A Methodology for Modeling Interoperability of Smart Sensors in Smart Grids

Eugene Y. Song, *Member, IEEE*, Gerald J. FitzPatrick, *Member, IEEE*, Kang B. Lee, *Life Fellow, IEEE*, and Edward Griffor

***Abstract*—Smart sensors in smart grids provide real-time data and status of bidirectional flows of energy for monitoring, protection, and control of grid operations to improve reliability and resilience. Smart sensor data interoperability is a major challenge for smart grids. This paper proposes a methodology for modeling interoperability of smart sensors in terms of interactions using labeled transition systems and finite state processes in order to quantitatively and automatically measure and assess the interoperability, identify and resolve interoperability issues, and improve interoperability. A generic interoperability model of synchronous message passing from a sender to a receiver is built based on the proposed methodology. A case study is provided to apply this methodology for modeling interoperability between the Institute of Electrical and Electronics Engineers C37.118 phasor measurement unit-based smart sensors and phasor data concentrators. The interoperability model can be used for the quantitative and automated measurement and assessment of the interoperability of phasor measurement unit-based smart sensors and phasor data concentrators to address interoperability issues. This methodology can also be applied to modeling interoperability of smart sensors based on other standard communication protocols in order to achieve and assure sensor data interoperability in smart grids.**

***Index Terms*—Finite state machine, finite state process, interaction, interoperability, labeled transition system, measurement and assessment, phasor measurement unit, smart grid, smart sensor.**

I. Introduction

THE SMART grid (SG) is a modernized grid that enables bidirectional flows of energy and uses two-way communication and control capabilities that will lead to an array of new functionalities and applications [1]. A key challenge is that the overall smart grid system is lacking widely accepted standards. This situation prevents easy integration of advanced applications, smart meters, smart sensors, devices, and renewable energy resources and so limits interoperability among them [2]. A smart sensor consists of a sensing module, a processing module, and a network communication module. Smart sensors have some intelligent capabilities, including intelligent data processing algorithms, time synchronization, network communication, self-identification, and self-description [3], [4]. Smart sensors play a critical role in real-time monitoring, protection, and control of electrical power generation, transmission, and distribution to customers. Sensor requirements in SGs may include network communications via standardized communication protocols [3]–[5].

The smart sensors deployed on electrical power grids are produced by a variety of vendors that employ, in distinct ways, standardized and proprietary interfaces and communication protocols. The distinct ways of implementations of these interfaces and protocols may result in the interoperability issues discussed in [6]–[8]. Some reasons for these interoperability issues of smart sensors are described in detail in [3], [4], [6], and [7]. Through smart sensor interoperability testing [6], [7], a number of challenges on the quantitative and automated measurement and assessment (M&A) interoperability of smart sensors based on huge amount of data collected during testing were encountered because manual M&A methods are laborious, time-consuming, error-prone, and costly. Hence, interoperability testing [6], [7], certification [9], and automated M&A for smart sensors are key to identifying interoperability issues, and improving, achieving, and assuring interoperability.

Interoperability issues have arisen due to a "lack of a measure of interoperability" [10]–[12]. Most of these interoperability M&A methods have not used mathematical approaches [11]. Mathematical representation and formal modeling are thus critical to achieving quantitative and automated M&A methods for identifying interoperability issues and improving interoperability. Measurement, in this context, encompasses both logical (data) and physical (analog) measurements. It is defined by assigning values (both numeric and logical) to properties of objects and events that occur during interactions. Measurement is not limited to grid-relevant physical quantities, but also includes tests that measure abstractions (non-physical quantities) like intelligence [13], [14]. The quantitative and automated M&A methods can be used to help identify interoperability problems and improve the interoperability of SG devices and systems.

This paper proposes a methodology for modeling interoperability of smart sensors in terms of interactions using labeled transition systems (LTSs) and finite state processes (FSPs) in order to quantitatively and automatically measure and assess the interoperability of smart sensors,

Manuscript received February 16, 2021; revised June 30, 2021 and September 20, 2021; accepted October 9, 2021. Date of publication November 2, 2021; date of current version December 23, 2021. Paper no. TSG-00243-2021. *(Corresponding author: Eugene Y. Song.)*

Eugene Y. Song and Edward Griffor are with the Smart Grid and Cyber-Physical Systems Program Office, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: eugene.song@nist.gov; edward.griffor@nist.gov).

Gerald J. FitzPatrick and Kang B. Lee are with the Quantum Measurement Division, National Institute of Standards and Technology, Gaithersburg, MD 20899 USA (e-mail: gerald.fitzpatrick@nist.gov; kang.lee@nist.gov).

Digital Object Identifier 10.1109/TSG.2021.3124490

identify interoperability issues, and improve sensor data interoperability. The interaction to pass messages is a concurrent action sending messages from the sender and receiving the messages by the receiver. This method defines evidence for information exchange and usage. Information usage relates to the functions performed by the device or system that has received the information. As an example, this paper applies the methodology for information exchanges and uses of the Institute of Electrical and Electronics Engineers (IEEE) C37.118 and C37.247 standards-based phasor measurement unit (PMU)-based smart sensors and phasor data concentrators (PDC) in terms of state changes of a state machine model to address the real interoperability issues between PMUs and PDCs [8]. This model can be used to automate interoperability assessment and the same methodology can be applied to model interoperability of smart sensors based on a variety of standard communication protocols. The model presented here includes both internal and external interactive behaviors and actions. It is of greatest use to developers and vendors who would have direct visibility into the internal behaviors of the applications and devices they supply to SGs. For standards conformity and interoperability test laboratories and end-users, only the external behaviors are visible, i.e., messaging between devices, and the internal behaviors can only be assumed.

The paper is organized as follows. Related work is described in Section II. The foundations of modeling interoperability are described in Section III. The interoperability of smart sensors is described in Section IV. Section V addresses a methodology for modeling interoperability of smart sensors, and Section VI illustrates a case study of applying this methodology for modeling interoperability for the IEEE C37.118 PMU-based smart sensors. The conclusion is provided in Section VII.

II. Related Work

Some interoperability M&A methods have been reviewed and analyzed [11], [12], [15], and [16]. Most of these M&A methods were not using mathematical methods [11]. The interoperability score (i-Score) is based on both operational threads and interoperability spins that indicate the quality of the pair's interoperation [17]. The i-Score is an object function to represent and maximize a summation of spins between system pairs along the operational threads. Nayebpour introduced the interoperability index model that accounts for direct interfaces among different systems and assigns a weight to each of them. This index made the i-Score much simpler and more applicable [18]. Ford proposed a general method of measuring interoperability and describing its impact on operational effectiveness [19]. Petri Nets were used to represent the interoperation relationship between message systems. The quality for interoperation is assigned by an incident matrix of system interoperation to measure the interoperability of a heterogeneous set of integrated systems [20]. Finite state machine (FSM)-based interoperability testing methods were proposed, and interoperability test sequences could be generated, though they were not used for interoperability M&A. The behaviors of components of a system were modeled by state machines that can be translated into LTS in [21], [22]. The

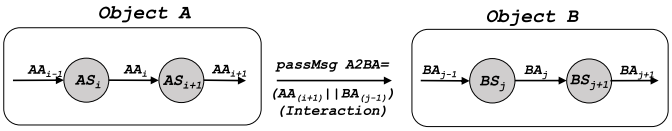

Fig. 1. Internal actions between states in an object and external interaction between states in two different objects.

interoperability of components is analyzed based on LTS against the interoperability specification defined [23]. This approach was only applied to interactions among components in a system. The interaction protocols of applications can be modeled using ontology-based FSPs to describe and analyze the interactive behavior of two processes. This approach was used to assess behavior matching of two processes [24]. The majority of interoperability assessment approaches is manual-conducted, which is a laborious and time-consuming process and in many times depends on the "subjective" knowledge of experts [16]. Leal *et al.* proposed an ontology-based interoperability assessment for a networked enterprise, which was supported by a semi-automated assessment tool to improve the assessment process efficiency by reducing the time [25]. An interoperability M&A method must be based on concept, definition, and standard definitions that relate to interoperability issues [11], [25].

An FSM comprises a finite number of states and transitions between those states. An interaction comprises actions between two objects that concurrently affect changes in the state of each. A definition of interoperability of smart sensors for information exchange and use in the smart grid is provided. An interaction between two objects is the notion of interoperability between two objects treated in this paper. LTS is used to describe the behavior of concurrent systems. FSP is a convenient formalism for specifying concurrent systems [24] and is used here to analyze and reason about interactive behaviors [27]. This paper focuses on a formal model of interoperability of smart sensors based on the notation of interactions, LTS, and FSP in order to quantitatively and automatically measure and assess the interoperability of smart sensors.

III. Foundation of Modeling Interoperability

A. Basic Concepts

- *Objects:* Entities that can observe the interactions with other entities. Objects are denoted by $O = \{O_0, O_1, \ldots, O_n\}$. Objects, the solid boxes shown in Fig. 1, can send and/or receive messages to/from each other and are communication protocol entities (e.g., sender and receiver).
- *States:* System status or conditions, which are denoted by $S = \{S_0, S_1, \ldots, S_n\}$. As shown in Fig. 1, each state is represented graphically as a solid circle. A state could be represented as $S_i = S(inAct_i, stateVar_i, outAct_i)$. States describe the status or information of a system. Each state has a set of state variables ($stateVar_i$), input variables ($inVar_i$) of input actions ($inAct_i$), and output variables ($outVar_i$) of the output actions ($outAct_i$). State

variables may include both physical variables ($pVar_i$) and logical variables ($lVar_i$). The physical variables are represented as $pVar_i = \{pVar_{i0}, pVar_{i1}, \ldots, pVar_{im}\}$, where $pVar_{ik}$ includes a set of physical state variables, for $k = 0, 1, \ldots, m$. A set of physical variables may include time, physical measurements, or others. The logical variables are represented as $lVar_i = \{lVar_{i0}, lVar_{i1}, \ldots, lVar_{in}\}$, where $lVar_i$ includes a set of logical variables, for $j = 0, 1, \ldots, n$. Logical variables may include messages (information) or the propositions (true or false) of input actions related to message conformity to the specific standard. These propositions' values are used to trigger transitions from one state to another. Each state also has a set of state functions that convert input variables to output variables. Each state may have multiple input actions and output actions.

- *Messages:* Information or data exchanged (sent and/or received) between two states in an object or different objects. Messages are denoted by $Msg = \{Msg_0, Msg_1, \ldots, Msg_n\}$. Message formats could be binary, eXtensible markup language (XML), or other formats. Messages may include information, such as encoded sensor metadata and data (physical measurements).
- *Actions:* Messaging events, such as sending a message and receiving a message. Actions are denoted by $A = \{A_0, A_1, \ldots, A_n\}$. Actions are used here as transition labels. Actions could be either internal events between two states of an object or external events (interactions) between two states in two different objects. As shown in Fig.1, an action could be defined as A = *(fromState, msg, toState),* which is a message action from the state *fromState* to the state *toState*. For example, the action AA_i between state AS_i and AS_{i+1} is an internal action in object A. The action *passMsg*, an external action shown in Fig. 1, is an interaction between object A and object B that concurrently connects the sending message action $\boldsymbol{AA_{i+1}}$ in object A and the receiving message action $\boldsymbol{BA_{j-1}}$ in object B. This interaction *passMsg* from object A to object B can be represented as $\boldsymbol{A2BA} = (\boldsymbol{AA_{i+1}}||\boldsymbol{BA_{j-1}})$. The visible actions and interactions (e.g., $\boldsymbol{AA_{i+1}}$, $\boldsymbol{BA_{j-1}}$, and $\boldsymbol{A2BA}$) are represented here in **boldface font**. The notation $\boldsymbol{AA_{i+1}}||\boldsymbol{BA_{j-1}}$ means that actions $\boldsymbol{AA_{i+1}}$ and $\boldsymbol{BA_{j-1}}$ occur concurrently or simultaneously. The interaction may be either synchronous or asynchronous. The results of messaging actions will vary with the functions of systems involved and can be evaluated based on the operational semantics of those expressions based on the communication processes and protocols [28]. Messaging actions trigger transitions from one state to another.
- *Transitions*: Relations between states induced by actions. A system transition from one state to another based on the conditions. Transitions are denoted by "→" or $T = \{T_0, T_1, \ldots, T_n\}$, where each T_i is a transition, for $i = 0, 1, \ldots, n$. Each transition is associated with an action between two states in an object. Thus, a transition can be represented as $T_i = (S_{i-1}, A_i, S_i)$. This is a transition from the state S_{i-1} to the state S_i with the action A_i.

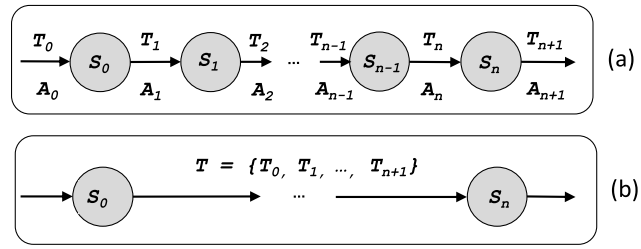

Fig. 2. Label transition system and finite state process.

- *Process:* A sequence of transitions with actions between states. A process is denoted by as $P = \{P_0, P_1, \ldots, P_n\}$, where each P_i could be a transition between two states or multiple transitions among states in an object. Operators for composing processes, depending on the related interactions, include sequential ("•"), alternative ("+"), and concurrent ("||") compositions. The sequential operator P_1 P_2 means that an action of process P_1 occurs and then an action of process P_2 occurs once the action of P_1 terminates. The alternative operator $P_1 + P_2$ means that either an action of P_1 or P_2 occurs. Finally, the concurrency operator $P_1 || P_2$ means that actions of processes P_1 and P_2 are performed simultaneously [29]. Concurrent interaction may be either synchronous or asynchronous interactions between two objects or processes.
- *Trace:* A sequence of actions [30], which are visible or observable. The trace is typically denoted by *trace* $(\rho) = A_0 A_1 \ldots A_n$.

B. Labeled Transition System

A transition system shown in Fig. 2 is a graph where circles are states and edges represent transitions between the states. An LTS consists of a set of transitions between states, labeled with actions associated with those states. An LTS can capture the interactive behaviors of a concurrent system and is a basis for describing the interactive behavior of processes. An LTS can be represented as a tuple $< S, A, T, I, AP, LF >$, where [31], [32]:

- S is a set of states.
- A is a set of actions.
- T is a set of transitions (→).
- I is a set of initial states.
- AP is a set of atomic propositions (AP) (true or false). The propositions include the results of message actions and message conformity to a specific standard specification.
- LF is the labeling function (LF) that assigns labels to transitions.

In Fig. 2, the state space of this LTS is $S = \{S_0, S_1, \ldots, S_n\}$. The set of actions is $A = \{A_0, A_1, \ldots, A_{n-1}\}$.The set of transitions is $T = \{T_0, T_1, \ldots, T_{n-1}\}$. The set of the initial states is $I = \{S_0\}$. For now, no atomic propositions are defined, so AP = Ø. An LTS is finite if S, A, and AP are finite.

C. Finite State Process

An LTS showed in Fig. 2(a) is a sequence of transitions (T) with actions of an object. Action $\boldsymbol{A_0}$ (receiving a message), action $\boldsymbol{A_{n+1}}$ (sending a message). An FSP, an algebraic

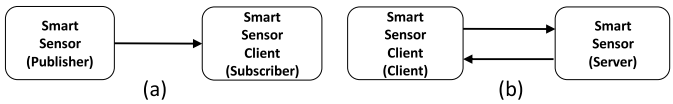

Fig. 3. Unidirectional and bidirectional communications.

representation of LTS, is a formalism for specifying concurrent components and for analyzing and reasoning about their behaviors. FSP also uses LTS to automate the reasoning and analysis of interaction protocols [27], specified as finite processes. An FSP P can be represented as: $P = T_0T_1 \ldots T_{n-1}T_n = (S_0, \boldsymbol{A_0}, S_0)(S_0, A_1, S_1) \ldots (S_{n-1}, A_n, S_n)(S_n, \boldsymbol{A_{n+1}}, S_n)$. A path is a finite sequence of states. A finite execution fragment of an LTS is a sequence of state transitions. A finite execution process shown in Fig. 2(b) is a finite execution fragment ρ of FSP, which is represented as [32]:

$$\begin{aligned}\rho(P) &= execute(P)\\ &= execute((S_0, \boldsymbol{A_0}, S_0)(S_0, A_1, S_1)\ldots(S_{n-1}, A_n, S_n)\\ &\quad (S_n, \boldsymbol{A_{n+1}}, S_n)) = (S_0\boldsymbol{A_0}A_1A_2\ldots A_n\boldsymbol{A_{n+1}}S_n) = (S_0A\ S_n)\end{aligned}$$

Where the

$$\begin{aligned}trace\ (\rho(P)) &= A = \boldsymbol{A_0}A_1A_2\ldots A_n\boldsymbol{A_{n+1}}\\ &= \boldsymbol{A_0}\tau\tau\ldots\tau\boldsymbol{A_{n+1}} = \mathbf{A}_0\mathbf{A}_{n+1}.\end{aligned}$$

This trace is a sequence of visible actions in an FSP.

IV. Interoperability of Smart Sensors

A. Smart Sensors

A smart sensor consists of a sensing module, a processing module, and a network communication module [3], [4]. The sensing module includes a set of sensors. The network communication module handles network communications through a standard communication protocol. The processing module includes four submodules: timing & synchronization, signal processing, data processing, and metadata submodules. Smart sensor has some intelligent capabilities, including intelligent data processing algorithms, time synchronization, network communication, self-identification, and self-description. Smart sensors in SGs may communicate with their clients using standardized communication protocols, such as the IEEE C37.118.2 synchrophasor data transfer for power system [33], IEEE 1815 distributed network protocol (DNP3) [34], International Electrotechnical Commission (IEC) 61850-9-2 sampled values [35], IEEE 1451.0 common functions and communication protocols standard for sensors and actuators [36], and others. Smart sensors may use different network communication models including client-server and publish-subscribe communication models. The IEC 61850-9-2 standard adopts the publish-subscribe communication model, whereas IEEE C37.118.2 and IEEE 1815 DNP3 use the client-server communication model. Based on these network communication models, the interoperability of smart sensors shown in Fig. 3 may be divided into two types: unidirectional (e.g., publish-subscribe) and bidirectional (e.g., client-server) interoperability [19]. Fig. 3(a) shows unidirectional communications for smart sensors and Fig. 3(b) shows bidirectional communication for smart sensors.

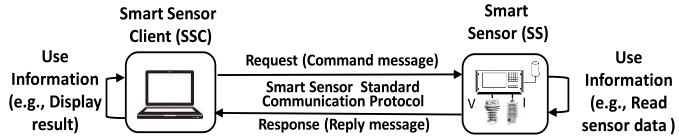

Fig. 4. Example of interoperability of smart sensors.

B. Interoperability of Smart Sensors

Based on technical, informational, and organizational groups, interoperability can be divided into eight categories, including basic connectivity, network interoperability, syntactic interoperability, semantic understanding, business context, business procedures, business objectives, and economic/regulatory policy [37]. In addition, the dimensions (cross cutting) of interoperability can be made explicit using the engineering aspects and concerns of the National Institute of Standards and Technology (NIST) Framework for Cyber-Physical Systems (CPS), including the functional, business, timing, and communications aspects as well as the trustworthiness aspect that comprises the safety, security, privacy, resilience and reliability concerns [38]. In the NIST Smart Grid Framework and Roadmap for Interoperability Standards [1], the definition of interoperability is similar to the IEEE definition [39]: "Interoperability is the capability of two or more networks, systems, devices, applications, or components to exchange and readily use information securely, effectively, and with little or no inconvenience to the user" [1]. An example is shown in Fig. 3 where the interoperability of a smart sensor and its clients is defined as *the ability of a smart sensor and its clients to exchange (send and/or receive) information and to use the information that has been exchanged.* Interoperability of smart sensors here focuses on network, syntactic and semantic interoperability categories. The interoperability of smart sensors is related to the NIST CPS Framework's data, communications and functional aspects, including the connectivity concern in the communications aspect [38]. Evidence or data for these interoperability categories here is based on different smart sensor communication standard protocols. Hence, interoperability of smart sensors can be assessed using 1) the syntactic formats or semantics of the information exchanged based on the standardized protocol, and 2) achieving the functions or goals of the standard protocol of the smart sensors as evidenced by observation of the actions and interactions of both client and sensors. These can be used to build measurement metrics of interoperability. Both checking message conformity to a specific standard protocol, and checking the achievement of the functions of the standard involved are outside of the scope of this paper. This paper focuses on the exchange of information and communication process modeling, based on the standard protocols, between smart sensors and their clients.

V. A Methodology for Modeling Interoperability of Smart Sensors

A. Network Communication Process

An information flow model of a communication process focused on information exchange between the sender and

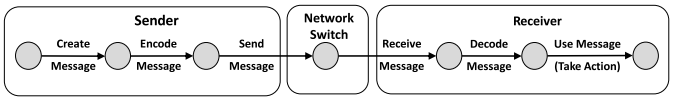

Fig. 5. Network communication process to pass a message from the sender to the receiver.

receiver [40]. Fig. 5 shows the network communication process that includes a series of messaging events (transitions) between states (represented as solid circles or nodes). A sender creates a message, encodes the message, and then sends the message to a network switch. A receiver receives the message from the network switch, decodes it, and then uses the message for applications.

B. A Methodology for Modeling Interoperability of Smart Sensors

The exchange of information between a smart sensor and its client is a real-time interactive communication process via a standardized protocol. The interactive behaviors between the sender and the receiver are modeled based on the interaction using LTS, and FSP. As shown in Fig. 1, the interaction *passMsg* ($\boldsymbol{S2RA} = \boldsymbol{A2BA} = (\boldsymbol{AA_{i+1}}||\boldsymbol{BA_{j-1}})$) to pass messages from the sender (object A) to the receiver (object B) is used to model interoperability between them. The interaction to pass messages here is a concurrent action that comprises the action of sending messages by the sender and the action of receiving the messages by the receiver. An interaction could have some attributes related to the concurrence of the processes they occur in, such as shared message, time, and space [41]. The interaction of passing a message may be synchronous, or asynchronous. The synchronous message passing process performs actions at the same time and simultaneous execution steps sometimes referred to as lock-step, such as synchronous client-server communication. In synchronous message passing, the interaction must be executed at the same time by two processes that participate in the shared action [29], which means that the interaction can share the message, time, and space. However, an asynchronous message passing process does not perform actions at the same time. In asynchronous message passing, an interaction is decoupled [41], i.e., publishers know neither subscribers' identities nor whether any subscribers with matching interests exist at all, the interaction to pass a message may share the message and space (address, channel, topic, or message queue), but not share time.

Fig. 5 shows a synchronous message passing process from the sender to the receiver via a network switch with a sender timeout. In the message passing process, there are three objects: sender, receiver, and switch. Each of these objects could be modeled as an FSM. To simplify this discussion here, it focuses on the synchronous message passing between the sender and receiver and the sender's and receiver's FSMs only.

Fig. 6 shows a unidirectional synchronous communication from the sender to the receiver. As shown in Fig. 6(a), the sender FSM consists of three states and seven message actions. And the receiver FSM consists of three states and four message actions. Each state has a message value, for example, the

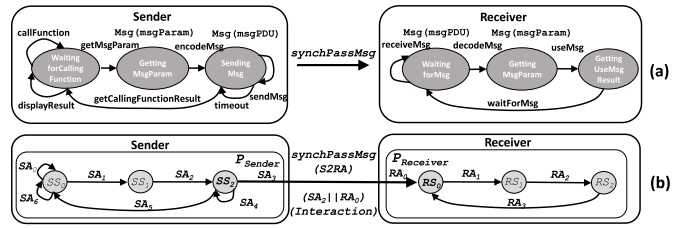

Fig. 6. Interoperability of smart sensor synchronous message passing based on interaction using LTS and FSP.

message value of the state SendingMsg is the message protocol data unit (msgPDU). Both the sender and receiver can also be modeled as an LTS. Fig. 6(b) shows the modeling of interoperability (information exchange or message passing process) between the sender and receiver using LTS and LSP. The state-space of the sender LTS is $SS = \{SS_0, SS_1, SS_2\}$. The set of message actions is $SA = \{\boldsymbol{SA_0}, SA_1, SA_2, \boldsymbol{SA_3}, SA_4, SA_5, \boldsymbol{SA}_6\}$. The set of initial states is $I = \{SS_0\}$. The state-space of the receiver LTS is $RS = \{RS_0, RS_1, RS_2\}$. The set of message actions is $RA = \{\boldsymbol{RA_0}, RA_1, RA_2, RA_3\}$. The set of initial states is $I = \{RS_0\}$. Both *inAct* and *inVar(msg)* conformity propositions of each state of the sender and receiver are defined, so $AP = \{inActs, inVars(msg)\}$. The values of these propositions of *inActs* and *inVars* are used to trigger transitions from one state to another. For example, when the value of *inVar(msgPDU)* of the *inAct* ($encodeMsg(SA_2)$) is true and the value of logical variable $lVar_{ij}(msgPDU)$ of the state *sendingMsg* (SS_2) in the sender is true, then the sender will trigger the action SA_3 to send the message to the receiver.

FSP is a convenient formalism for specifying concurrent components and analyzing and reasoning about their interactive behaviors [29]. The concurrent message-passing interactions between the sender and receiver can be modeled as a whole LTS and FSP based on the interactions. As shown in Fig. 6 (b), the interaction to synchronously pass the message, $synchPassMsg(\boldsymbol{S2RA} = (\boldsymbol{SA_3}||\boldsymbol{RA_0}))$ is to synchronously connect the sending message action $\boldsymbol{SA_3}$ of the sender to the receiving message action $\boldsymbol{RA_0}$ of the receiver. Fig. 6(b) shows an FSP of a synchronous message-passing process ($P_{Sender-Receiver}$). This FSP ($P_{Sender-Receiver}$) consists of the sender FSP (P_{Sender}) and the receiver FSP ($P_{Receiver}$). Each FSP consists of a number of actions. The synchronous message-passing process from the sender to the receiver and the execute segment can be represented as follows.

$$P_{Sender} = (SS_0, \boldsymbol{SA_0}, SS_0)(SS_0, SA_1, SS_1)(SS_1, SA_2, SS_2)$$
$$(SS_2, \boldsymbol{SA_3}, SS_2)(SS_2, SA_4, SS_2)(SS_2, SA_5, SS_0)(SS_0, \boldsymbol{SA_6}, SS_0)$$

The execute segment of P_{Sender}:

$$\begin{aligned}\rho(P_{Sender}) &= execute(P_{Sender})\\ &= execute(SS_0\boldsymbol{SA_0}SA_1SA_2\boldsymbol{SA_3}SA_4SA_5\boldsymbol{SA_6}SS_0)\\ &= execute(SS_0, \boldsymbol{SA_0}\tau\tau\boldsymbol{SA_3}\tau\tau\boldsymbol{SA_6}, SS_0)\\ &= execute(SS_0, \boldsymbol{SA_0SA_3SA_6}, SS_0)\\ P_{Receiver} &= (RS_0, \boldsymbol{RA_0}, RS_0)(RS_0, RA_1, RS_1)(RS_1, RA_2, RS_2)\\ &\quad (RS_2, RA_3, RS_0)\end{aligned}$$

The execute segment of $P_{Receiver}$:

$$\begin{aligned}\rho(P_{Receiver}) &= execute(P_{Receiver})\\ &= execute(RS_0, \boldsymbol{RA_0} RA_1 RA_2 RA_3, RS_0)\\ &= execute(RS_0, \boldsymbol{RA_0} \tau\tau\tau, RS_0)\\ &= execute(RS_0, \boldsymbol{RA_0}, RS_0)\\ P_{Sender-Receiver} &= P_{Sender} || P_{Receiver}\end{aligned}$$

The execute segment of concurrent passing message process FSP ($P_{Sender-Receiver}$):

$$\begin{aligned}&\rho(P_{Sender-Receiver}) = execute(P_{Sender} || P_{Receiver})\\ &= execute((SS_0, \boldsymbol{SA_0 SA_3 SA_6}, RS_0) || (RS_0, \boldsymbol{RA}_0, RS_0))\end{aligned}$$

The trace of the execute segment:

$$\begin{aligned}&Trace(\rho(P_{Sender-Receiver})) = (\boldsymbol{SA_0 SA_3 SA_6}) || (\boldsymbol{RA_0})\\ &= \boldsymbol{SA_0}(\boldsymbol{SA_3}) || (\boldsymbol{RA_0})\boldsymbol{SA_6} = \boldsymbol{SA_0}(\boldsymbol{SA_3} || \boldsymbol{RA_0})\boldsymbol{SA_6} = \boldsymbol{SA_0\ S2RA\ SA_6}\end{aligned}$$

As shown in Fig. 6 (b), the interaction *synchPassMsg* = $\boldsymbol{S2RA} = (\boldsymbol{SA_3} || \boldsymbol{RA_0})$ is to synchronously pass the message from the sender to the receiver. The trace of synchronous message passing process from the sender to the receiver is represented as follows: $Trace(\rho(P_{Sender-Receiver})) = \boldsymbol{SA_0\ S2RA\ SA_6}$.

This trace consists of two visible actions (including $\boldsymbol{SA_0}$ and $\boldsymbol{SA_6}$) and one interaction $\boldsymbol{S2RA}$ $(\boldsymbol{SA_3} || \boldsymbol{RA_0})$ that is an interaction between the sender and receiver with timeout constraints of sending and receiving timeouts. The interaction $\boldsymbol{S2RA}$ $(\boldsymbol{CA_3} || \boldsymbol{SA_0})$ is to synchronously pass the message from the sender to the receiver with the timeout constraint (SA_4). This trace is a model of interoperability of synchronous message passing, which can be modeled to measure and assess the interoperability of the sender and receiver based on the specific communication protocol used. For example, if $\boldsymbol{SA_0}$, $\boldsymbol{SA_6}$, and the interaction $\boldsymbol{S2RA}$ $(\boldsymbol{CA_3} || \boldsymbol{SA_0})$ are executed successfully, i.e., they all are true, this means that the sender and receiver are interoperable; and if any one of $\boldsymbol{SA_0}$, $\boldsymbol{SA_6}$, or the interaction $\boldsymbol{S2RA}$ $(\boldsymbol{CA_3} || \boldsymbol{SA}_0)$ is false, the sender and receiver are not interoperable, i.e., the action *timeout* (SA_4) is true. In the next section, this method is applied to the IEEE C37.118 protocol.

VI. Case Study: A Model of Interoperability for IEEE C37.118-Based Smart Sensors

Smart sensor communication models include the client-server and publish-subscribe communication models. The client-server could be either asynchronous or synchronous, however, publish-subscribe is asynchronous only. In this case study, it is considered to take the case of smart sensor synchronous client-server communication as an example of how our formalization of interoperability may be applied.

A. IEEE C37.118 Standard Communication Protocol

IEEE 1344-1995 [42] was the first standard for synchrophasors, which was replaced by IEEE C37.118-2005 [43]. Both measurement and real-time data transfer requirements were included in IEEE C37.118-2005 that defined four messages: data frames, header frames, configuration, and command messages. It was later extended in IEEE C37.118.2-2011 [33] with a new configuration message (CFG-3) and a continuous-time quality field in the 2-byte synchronization word (SYNC). As shown in Fig. 7, a synchrophasor data collection network consists of a number of PMU-based smart sensors, a local PDC, a corporate PDC, and a regional PDC. In addition, Fig. 7 also shows the IEEE C37.118 client-server communication protocol that includes six communication functions between the local PDC and PMU-based smart sensors. The local PDC (client) sends a command to the PMU-based smart sensors, which send a response back to the local PDC that uses the response message for its functions or applications. The local PDC also forwards them to the corporate PDC. Then the corporate PDC forwards the data to the regional PDC. However, the most current commercially available PMUs are IEEE C37.118-2005 compatible. Here, we created CmdMsg and RspMsg to represent respectively any of the command and response messages as summarized in the IEEE C37.118-2005 standard for modeling purposes. As shown in Fig. 7, IEEE C37.247-2019 defines a set of PDC functions, including communication, time-alignment of data, data forwarding, system monitoring, and so on [44]. These functions may use synchrophasor data. This paper focuses on the interoperability between the PDC as a client and PMU-based smart sensors as a server. In this case, the interoperability mainly centers on the C37.118 communications between the local PDC and the PMU-based smart sensors.

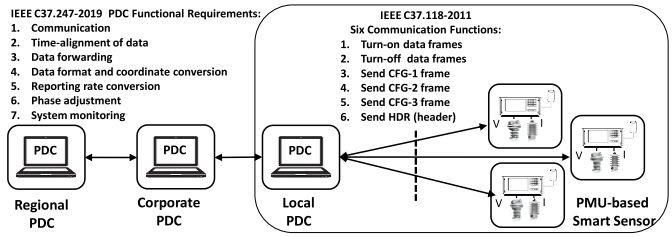

Fig. 7. Synchrophasor data collection network and IEEE C37.118 client-server communication protocol.

B. A Model of Interoperability for IEEE C37.118 PMU-Based Smart Sensor Synchronous Communication

In IEEE C37.118 PMU-based smart sensor client-server synchronous communication via a network switch, there are three objects: a PDC (client), PMU-based smart sensor (server), and network switch. Each of these objects can be modeled as an FSM. To simplify this discussion here, it focuses on the client and server shown in Fig. 8(a). These are two finite state machines of the PMU-based smart sensors and the PDC. The PDC FSM includes five states and twelve message actions. The useRspMsg action is for the PDC to use the response message for its functions defined in [44]. The PMU-based smart sensor FSM includes five states and seven message actions. Each state has a message value. For example, the message value of state the SendingCmdMsg is C37.118CmdPDU that is the C37.118 command protocol data unit (PDU). Fig. 8(b) shows a model of interoperability for PMU-based smart sensor client-server synchronous communication using LTS and FSP. The state-space of client LTS is $CS = \{CS_0, CS_1, CS_2, CS_3, CS_4\}$. The set of actions is $CA =$

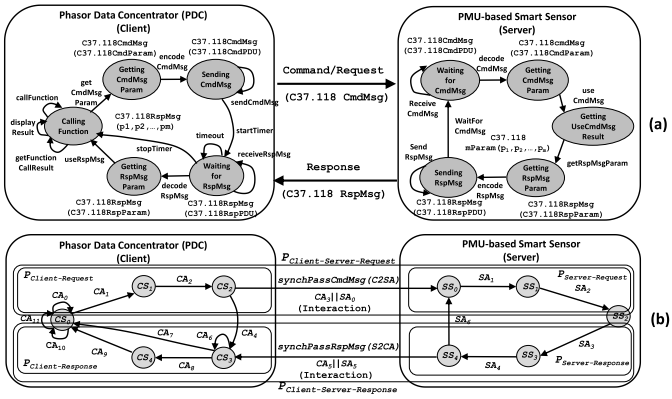

Fig. 8. A model of interoperability for IEEE C37.118 PMU-based smart sensor and PDC.

$\{\boldsymbol{CA_0}, CA_1, CA_2, \boldsymbol{CA_3}, CA_4, \boldsymbol{CA_5}, CA_6, CA_7, CA_8, CA_9, CA_{10}, \boldsymbol{CA_{11}}\}$. The set of initial states is $I = \{CS_0\}$. The state-space of PMU-based smart sensor LTS is $SS = \{SS_0, SS_1, SS_2, SS_3, SS_4\}$. The set of actions is $SA = \{\boldsymbol{SA_0}, SA_1, SA_2, SA_3, SA_4, \boldsymbol{SA_5}, SA_6\}$. The set of initial states is $I = \{SS_0\}$. For now, both *inAct* and *inVar(msg)* conformity propositions of each state are defined, so $AP = \{inActs, inVars(msg)\}$. Fig. 8 (b) shows an FSP of a PMU-based smart sensor client-server synchronous communication process ($P_{C37.118Client-Server}$) between the PDC and the PMU-based smart sensor via the IEEE C37.118 command and response messages. As shown in Fig. 8(b), two interactions, to pass command and response messages, are proposed to model the interoperability of the PDC and the PMU-based smart sensor. The first interaction *synchPassCmdMsg* ($\boldsymbol{C2SA} = (\boldsymbol{CA_3}||\boldsymbol{SA_0})$) is to synchronously pass the command message from the client to the server. The second interaction *synchPassRspMsg* ($\boldsymbol{S2CA} = (\boldsymbol{SA_5}||\boldsymbol{CA_5})$) is to synchronously pass the response message from the server back to the client. Both interactions are with the timeout constraint.

The client-server synchronous communication between a client (PDC) and smart sensor can be modeled as a single LTS and FSP based on the interactions. An IEEE C37.118 client-server FSP ($P_{C37.118-Client-Server}$) consists of a client process ($P_{C37.118-Client}$) and a server (smart sensor) process ($P_{C37.118-Server}$). The client process ($P_{C37.118-Client}$) includes a client request process ($P_{C37.118-Client-Request}$) and a client response process ($P_{C37.118-Client-Response}$). The server process ($P_{C37.118-Server}$) includes a $P_{C37.118-Server-Request-Response}$). Each component of these processes is also an FSP, which is represented formally as follows. And the execution segments of each component (FSP) are as follows, respectively. The actions in bold text are visible and others are not visible.

- Client request timeout FSP:

$$
\begin{aligned}
&P_{C37.118-Client-Request-timeout}\\
&= (CS_0, \boldsymbol{CA_0}, CS_0)(CS_0, CA_1, CS_1)(CS_1, CA_2, CS_2)\\
&\quad (CS_2, \boldsymbol{CA_3}, CS_2)(CS_2, CA_4, CS_3)\\
&\quad (CS_3, CA_6, CS_3)(CS_3, CA_7, CS_0)
\end{aligned}
$$

The execute segments of $P_{C37.118-Client-Request-timeout}$:

$$
\begin{aligned}
&\rho(P_{C37.118-Client-Request-timeout})\\
&= execute((CS_0, \boldsymbol{CA_0}CA_1CA_2\boldsymbol{CA_3}CA_4CA_6CA_7, CS_3))\\
&= execute((CS_0, \boldsymbol{CA_0}\tau\tau\boldsymbol{CA_3}\tau\tau\tau, CS_3))\\
&= execute((CS_0, \boldsymbol{CA_0CA_3}, CS_3))
\end{aligned}
$$

- Client response FSP:

$$
\begin{aligned}
&P_{C37.118-Client-Response}\\
&= (CS_3, \boldsymbol{CA_5}, CS_3)(CS_3, CA_8, CS_4)(CS_4, CA_9, CS_0)\\
&(CS_0, CA_{10}, CS_0)(CS_0, \boldsymbol{CA_{11}}, CS_0)
\end{aligned}
$$

The execute segments of $P_{C37.118-Client-Response}$:

$$
\begin{aligned}
&\rho(P_{C37.118-Client-Response})\\
&= execute((CS_3, \boldsymbol{CA_5}CA_8CA_9CA_{10}\boldsymbol{CA_{11}}, CS_0))\\
&= execute((CS_3, \boldsymbol{CA_5}\tau\tau\tau\boldsymbol{CA_{11}}, CS_0))\\
&= execute((CS_3, \boldsymbol{CA_5CA_{11}}, CS_0))
\end{aligned}
$$

- Server request-response FSP:

$$
\begin{aligned}
&P_{C37.118-Server-Request-Response}\\
&= (SS_0, \boldsymbol{SA_0}, SS_0)(SS_0, SA_1, SS_1)(SS_1, SA_2, SS_2)(SS_2, SA_3, SS_3)\\
&(SS_3, SA_4, SS_4)(SS_4, \boldsymbol{SA_5}, SS_4)(SS_4, SA_6, SS_0)
\end{aligned}
$$

The execute segments of $P_{C37.118-Server-Request-Response}$:

$$
\begin{aligned}
&\rho(P_{C37.118-Server-Request-Response})\\
&= execute(P_{C37.118-Server-Request-Response})\\
&= execute((SS_0, \boldsymbol{SA_0}SA_1SA_2SA_3SA_4\boldsymbol{SA_5}SA_6, SS_4))\\
&= execute((SS_0, \boldsymbol{SA_0}\tau\tau\tau\tau\boldsymbol{SA_5}\tau, SS_4))\\
&= execute((SS_0, \boldsymbol{SA_0SA_5}, SS_4))
\end{aligned}
$$

An FSP of PMU-based smart sensor client-server synchronous communication process ($P_{C37.118-Client-Server}$) can be represented as follows, based on both the interaction (***C2SA***) from the client to the server and the interaction (***S2CA***) from the server to the client.

$$
\begin{aligned}
&P_{C37.118-Client-Server} = (P_{C37.118-Client-Request-timeout}\\
&\quad || \; P_{C37.118-Server-Request-Response} \; || \; P_{C37.118-Client-Response})
\end{aligned}
$$

The execute segment of synchronous message passing process FSP ($P_{C37.118-Client-Server}$) is as follows:

$$
\begin{aligned}
&\rho(P_{C37.118-Client-Server})\\
&= execute(P_{C37.118-Client-Request-timeout}\\
&||P_{C37.118-Server-Request-Response}||P_{C37.118-Client-Response})\\
&= execute(((CS_0, \boldsymbol{CA_0CA_3}, CS_3))||(SS_0, \boldsymbol{SA_0SA_5}, SS_4)\\
&||(CS_3, \boldsymbol{CA_5CA_{11}}, CS_0))
\end{aligned}
$$

The trace of the execute segment is:

$$
\begin{aligned}
&Trace(\rho(P_{C37.118-Client-Server}))\\
&= \boldsymbol{CA_0CA_3}||\boldsymbol{SA_0SA_5}||\boldsymbol{CA_5CA_{11}}\\
&= \boldsymbol{CA_0}(\boldsymbol{CA_3}||\boldsymbol{SA_0})(\boldsymbol{SA_5}||\boldsymbol{CA_5})\boldsymbol{CA_{11}}\\
&= \boldsymbol{CA_0}(\boldsymbol{CA_3}||\boldsymbol{SA_0})(\boldsymbol{SA_5}||\boldsymbol{CA_5})\boldsymbol{CA_{11}}\\
&= \boldsymbol{CA_0\ C2SA\ S2CA\ CA_{11}}
\end{aligned}
$$

The trace of IEEE C37.118 PMU-based smart sensor client-server synchronous communication process can be represented as:

$$Trace(\rho_{C37.118-Client-Server}) = \boldsymbol{CA_0}\ \boldsymbol{C2SA}\ \boldsymbol{S2CA}\ \boldsymbol{CA_{11}}$$

The trace is a model of interoperability for IEC C37.118 PMU-based smart sensors, which consists of two visible actions (including $\boldsymbol{CA_0}$ and $\boldsymbol{CA_{11}}$) and two sequential interactions to pass the IEEE C37.118 command/response messages between the PMU-based smart sensor and its client (PDC) with a timeout constraint (CA_6). This model can be used to assess the interoperability of PMU-based smart sensors and PDCs based on the logical reasoning method. For example, if $\boldsymbol{CA_0}$, $\boldsymbol{CA_{11}}$, and both interactions ($\boldsymbol{CA_3}||\boldsymbol{SA_0}$) and ($\boldsymbol{SA_5}||\boldsymbol{CA_5}$) are executed successfully, i.e., all four are true, this means that the PMU-based smart sensor and PDC are interoperable. And if $\boldsymbol{CA_0}$, or $\boldsymbol{CA_{11}}$, ($\boldsymbol{CA_3}||\boldsymbol{SA_0}$) or ($\boldsymbol{SA_5}||\boldsymbol{CA_5}$) is false, this means that they are not interoperable, i.e., the action *timeout* (CA_6) is true.

VII. Conclusion

This paper presents a methodology for modeling the interoperability of smart sensors in terms of the notion of interactions using labeled transition systems and finite state processes. A generic interoperability model is constructed for synchronous message passing from a sender to a receiver. A case study is provided to illustrate modeling interoperability between the IEEE C37.118 phasor measurement unit-based smart sensor and the phasor data concentrator. The interoperability model presented here can be used for quantitative and automated measurement and assessment (M&A) of the interoperability of smart sensors to help identify interoperability issues and improve interoperability. This methodology can also be applied to model interoperability of other smart sensors that are based on other standardized communication protocols.

Future work will involve the study of quantitative and automated M&A methods of interoperability based on the interoperability model presented in this paper in order to identify and solve interoperability issues or problems, thus improve, achieve and assure the interoperability of smart sensors in smart grids.

Acknowledgment

The authors gratefully acknowledge the contributions of Chris Greer and Avi Gopstein for their supports.

Certain commercial equipment, instruments, or materials (or suppliers, or software) are identified in this paper to foster understanding. Such identification does not imply recommendation or endorsement by the National Institute of Standards and Technology, nor does it imply that the materials or equipment identified are necessarily the best available for the purpose.

References

[1] C. Greer *et al.*, "NIST framework and roadmap for smart grid interoperability standards, release 3.0," NIST, Gaithersburg, MD, USA, Rep. NIST SP 1108r3, 2014. [Online]. Available: https://www.nist.gov/publications/nist-framework-and-roadmap-smart-grid-interoperability-standards-release-30

[2] V. C. Güngör *et al.*, "Smart grid technologies: Communication technologies and standards," *IEEE Trans. Ind. Informat.*, vol. 7, no. 4, pp. 529–539, Nov. 2011.

[3] E. Y. Song, G. J. FitzPatrick, and K. B. Lee, "Smart sensors and standard-based interoperability in smart grids," *IEEE Sensors J.*, vol. 17, no. 23, pp. 7723–7730, Dec. 2017.

[4] E. Y. Song, G. J. FitzPatrick, K. B. Lee, A. M. Gopstein, and P. A. Boynton, "Interoperability testbed for smart sensors in smart grids," in *Proc. IEEE PES ISGT*, Washington, DC, USA, Feb. 2018, pp. 1–5.

[5] R. Moghe, F. C. Lambert, and D. Divan, "Smart "stick-on" sensors for the smart grid," *IEEE Trans. Smart Grid*, vol. 3, no. 1, pp. 241–252, Mar. 2012.

[6] E. Y. Song and G. J. FitzPatrick, "Interoperability test for IEEE C37.118 standard-based phasor measurement units (PMUs)," in *Proc. IEEE PES ISGT*, Minneapolis, MN, USA, Sep. 2016, pp. 1–5.

[7] E. Y. Song, K. B. Lee, G. J. FitzPatrick, and Y. Zhang, "Interoperability test for IEC 61850-9-2 standard-based merging units," in *Proc. IEEE PES ISGT*, Washington, DC, USA, Apr. 2017, pp. 1–6.

[8] Y. Guan, M. Kezunovic, A. Sprintson, and M. Yan, "Verifying interoperability and application performance of PDCs in synchrophasor system solution," in *Proc. North Amer. Power Symp. (NAPS)*, Champaign, IL, USA, Sep. 2012, pp. 1–6.

[9] "Smart Grid Testing and Certification." 2012. [Online]. Available: https://www.nist.gov/programs-projects/smart-grid-testing-and-certification

[10] G. E. LaVean, "Interoperability in defense communications," *IEEE Trans. Commun.*, vol. 28, no. 9, pp. 1445–1455, Sep. 1980.

[11] T. C. Ford, J. M. Colombi, S. R. Graham, and D. R. Jacques, "A Survey on Interoperability Measurement." [Online]. Available: http://www.dodccrp.org/events/12th_ICCRTS/CD/html/papers/096.pdf (Accessed: May 6, 2017).

[12] R. Rezaei, T. K. Chiew, S. P. Lee, and Z. S. Aliee, "Interoperability evaluation models: A systematic review," *Comput. Ind.*, vol. 65, no. 1, pp. 1–23, Jan. 2014.

[13] S. Khan and L. Finkelstein, "Mathematical modelling in the analysis and design of hard and soft measurement systems," *Measurement*, vol. 46, no 8, pp. 2936–2941, Oct. 2013.

[14] L. Finkelstein, "Problems of measurement in soft systems," *Measurement*, vol. 38, no. 4, pp. 267–274, Dec. 2005.

[15] M. Camara, Y. Ducq, and R. Dupas, "Methodology for prior evaluation of interoperability," in *Proc. 11th IFIP WG 5.5 Working Conf. Virtual Enterprises (PRO-VE)*, 2010, pp. 697–704. [Online]. Available: https://hal.inria.fr/hal-01055929/document

[16] G. da Silva Serapião Leal, W. Guédria, H. Panetto, "Interoperability assessment: A systematic literature review," *Comput. Ind.*, vol. 106, pp. 111–132, Apr. 2019.

[17] T. Ford, J. Colombi, S. Graham, and D. Jacques, "The Interoperability Score." 2007. [Online]. Available: https://pdfs.semanticscholar.org/7689/4d3dd0ffe0259ec90e23c3b3dd0c470fb42c.pdf

[18] M. Nayebpour, "The interoperability index model: Improving I-score model for interoperability measurement," *Int. J. Adv. Res. Eng. Appl. Sci.*, vol. 4, no. 11, pp. 2278–6252, Nov. 2015.

[19] T. C. Ford, J. M. Colombi, D. R. Jacques, and S. R. Graham, "A general method of measuring interoperability and describing its impact on operational effectiveness," *J. Defense Model. Simul. Appl. Methodol. Technol.*, vol. 6, no. 1, pp. 17–32, Jul. 2009. [Online]. Available: https://journals.sagepub.com/doi/10.1177/1548512909342079

[20] F. Guo, M. Wang, and J. Liao, "A measurement method of the interoperability of integrated systems based on process net," *Appl. Mech. Mater.*, vols. 88–89, pp. 99–103, Aug. 2011.

[21] K. El-Fakih, V. Trenkaev, N. Spitsyna, and N. Yevtushenko, "FSM based interoperability testing methods for multi stimuli model," in *Proc. IFIP Int. Conf. Test. Commun. Syst.*, 2004, pp. 60–75.

[22] Y. Guo, X. Huang, M. Di, and Y. Ma, "Research on modeling and development process for passive interoperability testing," in *Proc. IEEE 3rd Int. Conf. Cloud Comput. Intell. Syst.*, Shenzhen, China, Nov. 2014, pp. 732–737.

[23] T. Lambolais, A.-L. Courbis, H.-V. Luong, and T.-L. Phan, "Interoperability analysis of systems," *IFAC Proc.*, vol. 44, no. 1, pp. 7879–7884, Jan. 2011.

[24] A. Bennaceur, V. Issarny, R. Spalazzese, and S. Tyagi, "Achieving interoperability through Semantics-based Technologies: The instant messaging case," in *Proc. Int. Semantic Web Conf. (ISWC)*, Boston, MA, USA, Sep. 2012, pp. 17–33.

[25] G. S. S. Leal, W. Guédria, H. Panetto, and E. Proper, "Towards a semi-automated tool for interoperability assessment: An ontology-based

approach," in *Proc. Int. Conf. SPICE Softw. Process Improvement Capab. Determination*, Palmaa de Mallorca, Spain, Oct. 2017, pp. 241–254,

[26] J.-P. Jacquet and A. Arbran, "From software metrics to software measurement methods: A process model," presented at the IEEE Int. Symp. Softw. Eng. Stand., Walnut Creek, CA, USA, Jun. 1997, pp. 128–135.

[27] J. Magee and J. Kramer, *Concurrency: State Models & Java Programs*. Hoboken, NJ, USA: Wiley, 2006.

[28] M. Broy, "Semantics of communicating processes," *Inf. Control*, vol. 61, no. 3, pp. 202–246, Jun. 1984.

[29] "Chapter 3: Concurrent Execution." [Online]. Available: https://imada.sdu.dk/∼petersk/DM519/slides/slides2.pdf (Accessed: Dec. 22, 2019).

[30] J. F. Groote and M. R. Mousavi, *Modeling and Analysis of Communicating Systems*. Cambridge, MA, USA: MIT Press, Aug. 29, 2014.

[31] R. Gorrieri, *Process Algebras for Petri Nets*, Cham, Switzerland: Springer, Apr. 2017, pp. 15–34.

[32] R. B. France, "Basics of Linear Temporal Properties." [Online]. Available: https://www.cs.colostate.edu/∼france/CS517/Slides/CurrentSlides/LTP1.pdf (Accessed: Dec. 19, 2019).

[33] *Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118.2-2011, 2011.

[34] *Electric Power Systems Communications-Distributed Network Protocol (DNP3)*, IEEE Standard 1815-2012, 2012.

[35] *Communication Networks and Systems for Power Utility Automation—Part 9-2: Specific Communication Service Mapping (SCSM)—Sampled Values Over ISO/IEC 8802-3*, IEC Standard 61850-9-2:2011, 2011.

[36] *Standard for a Smart Transducer Interface for Sensors and Actuators—Common Functions, Communication Protocols, and Transducer Electronic Data Sheet (TEDS) Formats*, IEEE Standard 1451.0-2007, 2007.

[37] "GridWise Interoperability Context-Setting Framework V 1.1." 2018. [Online]. Available: https://www.gridwiseac.org/pdfs/interopframework_v1_1.pdf

[38] *Framework for Cyber-Physical Systems: Volume 1, Overview Version 1.0*, document NIST Special Publication 1500-201, NIST, Gaithersburg, MD, USA, 2017. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/SpecialPublications/NIST.SP.1500-201.pdf

[39] *Standard Glossaries of Software Engineering Terminology*, IEEE Standard 610.12-1990, 1990.

[40] S. AI-Fedaghi and A. Alsaqa, "Conceptual model for communication," *Int. J. Comput. Sci. Inf. Security*, vol. 6, no. 2, pp. 29–41, Dec. 2009.

[41] P. T. Eugster, P. A. Felber, R. Guerraoui, and A.-M. Kermarrec, "The many faces of publish/subscribe," *ACM Comput. Surv.*, vol. 35, no. 2, pp. 114–131, 2003. [Online]. Available: https://dl.acm.org/doi/pdf/10.1145/857076.857078

[42] *Standard for Synchrophasors for Power Systems*, IEEE Standard 1344-1995, Dec. 1995.

[43] *Standard for Synchrophasor Measurements for Power Systems*, IEEE Standard C37.118-2005, Mar. 2006.

[44] *Standard for Phasor Data Concentrator for Power Systems*, IEEE Standard C37.247-2019, Sep. 2019.

Eugene Y. Song (Member, IEEE) received the Ph.D. degree in manufacturing engineering from Tsinghua University, Beijing, China, in 1998. He was formerly a Full Professor with the Hebei University of Technology, Hebei, China. He is an Electronics Engineer with the Smart Grid (SG) Program and Internet of Things (IoT)/Cyber–Physical System (CPS) Program, National Institute of Standards and Technology (NIST), Gaithersburg, MD, USA. His current research interests include smart sensors, interface standards for smart sensors and sensor networks, interoperability testing and modeling, measurement and assessment methods, wide-area monitoring and control for SG, high-frequency harmonics measurement, and harmonics spectrum and distortion indicators analysis for distribution grid in the NIST SG Testbed. His research interests also include digital twins, co-simulation for IoT/CPS, and automated federate testing framework for the Universal CPS Environment for Federation federation in the NIST IoT/CPS Testbed. He is the Chair of the IEEE 1451.0 Standard Working Group, which is developing an interface standard for smart sensors and actuators, and sensor networks suitable for SG/IoT/CPS applications.

Gerald J. FitzPatrick (Member, IEEE) received the Ph.D. degree in electrical engineering from the State University of New York at Buffalo. He is the Chief of the Quantum Measurement Division and the Leader of the Wide Area Monitoring and Control and Smart Meters Projects. He oversees research in precision measurement techniques and interoperability, electric power and energy measurements, as well as measurement services for calibration of electrical artifact standards. His research interests include performance characterization, interoperability standards, development of smart sensors and meters, phasor measurement units, and merging units using the NIST Smart Grid Testbed. He is the Past Chair of the IEEE PES Power Systems Instrumentation Committee.

Kang B. Lee (Life Fellow, IEEE) received the B.E.S. degree from Johns Hopkins University and the M.S.E.E. degree from the University of Maryland. He is an Electronics Engineer who started work with the National Bureau of Standards in 1974 and later retired with the National Institute of Standards and Technology with over 40 years of U.S. Government services and has continued working as a Research Associate today. He has authored/coauthored research papers and given keynotes on the subjects of smart sensors, RFID, IoT, and PTP worldwide. His experience and research interests have been in sensor-based measurement and control, smart and wireless sensor technologies, and standards for smart manufacturing and smart grids. He has developed advanced manufacturing technology for the U.S. Navy for just-in-time manufacture of precision submarine parts based on research conducted on error-compensation techniques at the Advanced Manufacturing Research Facility. The techniques have been adopted by manufacturers to build more accurate production machines at much lower costs. He is the Chair of IEEE Instrumentation and Measurement Society's Technical Committee on Sensor Technology and TC-9. This committee has developed a suite of 15 IEEE 1451 (ISO/IEC 21451) standards on smart transducer interfaces for sensors and actuators and the IEEE 1588 (IEC 61588) precision clock synchronization protocol standard known globally as the precision time protocol suitable for IoT, IIoT, smart grid, and CPS applications. He was profiled by EE Times's Great Mind, Great Ideas project as one of 29 innovators in the world in 2006. The innovation: IEEE 1451 Empowering Smart Sensors Revolution.

Edward Griffor received the Ph.D. degree in mathematics from MIT and the Habilitation degree from the University of Oslo.

He is an Associate Director of Cyber–Physical Systems with the National Institute of Standards and Technology (NIST). He has served on the research faculties of universities in the U.S., Europe, Asia, and South America and on the EU Committee on High Performance Computing. He is an Adjunct Professor with the Center for Molecular Medicine and Genetics, Wayne State University in systems biology and the University of Grenoble in computation and artificial intelligence. At NIST, he leads foundations and applications of the NIST Framework for Cyber–Physical Systems to system trustworthiness and simulation. He and his team at NIST strive to meet industry needs for mathematical and engineering science for assurance, including applications in transportation, manufacturing, energy, infrastructure, and medicine. In industry, he led advanced efforts in software assurance and autonomous vehicles. His current research applies methods of physics, mathematics, and biology to the assurance of cyber–physical systems and Internet of Things, including the safety measurement methodologies for automated driving systems. His books include the *Handbook of System Safety and Security*, *Handbook of Computability*, and the *Mathematical Theory of Domains*. He has published extensively in journals and has been an Invited Lecturer to the American Mathematical Society, the North American Software Certification Consortium, the SAE International, and the U.S. Federal Reserve Bank. He served as a Walter P. Chrysler Technical Fellow and the Chair of the Chrysler Technology Council and the MIT Alliance. He is an NSF/NATO Postdoctoral Fellow.