128
To cite this reference: https://hdl.handle.net/10067/2025170151162165141

Towards an Ontological Framework for Validity Frames

Rakshit Mittal*, Raheleh Eslampanah*, Lucas Lima*‡, Hans Vangheluwe* and Dominique Blouin†
*Department of Computer Science, University of Antwerp - Flanders Make, Belgium.
Email: (rakshit.mittal, raheleh.eslampanah, hans.vangheluwe) @uantwerpen.be
‡Departamento de Computação, Universidade Federal Rural de Pernambuco, Recife, Brazil
Email: lucas.albertins@ufrpe.br
†Telecom Paris, Institut Polytechnique de Paris, France
Email: dominique.blouin@telecom-paris.fr

***Abstract*—A Validity Frame captures the set of contexts in which a model (and its analysis, often by means of simulation) of a system is able to replace that system with respect to questions about a set of salient properties of interest. Even though the utility of validity frames has been reported in current literature, there does not exist any precise and general definition of the concept. This paper presents our on-going development of a framework for designing and using validity frames. This framework both uses and supports model management. We have developed an ontology in order to precisely define the concepts of the model validity domain. The framework currently consists of ontological definitions integrated in a workflow model that describes a general experiment, validation experiments, and the construction of validity frames. A simple resistor model validation case-study is used as running example to describe the concepts. The validity frames of different resistor models are computed. How to use the framework in different scenarios is sketched.**

***Index Terms*—validity frame, ontology, experimental frame, validation, modeling**

I. INTRODUCTION

Validation is at the heart of the 'scientific method'. Validation activities help establish confidence in the *model*, which is an abstraction of the *system-under-study (SuS)* with respect to certain *properties-of-interest (PoIs)*. Assessing validity is done by performing validation experiments which compare these relevant PoIs between the model and the system. PoIs are obtained by analysis (often simulation) of "virtual" experiments in case of the model and "real-world" experiments in case of the SuS. If the values of the PoIs of the model are close enough to those of the SuS, the model is said to be validated and can be used as a faithful *substitute* for the system w.r.t reasoning about those PoIs.

However, this story is incomplete: a Boolean validity characteristic for a model is often not enough. For example, the Ohmic model of a resistor is not valid at high signal voltages, ambient temperature, or signal frequency. The validity of a model depends on a number of factors:

- the *experimental frame* [1], which specifies the conditions under which an experiment (real or virtual) is conducted;
- the *distance metric*, which specifies how the distance between the PoIs of the system and its model are computed;
- the *threshold*, which specifies the acceptable limit of the distance that is used to assert (in)validity.

The concept of *validity frame* was proposed to capture and account for the above-mentioned factors a model's validity [2] depends on. The validity frame can be defined as a subspace of the experimental frame-space, in which a model is valid. The existing literature has described the utility of validity frames: [3] proposes the use of validity frames in the context of computational design synthesis to prune nonsensical invalid compositions from the design space to improve efficiency and reliability of a search algorithm; [4] describes a calibration workflow which uses information of validity frames for guaranteed validity of the calibration parameters; [5] provides a framework of model-reuse based on validity frames and also describes the contents of a validity frame.

As the number of models and simulations grows, model-management problems arise in the context of validity frames:

- **P1** - the existence of multiple models for the same system: how to select the most appropriate model?
- **P2** - the evolution of requirements: how can models be re-used with guarantees of validity, when requirements change? Can results from previous validation experiments be reused?
- **P3** - automated validation: for large-scale validation, how can experimental frames be generated for all the simulation runs, and how can reasoning be automated based on the generated data?
- **P4** - run-time monitoring: how can the validity of a model be assessed during run-time and how can the model be replaced if it becomes invalid?

The realization of a description for validity frames, which also accommodates model management processes, has become increasingly crucial. The existing literature on experimental and validity frames is highly ad-hoc in nature, and there are no clear formal definitions. In this article, we present our ongoing attempt at creating a formal, ontological, process-oriented definition of validity frames in the context of a larger system-model management framework. An ontological model allows us to build unified definitions of the domain. Moreover, an ontology serves as the basis for creating a knowledge graph, which acts as a model base in the model-management context but also supports reasoning and querying over data.

Model management is inherently process-oriented. Hence it is important to record the relevant processes encountered in a validation experiment.

Thus, using an ontological-based framework allow us to properly characterize models to be selected (queried) according to desired (ontological-supported) criteria (**P1**). They can be stored and reused when the distances are properly calculated and satisfied (**P2**). In addition, the ontology-based knowledge graph allows us to reason on the stored data from previous experiments to automatically define experimental frames (**P3**). Finally, given that the processes are also formally (ontological) defined, the framework can support decision-making during their enactment (**P4**).

Section II details a resistor model validation experiment, serving as a running example throughout this discussion. The ontological definitions and associated workflows are described in Section III. We discuss the prototypical implementation, and the related technology stack in Section IV. Section V presents points of discussion and possible use-case scenarios of our work, while Section VI concludes the article.

II. RUNNING EXAMPLE

A resistor is a component commonly used in electrical circuits. There is a wide variety of resistors available; however, all resistors have the ability to decrease the flow of electric current, usually by dissipating energy in the form of heat.

Fig.1 shows on the right side two models for the resistor. Model M_0 represents a simplified resistor neglecting the effect of temperature on the resistance value where the resistance (R) remains constant at R_0. On the other hand, Model M_1 incorporates temperature-dependence. In this model, R is calculated by multiplying the reference resistance (R_0) by the factor $(1+\alpha(T-T_0))$ where α is the temperature coefficient, which quantifies the change in resistance per unit change in temperature and depends on the material of the resistor. T denotes the current temperature and T_0 represents the reference temperature.

In this running example, an experimental setup was designed to measure the current flowing through the resistor using a DC supply voltage, while simultaneously monitoring the resistor temperature using a precision infrared camera shown on the left side of Fig.1.

Assume that the resistor needs to operate in high temperatures environments. The goal of the resistor models' validation experiment is to verify their validity within the required temperature range. If both models are valid, model M_0 is preferred thanks to its lower computational overhead, else the valid model is used. However, if none of the models is found to be valid, this indicates that a more precise model is required, or the requirements need to be relaxed.

III. FRAMEWORK OF VALIDATION EXPERIMENTS

In this section, we present the concepts of our ontology contextualized in a validation experiment workflow whose purpose is to identify the experiment validity frames.

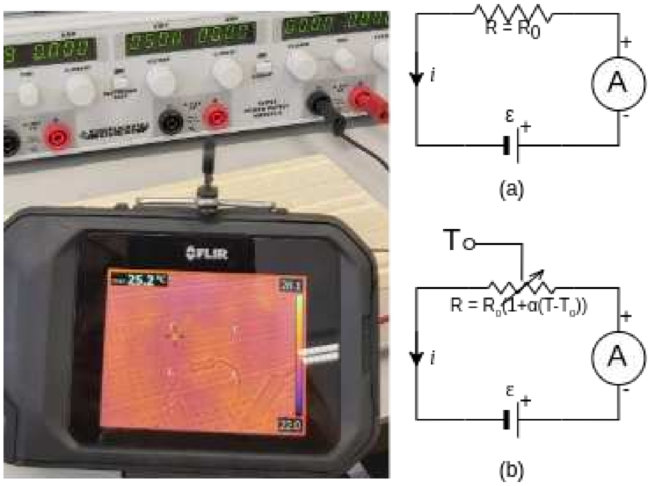

Fig. 1. Left: System-under-study and its experimental set-up; Right: (a) Circuit with Ohmic model M_0 of the resistor, (b) Circuit with temperature-dependant model M_1 of the resistor

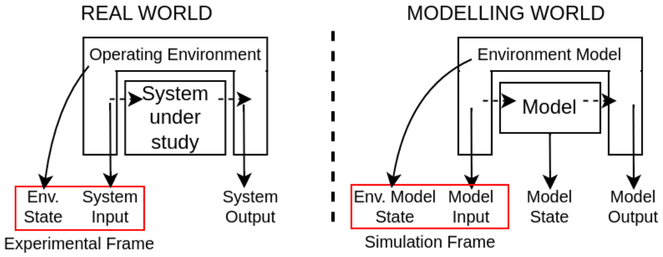

Fig. 2. The 7 VoIs from the real world and modeling world, that contribute in the definition of the corresponding experimental and simulation frames.

A. Experiments and Experimental Frames

A *frame* is loosely defined as the circumstances in which an experiment takes place [1]. A frame is specified by a number of *frame specifications*, which act as descriptors of the frame. Frame specifications include the operating environment of the system, the actors and simulators performing the experiment, the values of system and environment parameters and variables, etc.

An *experiment* is a scientific *activity* which, through *experiment observations* produces *experiment traces*. The input artifacts to an experiment include the system, the environment, and its frame. The frame of an experiment is referred to as its *experimental frame*. A *concrete frame* represents the frame of a specific experiment that has been performed.

In the running example, the experiment is 'the measurement of the resistor's resistance'. The SuS is the resistor; the environment is the circuit connected to the resistor and the experimental surroundings. A concrete frame is one which specifies environment temperature and supply voltage.

The generated experiment traces correspond to the observed values of the *Variables of Interest* (VoIs). These are quantities that are used to compute the PoI. Multiple experiment traces from several experiments may be required to compute the PoIs. Fig. 2 describes the types of VoI associated with traces in

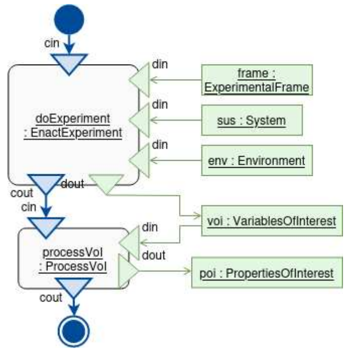

Fig. 3. The Process Model of experimentation, the *PerformExperiment* activity. Blue edges represent control flow, green edges represent data flow.

validation experiments. The VoI may be parameters (constant-valued over a single experiment) or variables (changing value during experimentation). The environment state represents the state of the environment (which may affect the behavior of the system); the system input represents specific inputs to the system-under-study; system output refers to outputs generated by the system.

In the running example, the VoIs correspond to the observed current across the resistor circuit and the DC voltage supplied by the power unit. Both of these VoIs represent the state of the environment. The temperature is another VoI that represents the state of the environment. However, in the simulation of model M_0, the temperature is not considered as a system input, whereas in the simulation of model M_1, it is taken into account. The resistance of the resistor is also a VoI and is not considered as part of the experimental or simulation frame. Instead, it is a system output that can have an impact on the environment. On the other hand, the parameters R_0, α, and T_0 are resistor model parameters.

These VoIs, in turn, are used to compute the PoI.

In the running example, the PoI is the resistance of the resistor, and it is computed as follows:

$$R(V, i) = V/i \tag{1}$$

The workflow for performing experiments is described in Fig. 3. *EnactExperiment* is an atomic activity which is the actual experimentation / simulation. Enacting this activity generates experiment traces in the form of VoI. The VoIs are then processed by the *ProcessVoI* activity that generates the final PoI that will be used to evaluate the experiment. These activities are sequentially encapsulated within the *performExperiment* activity, which will be used in a validation experiment described in the next section.

The experimenter is encouraged to provide a comprehensive specification of the experimental frame. This is because an experimental frame acts as a model of definitional uncertainty which is the practical minimum measurement uncertainty achievable in any measurement [6] stemming from the inadequate definition of obtaining the measurement itself. For example, in the resistor case study, the quality of the measuring instruments, the quality of the laboratory grounding system or influences from a strong electromagnetic field in the vicinity can impact the experiment traces. In the simulation, numerical and/or sampling approximations often impact the outcome. Hence, it is important to model the experimental frame to include all these effects, to enable reproducibility.

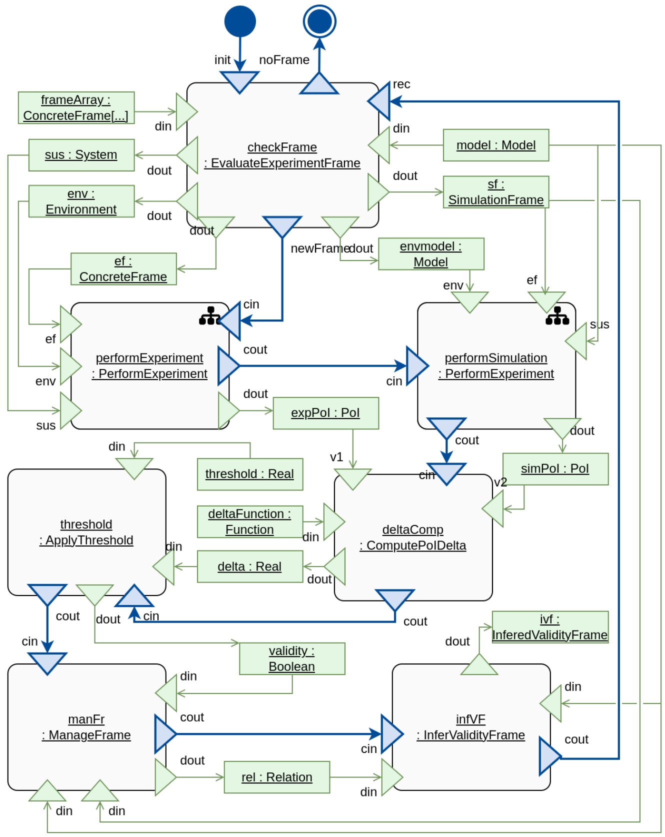

Fig. 4. The Process Model of a validation experiment. Note that activities *performExperiment* and *performSimulation* have the same type: PerformExperiment (described in Fig. 3); since a simulation is also considered an experiment in-silico.

B. Validation Experiments

A *validation experiment* comprises a real-world experiment and a matching virtual (simulation) experiment. The latter, using the system model to be validated. Since they are essentially a process consisting of many activities, we will describe validation experiments as a general workflow, which is shown in Fig. 4.

The first activity in a validation experiment is called *EvaluateExperimentFrame*, and it determines which concrete frame needs to be experimented. This activity is responsible for querying the knowledge base to check if there exist any prior concrete frame data that can be used to answer the model's validity in the requested frame.

In the running example, EvaluateExperimentFrame has inputs: the model of the resistor (M_0 or M_1), the environment model ($i = V/R$), and a set of concrete frames with varying voltages and temperatures, that should be performed. This activity, upon evaluating the provided set of concrete frames, determines whether additional experiments need to be conducted (a simple recursion rule over the set of provided concrete frames). Depending on the outcome, it either proceeds to the next stage in the control flow loop or concludes the validation experiment process. The activity also generates a *Simulation Frame* from the experimental frame, which specifies the frame of a simulation. This simulation frame is used to simulate the resistor circuit model.

Fig. 5 shows the PoI obtained by the experiments and simulations on models M_0 and M_1 in the running example.

Once the PoIs are generated, the next activity *ComputePoIDelta* in a validation experiment is the computation of the *distance*, Δ, between the generated PoIs. Note that in a validation experiment, it is necessary that the PoIs from the system-under-study and from the model correspond. The distance is computed through an explicitly specified *delta function* (*distance function*). This is a function that takes at least 2 inputs (the PoIs from the system, and from its model) and gives a non-negative output, the distance. Multiple Δs may need to be computed if there are multiple PoIs.

In the running example, the Δ function:

$$\Delta(R_{sys}, R_{model}) = |R_{sys} - R_{model}| \tag{2}$$

The next activity in a validation experiment is *ApplyThreshold*. The user specifies a threshold, $\epsilon \in \mathbb{R}^+$, as an upper limit to ensure validity of the simulation PoI w.r.t the experiment PoI. Based on these inputs, the activity generates a *Validity* Boolean output variable, which specifies whether the simulated model is valid or not (w.r.t that concrete frame). The threshold can be a relative or absolute value.

In the running example, this threshold is set to 0.1Ω. This means that if there is a difference of greater than 0.1Ω between the measured and simulated resistance, the model of the resistor is invalid in that concrete frame. Note, that this is a very naive threshold used only to demonstrate the workflow.

ApplyThreshold is followed by activities *ManageFrame* and *InferValidityFrame* related to validity frames which are described in the next section.

C. Validity Frames

A *validity frame* is defined as a collection of experimental frames in which a model is deemed a valid representation of the system it models. The *abstract validity frame* is the possibly infinite set of all experimental frames in which a model is valid. The *concrete validity frame* is the set of all concrete frames in which a model has been found to be valid in prior validation experiments. The *concrete invalidity frame* of a model is the set of all concrete frames in which a model has been found to be invalid in prior validation experiments. The current literature does not report the notion of the invalidity frame, but it is important to specify and define this, to enable more efficient and transparent experiment and model reuse.

The next activity in a validation experiment, after *ApplyThreshold*, is *ManageFrame*. This is an important activity from the model management perspective because it is responsible for using the validity Boolean output by the threshold activity to create relations (in the knowledge base), from the concrete simulation frame to:

- the model, as a concrete validity or invalidity frame,
- the distance function used to compute the Δ,
- the Δ,
- and the threshold ϵ used to assess validity.

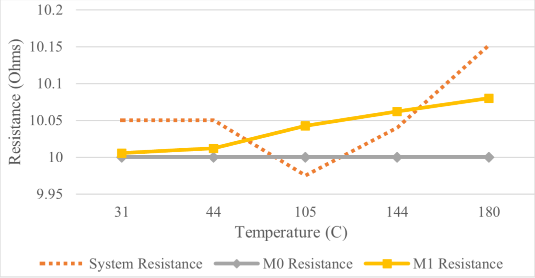

Fig. 5. The PoI (i.e., resistance) obtained from the three systems that were evaluated: the resistor, model M_0 and model M_1.

From the running example, the graph in Fig. 5 shows that the resistance of model M_1 remains within the threshold i.e., 0.1Ω in all the 5 concrete frames (different temperatures), whereas the resistance of model M_0 deviates more than 0.1Ω from that of the resistor, in the concrete frame when the temperature of the operating environment is $180°C$. Hence, the *ManageValidityFrame* activity creates 5 concrete validity frame relations with model M_1 corresponding to the 5 concrete frames, and 4 concrete validity frames and 1 concrete invalidity frame (at T = $180°C$) with model M_0.

The final step in a validation experiment control loop is *InferValidityFrame*. The activity takes as input the model, all of that model's related concrete frames, and an inferencing algorithm. Based on the information of validity and invalidity of the concrete frames, the algorithm is used to infer a set of experimental frames called an *Inferred Validity Frame*. The inferred validity frame of a model is estimated based on domain-specific knowledge and knowledge from the concrete frames of the model. Since it is impossible to create (seemingly infinite) concrete frames to assess the entire experiment frame space, it is necessary to infer a (possibly infinite) validity frame, called the inferred validity frame. The goal of a validation engineer is to infer a validity frame as close to the abstract validity frame of the model as possible. This inferred validity frame is used as an assumption of the domain of validity of the model in all analysis and applications after the validation experiment.

In the running example, the inferred validity frame is the collection of the experiment sub-space that lies in between two valid concrete frames. Hence, for model M_0, the inferred validity frame is the temperature range $31°C$ - $144°C$, whereas for model M_1, the inferred validity frame is $31°C$ - $180°C$.

IV. PROTOTYPE TECHNOLOGY STACK

We use the Ontological Modeling Language (OML) [7] to implement the vocabulary of validation experiments. A vocabulary in OML essentially defines the types and relations of a particular domain. We have used the RDF-triplestore [8] based knowledge graph technology stack to facilitate querying of objects, and their relations, within a larger model-management framework. Querying is performed with SPARQL [9], an SQL-like language, to query RDF graphs.

We model workflows associated with validation experiments using the FTG+PM++ framework [10]. The FTG+PM++ is implemented as a plugin in the `diagrams.net` diagramming environment. It is also specified as an OML vocabulary, which allows the storage of workflow models in our knowledge graph. Our framework contains an enactment engine that guides users through these workflows and stores their related data in the knowledge graph, which allows reasoning these data later on.

V. DISCUSSION

The concepts of experimental and validity frames can be used in a number of scenarios.

Traceability of Experiment Data: Storing experimental and validity frames in a standardized machine-readable format allows meaningful re-use of experiment data. Often, in industry, the experiment data is not well formalized and assumptions not recorded. It soon becomes difficult to remember the motivation for the experiment and the conditions under which it was performed. This is in particular true when the experiment is not carried out by the same person who wishes to re-use the experiment data. Experiment Replicability: A standard format for experimental frames also allows for, possibly automated, replication of experiments. Model Substitutability: Information about the validity frame allows for run- or design-time substitution of models that are concurrently valid in the frame of operation of the system. For example, the resistor model M_0 can substitute for M_1 in lower temperature conditions. It will give equally correct results, but it requires fewer computational resources (such as time or memory). Substitution can also be performed adaptively, at simulation run-time. Such *adaptive abstraction* [11] may can be crucial in settings with limited computational resources. Consistent twinning: During run-time of a digital shadow or twin, a component within the system may be operated outside its validity frame. To detect and diagnose such situations, which leads to deviations between the twin and the real system, the validity frame of each component is required. If this monitoring of the validity of a model reveals a discrepancy, a different model with an appropriate (broader) validity frame, may have to be used. This ensures that the behavior of the digital shadow / twin remains consistent with that of the system (with respect to PoIs). Pruning the Design-Space: In design space exploration, simulating a candidate design may reveal that component models are used outside their validity frames. This implies that the simulation results may be incorrect w.r.t. reality. During exploration, such design candidates must hence not be retained. This pruning of the search space may drastically reduce it [2]. = The source code related to this project, including the entire OML vocabulary, as well as reference manuals are available at: https://gitlab.telecom-paris.fr/mbe-tools/vafl/

VI. CONCLUSION

In this paper, we presented a work-in-progress general framework for assessing and representing the validity (i.e., closeness to real-world truth) of models. The framework is supported by an ontological definition, which, when implemented as a knowledge graph, serves as a model-base enabling model management. The framework is also based on a workflow model since a validation experiment is essentially a collection of human and/or computer-based activities with dependencies between them imposing a partial ordering. Next comes the implementation of domain-specific use-cases as discussed in section V, with experimental and validity frames, to demonstrate the feasibility and usefulness of our framework.

ACKNOWLEDGEMENT

The authors acknowledge Rhys Goldstein, Autodesk Research Toronto, for making the distinction between abstract and concrete validity frames. Thanks also to Joeri Exelmans and Arkadiusz Rys, University of Antwerp.

REFERENCES

[1] Zeigler, B. P. *Multifaceted Modelling and Discrete Event Simulation*. London, Academic Press. 1984.

[2] Simon Van Mierlo et al. “Exploring Validity Frames in Practice”. In: *Systems Modelling and Management*. Springer, 2020, pp. 131–148.

[3] Bert Van Acker et al. “Validity Frame Driven Computational Design Synthesis for Complex Cyber-Physical Systems”. In: *Systems Modelling and Management*. Springer International Publishing, 2020.

[4] Joachim Denil et al. “The Experiment Model and Validity Frame in M&S”. In: *Proceedings of the Symposium on Theory of Modeling & Simulation*. 2017.

[5] Bert Van Acker et al. “Valid (Re-)Use of Models-of-the-Physics in Cyber-Physical Systems Using Validity Frames”. In: *2019 Spring Simulation Conference (SpringSim)*. 2019, pp. 1–12.

[6] International Bureau of Weights and Measures. “VIM 3: International Vocabulary of Metrology”. In: 2008.

[7] D. A. Wagner et al. “Ontological Metamodeling and Analysis Using openCAESAR”. In: *Handbook of Model-Based Systems Engineering*. 2020.

[8] Brian McBride. “The Resource Description Framework (RDF) and its Vocabulary Description Language RDFS”. In: *Handbook on Ontologies*. 2004, pp. 51–65.

[9] Jorge Pérez, Marcelo Arenas, and Claudio Gutierrez. “Semantics and Complexity of SPARQL”. In: *ACM Trans. Database Syst.* 34.3 (Sept. 2009).

[10] Randy Paredis, Joeri Exelmans, and Hans Vangheluwe. “Multi-Paradigm Modelling For Model Based Systems Engineering: Extending The FTG + PM”. In: *Annual Modeling and Simulation Conference (ANNSIM)*. 2022.

[11] Romain Franceschini, Simon Van Mierlo, and Hans Vangheluwe. “Towards Adaptive Abstraction in Agent Based Simulation”. In: *Proceedings of the Winter Simulation Conference (WSC)*. IEEE, 2019, pp. 2725–2736.