RELATIONAL MODELING FRAMEWORK FOR COMPLEX SYSTEMS

Dragos Constantin POPESCU[1], Ioan DUMITRACHE[2]

This paper addresses the topic of modeling complex systems and introduces a new formal framework. It starts with an extensive overview on the context of complex systems, major research and application directions and general requirements in terms of modeling with an emphasis on the Cyber Physical Systems paradigm. Afterwards, five key modeling formalisms which address various aspects of complex systems are presented. The formal contribution of this paper is ultimately introduced. The approach is based on integrating in a unified architecture multiple modeling formalisms that are dynamically re-configured inside a modeling environment. A particular implementation of the framework is further defined for integrating relational models, probabilities and numerical information.

Keywords: complex systems, modeling formalism, relational modeling, distributed expertise, concurrent modeling

1. Introduction

In the last 30 years, due to the intense industrial and technological development, complexity started to become a more and more obvious concern [1]. Simple tools have been extensively developed becoming smart tools, new electronic devices have emerged and spread worldwide, and simple systems have become complex ones. Furthermore, with the development of large scale digital communication technologies and of the internet, a new requirement to interconnect all of these devices and systems emerged, which paved the way for new research and engineering topics as the Internet of Things (IoT) or Industrial Internet.

As a result of this context, the interest was focused towards the development of large scale integrative Systems of Systems [2] for which the main objective is to provide integrated smart services with increased efficiency, global dynamic stability and operational reliability [3]. Such services include:

[1] PhD Student, Dept. of Automatic Control and Systems Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: dragos.popescu@acse.pub.ro
[2] Prof., Dept. of Automatic Control and Systems Engineering, University POLITEHNICA of Bucharest, Romania, e-mail: ioan.dumitrache@acse.pub.ro

- **Intelligent and multi-modal transportation systems**, integrating in a unified architecture: smart and autonomous vehicles, smart roads and software platforms for optimizing the traffic and intelligent routing.
- **Smart grids**, integrating in a unified network: prosumers, distributed energy storage systems, renewable energy sources and smart energy routing capabilities.
- **Smart cities**, composed of **smart buildings** with intelligent and integrated utility services like: smart power systems, intelligent Heating Ventilation and Air Conditioning (HVAC), and advanced Building Management Systems (BMS) for increased security and efficiency.
- **Smart factories** integrated in **Smart Markets** where the manufacturing is intelligent, flexible and demand-driven and the supply chain and logistics are assisted by dedicated modeling and operating software platforms for increased efficiency.

The design of such complex contexts having a great impact all around us, that provide robustness, dynamic stability, operational reliability, safety and efficiency, demand for new theoretical and engineering tools [4]. The systems involved are characterized by spatio-temporal distribution, diverse dynamics and increased connectivity that needs to be analyzed and intimately understood, reasoned and inferred, and the development process needs to be address in a systematic manner [5]. Proper analytical tools are exponentially difficult to develop because their formal complexity has to exceed the complexity of the context involved [6].

This article addresses the topic of modeling complex systems and provides an overview of what are the requirements for such new tools (Section 2), what are the research results developed in literature for this topic (Section 3) and introduces a new conceptual modeling framework, a systematic mean of integrating existing modeling formalisms and developing new ones for complex systems (Section 4).

2. General requirements for modeling complex systems

With the development and the widespread use of processor-based electronic devices, the relay-based analog control was replaced by computers. This provided more powerful tools for managing efficiently the increasing complexity and led to the beginning of the **C2** paradigm (**C**omputer for **C**ontrol). Networking equipment was further required in order to cope with large scale systems. Consequently, the **C3** paradigm emerged (**C**omputer and **C**ommunication for **C**ontrol). The components integrated in the systems become very diverse and the dynamic stability hard to achieve. Such heterogeneous contexts required new advanced and intelligent control strategies, including fuzzy reasoning, neural network models, bio-inspired evolutionary algorithms or cognitive techniques, hence the **C4** paradigm was developed (**C**omputer, **C**ommunication and **C**ognition for **C**ontrol) [7]. As both the systems and the control architectures become more and more complex and heterogeneous, classical modeling and design formalisms began to show their limits. In response, a new paradigm emerged, the **CPS** (Cyber Physical Systems), where the development is achieved through a systematic manner based on modeling and integrating the dynamics of the environment, physical processes, interface devices, computation components, communication modules and human actors involved. Such systems include a high number of distributed modules that are strongly interacting with the environment and with each other [8]. A conceptual example of a CPS is depicted in Fig. 1.

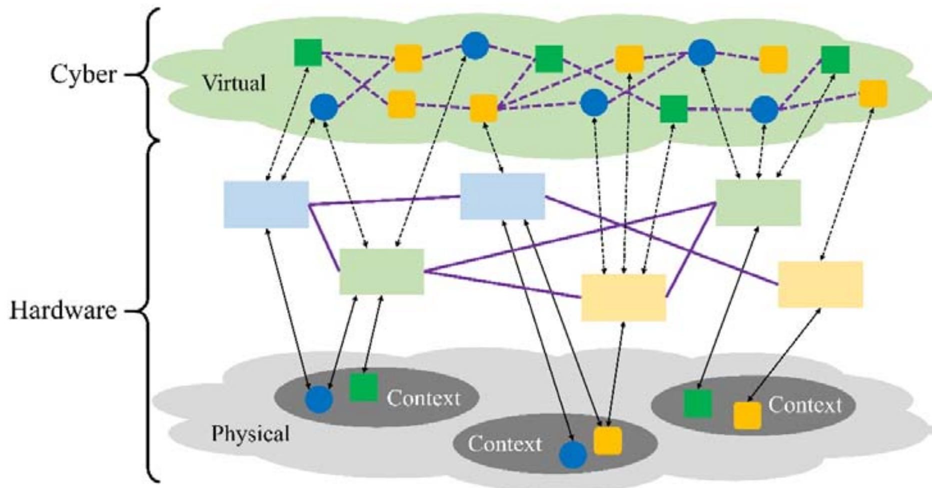

Fig. 1. Conceptual diagram of Cyber - Physical Systems architecture.

There are two corresponding environments, a physical and a virtual one. The physical environment is comprised of multiple contexts which are linked in various workflows. Each element has an equivalent representation with some degree of abstraction in the Cyber Infrastructure. The Hardware Infrastructure is comprised of field interface components (sensors and actuators), local embedded modules and dedicated network equipment which belong to the Data Infrastructure. Each CPS is comprised of three fundamental components. The **computation** is provided by local embedded modules, dedicated devices (like smartphones, computers or industrial equipment) or inside the virtual environment through various procedural, Machine Learning or optimization algorithms. The **communication** is both physical (using dedicated hardware equipment and infrastructures for gathering and sending data between the physical and the virtual environments) and virtual (between the abstract components, models and software modules inside the Cyber Infrastructure). The **control** is achieved through dedicated field equipment from the Hardware Infrastructure or control algorithms inside the Cyber Infrastructure.

The design of such systems involve conceptualizing the architecture and implementing the software infrastructure and algorithms or developing a meta-design approach, where formal tools are implemented so as the system achieve autonomy and real-time flexibility, materialized as self - design. In terms of objectives, CPS aims at extracting, transporting, storing, processing and reasoning throughout huge amount data using specific perception mechanisms [9]. This is further purposed for decision making in the physical environment workflows, in order to ensure the fulfillment of specific performance criteria like:

- **robustness**, which can be ensured either through control or through the structural design of the system (self - robustness)
- **evolving capabilities**, which are a meta – process (the evolution of the evolving dynamics)
- **adaptiveness**, which can be materialized for the parameters, the structure and the processes
- **scalability**, which is ensured through compositionality and composability
- **resilience**, which is achieved through adaptiveness
- **redundancy**, which is a consequence of the requirements of robustness
- **optimality**, which is strongly correlated with the resource constraints

The complex systems are characterized by [10]:

- the large scale of the systems
- the complex dynamics distributed along all levels of granularity
- the interactions which must be preserved and thus reductionism approaches cannot be used
- the significant contribution of the local details in the whole picture through emergence
- the presence of structural uncertainties
- the wide geographical distribution
- the vast diversity and heterogeneity of the systems and processes involved
- the large number of procedures and workflows simultaneously operated
- the involvement of many human actors having varied and complementary expertise
- the strict safety requirements
- the need to achieve a high operational reliability

Consequently, classical formalism are not useful in these contexts and new formal methods and tools adapted for such specificities are required to be developed in order to:

- intuitively and intimately understand global emergent particularities

- analyze global input – output behaviours
- design control strategies
- analyze constraints
- analyze the interactions between systems and components
- analyze emerging contradictions and non-causalities

The main goal pursued in the development of new modeling formalisms is to obtain an optimal balance between **complexity**, **performance** and **efficiency**. For classical modeling methods, the performance and efficiency is achieved through an exponential increase of complexity of the model and the modeling process. The performance and efficiency bring versatility while the simplicity (lack of complexity) provides an intuitive character and support for the user. A powerful and efficient modeling formalism has:

- a high degree of generality and flexibility, being suitable for varied contexts
- a high degree of fidelity, being able to capture all the particularities of a complex system
- a modular nature using scalable and composable models
- a multi – resolution character simultaneously supporting both top – down and bottom – up approaches
- mechanisms for describing and managing real uncertainties and nonlinearities

The high level of heterogeneity of the actors involved in the modeling process and their expertise is focused more towards the technological process than on the behavioural modeling. Moreover, the formalism should assist and support the users and provide an environment for developing ideas and for organizing, clarifying and validating their concept. Consequently, the formalism should be simple and characterized by:

- transparency for the user
- an intuitive language, close to human reasoning
- the inclusion of semantics
- the facilitation of a heuristic analysis of the phenomenon and the provision of an intuitive picture
- low language complexity and linguistic universality

3. Modeling formalisms for complex systems

In this Section, several key modeling formalisms are detailed and a classifying picture map is outlined. Each of these methods was designed for addressing specific and limited facets from the whole picture of the Complex Systems. Such being the case, up to this moment, there is no general method or integrative approach which can be applied in a straight-forward manner to any type of context. The exposition below is structured as a guideline having as main focus the characteristics of the contexts in which the methods are preferably used and includes: multi – agent systems, contracts, network structural analysis, dependability analysis, passivity theory and hybrid systems.

The agents are an abstract representation that encapsulates the behavior of atomic entities which interact with each other and with the environment [11]. Therefore, they exhibit attributes of intelligence like the autonomy, adaptability, learning capability and reactivity. The internal behaviour can be implemented in many formal manners which include: state space dynamic models, differential equations, state machines, algorithms and procedures, logical rules or statistical models. The interaction between the agents and the environment is assisted by dedicated semantic interfaces. This formalism can be used for modeling contexts where the behaviour is more of a consequence of the interactions than an outcome of algorithmic procedures or supervised coordination. Such applications include: traffic modeling, crowd simulation, trading and economic behaviour modeling or social behaviour modeling. In such emergent and chaotic contexts, the modeling is not explicitly done by an actor, but achieved through simulation. In addition to this, agents can be used for operational purposes. Many entities with dedicated tasks are developed and left to interact in an environment so as to achieve a global system with advanced capabilities. Such applications include: advanced cyber security software systems, robotic process automation or autonomous warehouse robots.

Contracts are a formal method of validating and generating abstract behaviours based on requirement analysis and design [12]. Like the agents, multiple entities are developed which are meant to interact with each other and with the environment. Using a dedicated formal language, a set of local constraints are defined for each entity. Further on, this can be exploited in two approaches, analysis and synthesis. The design is validated by analyzing if the global constraints are satisfied. This is achieved using dedicated composition operators used to infer global contracts from local ones. Additionally, local and global behaviours, also represented as contracts, can be defined so as to ensure that global constraints are satisfied. Typical tools used for contracts include temporal logic theory in conjunction with interacting state machines for which formal validation of the evolution can be achieved. Suitable applications to use contracts include: design of large scale software systems, development of critical embedded systems or validation of parallel software tasks with critical scheduling constraints.

The network structural analysis is an extension of the graph theory where previously developed theoretical results are adapted for analyzing the interactions in complex systems [13]. The nature of the interactions between elements, and the topology of the network have a major impact in the dynamics, stability, reliability, robustness and performance of the whole system. Consequently, this outcome can be assessed in terms of specific features using dedicated theoretical tools. Global properties are defined for graphs and portions of the graphs, like the degree distribution or the clustering coefficient, which provide a broad picture of the system's architecture. Local properties for nodes and connections, like the node centrality or edge betweenness give insightful details into system components which can highlight critical areas where special measures should be taken. This formalism is commonly used for social modeling, structural robustness analysis for complex systems or critical network design.

The dependability analysis is a variation of the structural analysis and is also based on the graph theory. However, in this approach the focus is more towards analyzing the causality of the network relationships [14]. In this respect, the graphs used are primarily oriented and the properties analyzed are local. In data processing for sensor networks or IoT, the flow is usually sourced from many distributed modules towards singular, aggregating nodes, while supervision and configuration communication flow in the opposite direction. The topology of the network dictates how the routing algorithms should work and if there are any critical nodes which should be replicated. In event detection applications the system is similar, but the focus is more towards analyzing the event ontology in order to determine how the data gets processed. In critical applications, where high availability is required, dynamic reconfiguration mechanisms are developed. They need to consider all intimate correlations between system resources, components and the trajectory of allowed states during the reconfiguration. Task allocation in large scale computing clusters is yet another domain in which this formalism is useful. A perfect matching between the structural dependencies of the problem to be solved and the dependencies in hardware architecture is envisaged.

Passivity theory is a dynamic design method used for ensuring the composability of systems [15]. It is suitable for applications where global stability is required when interconnecting multiple components and subsystems which are not necessarily stable. The main idea of this approach is to ensure the passivity locally through designed dynamics which is further preserved during the coupling. Applications dedicated to this formalism include the design of Networked Control Systems (NCS) where multiple heterogeneous components (physical systems, computing modules, sensors and actuators) are interconnected through unreliable networks where global robustness is required. Also it can be used for critical dynamic systems where specialized technological processes, which are highly nonlinear and unstable, like nuclear and thermal power plants or jet planes, need to be managed.

Hybrid systems are a class of dynamic design methods used in conceptualizing and analyzing complex control systems [16, 17]. This formalism manages to capture both facets of such systems, the discrete time, which is a characteristic of the computing modules, communication and control algorithms, and the continuous time, which is due to the physical processes that are involved. A dedicated modeling language is provided to describe local hybrid behaviour which is further combined in order to analyze it at global level. The synthesis is made either by simulation using a trial and error approach, either using dedicated formal methods and algorithms to develop control strategies. This formalism is used for designing supervision strategies or scheduling in complex dynamic contexts.

4. Relational modeling for complex systems design

All the above detailed formalisms are narrow and specialized and there is a strong need for new methods that address horizontally multiple facets of Complex Systems. A common approach is to embed previously developed research results which through combination, adaptation and development were found to be effective in specific applications like multi-agent systems, autonomous collaborative robots, social networks or real time optimization using bio-inspired and emergent methods.

Local features of complex systems are very specific and human actors involved have diverse and well-delimited expertise. In this particular context, defined as **distributed expertise**, all local models developed by each actor can be unified using specialized mechanisms. This approach is defined as **concurrent modeling**. The main difficulty arises from the fact that there is an increased modeling equivalence encountered. The same features can be modeled using multiple approaches that are partly equivalent and yet particular. So, the modeling process is more of a craftsmanship to chose, to adapt, to combine and to enhance one or more methods in a very tight connection with all the specificities of the application involved.

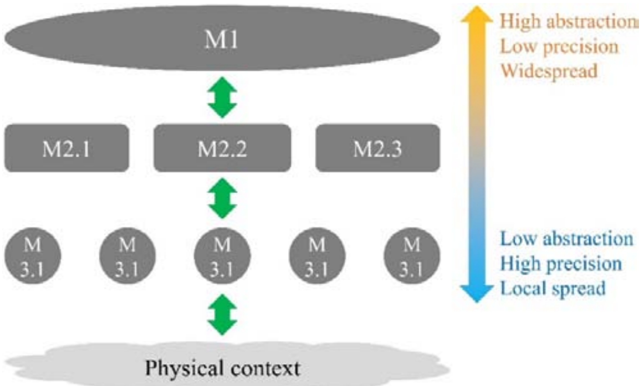

Fig. 2. Equivalent approach in modeling a certain physical context using different formalism having particular features in terms of abstraction, precision and spreading.

In Fig. 2 is depicted a conceptual example where multiple formalisms in different configurations can be equivalently used for modeling a certain context. Each approach is equivalent, considering that the same physical context is modeled with a high degree of fidelity. However, each of them have specific features being suitable in certain applications. Modeling formalisms subscribe to the IPDI (increasing precision with decreasing intelligence) concept and there is always a tradeoff that has to be made [18]. High abstraction formalisms are powerful and are able to cover large aspects and facets of the complex system with the drawback of having low precision. Equivalently, multiple low abstraction formalism can be used, which provide high precision insights, but limited and narrow spread.

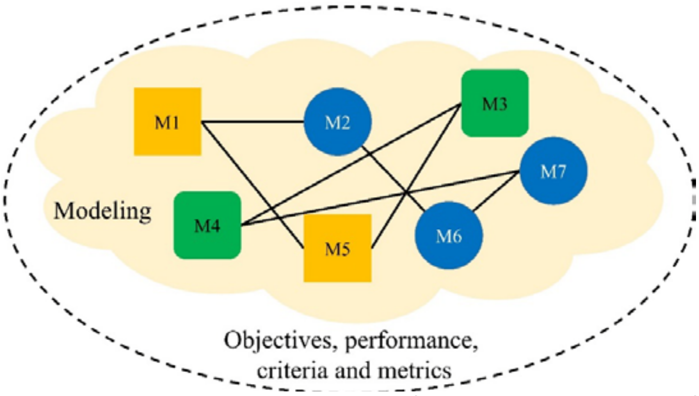

Fig. 3. The Modeling environment inside the Cyber Infrastructure with multiple formalisms and models combined in a unified configuration.

Merging different formalisms and models takes place inside the Cyber Infrastructure as depicted in Fig. 3. The configuration of models inside the Modeling environment is highly dynamical so as to cope in real time with required objectives, performance criteria and metrics. Consequently, the environment is highly **heterogeneous**, integrating multiple types of formalisms (e.g. mathematical models, probabilities, rule-based models, fuzzy logic, state machines, bio-inspired models etc.). Therefore, the implementation of the models require increased **composability** and **compositionality** in order to facilitate the dynamic matching and reconfiguration process.

This is achieved through proper design of dedicated interfaces that support the interconnection of different formalisms and models which become, together with the modeling environment, a **meta-model** [19].

A first approach to address this concept is based on the relational modeling [20, 21] which is extended in multiple directions and adapted in the context of complex systems.

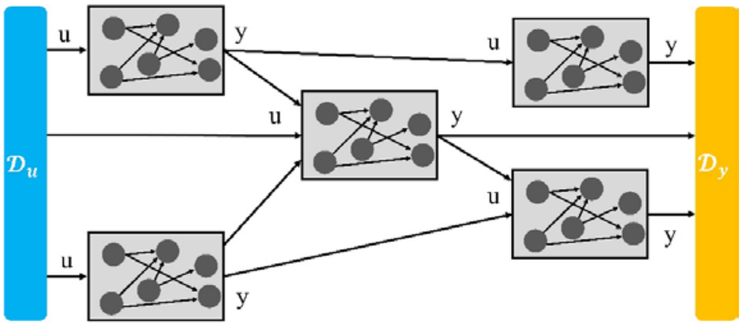

Fig. 4. A modeling environment example with five interconnected models.

The conceptual modeling framework combine three complementary formalisms:

- **logic – algebraic relational models,** where the knowledge and the behaviours are represented though a set of logical predicates combining multiple attributes
- **uncertainties,** represented through probabilities
- **numerical parameters** and algebraic conversion equations for describing precise information

As depicted in Fig. 4, the environment is a composite of multiple interconnected relational models, each of them abstracting a real atomic entity. For example, a model is considered, having two inputs, $\boldsymbol{u_1}$ and $\boldsymbol{u_2}$, one internal parameter, $\boldsymbol{w_1}$ and two outputs, $\boldsymbol{y_1}$ and $\boldsymbol{y_2}$. Each of them is an abstraction of a physical parameter or component of the physical entity considered, for example: a temperature, a pressure, a command, a signal, a state, a module etc. For each $\boldsymbol{u_k}$ input, $\boldsymbol{w_k}$ internal parameter and $\boldsymbol{y_k}$ output, a set of attributes is defined:

$$\alpha_{u_k} = \begin{bmatrix} \alpha_{u_k}^{1} \\ \alpha_{u_k}^{2} \\ \vdots \end{bmatrix}; \alpha_{w_k} = \begin{bmatrix} \alpha_{w_k}^{1} \\ \alpha_{w_k}^{2} \\ \vdots \end{bmatrix}; \alpha_{y_k} = \begin{bmatrix} \alpha_{y_k}^{1} \\ \alpha_{y_k}^{2} \\ \vdots \end{bmatrix} \tag{1}$$

Each attribute expresses a feature or a quality the related parameter owns. For example: "*the temperature is optimal*", "*the pressure is exceeded*" or "*the compressor is malfunctioning*". Some of the inputs and the outputs can be treated numerically, especially for signals and measurements. Consequently, for each of its attributes, an algebraic conversion equation is defined, which correlates the numerical value of the parameter with the truth value of the attribute:

$$\hat{\alpha}_{u_i} = \begin{bmatrix} \hat{\alpha}_{u_i}^{1} \\ \hat{\alpha}_{u_i}^{2} \\ \vdots \end{bmatrix}; \hat{\alpha}_{y_i} = \begin{bmatrix} \hat{\alpha}_{y_i}^{1} \\ \hat{\alpha}_{y_i}^{2} \\ \vdots \end{bmatrix} \tag{2}$$

The input – output behaviour of the model is defined using a set of logical predicates ($\boldsymbol{F}$), which defines strutural relationships between all the attributes. This creates an inference mechanism for reasoning output parameters from input and internal ones. For example, a consequence is defined using a conditional predicate:

"***IF*** *the temperature is optimal and the pressure is exceeded*
THEN *the compressor is malfunctioning*"

which can be symbolically defined as:

$$F_1\colon \boldsymbol{IF}\ \alpha_{u_1}^{3} \wedge \alpha_{u_2}^{2}\ \boldsymbol{THEN}\ \alpha_{y_1}^{1} \tag{3}$$

A priority constraint between two attributes, for example:

"The compressor is not working, unless the pressure is not optimal "

can be symbolically defined using standard logical operators as:

$$F_2\colon \alpha_{u_2}^{1} \vee \neg\alpha_{y_1}^{2} \tag{4}$$

Each of the predicates has a **trust factor** attached represented through a probability within $[\mathbf{0},\mathbf{1}]$:

$$(F_1,\Pi_1)\ ;\ (F_2,\Pi_2) \tag{5}$$

These trust factors express modeling uncertainties which can originate from periodic or uncertain phenomena, previous experience or the degree of expertise of the actor which defined the predicate.

Input and output domains ($\boldsymbol{\mathcal{D}_u}$ and $\boldsymbol{\mathcal{D}_y}$) are defined either through sets of predicates with attached trust factors ($\boldsymbol{F_u},\boldsymbol{\Pi_u}$ and $\boldsymbol{F_y},\boldsymbol{\Pi_y}$), for logical parameters, which express structural relationships between some input and output attributes, either through probability density functions ($\boldsymbol{p_u}(\boldsymbol{u})$ and $\boldsymbol{p_y}(\boldsymbol{y})$), for numerically treated parameters, like Gaussian distributions or interval distributions.

The two categories of probabilities used in defining the input and output domains allow capturing the uncertainties that may arise in signals, data and information due to noise effects or untrusty estimations. Three fundamental environment – level procedures are defined: the **analysis** problem, the **decision making** problem and the **structural equivalence**.

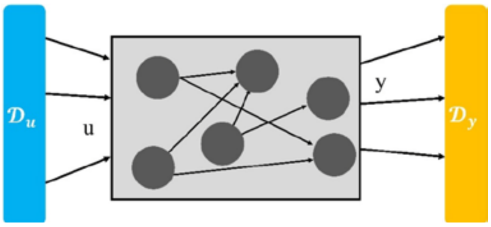

Fig. 5. Structural equivalence is determining the model which is the equivalent of the entire environment in terms of input-output behaviour

The analysis problem is defined as follows: given all the knowledge for all the models in the environment $(\boldsymbol{F_i}, \boldsymbol{\Pi_i})$ and the input domain $(\boldsymbol{\mathcal{D}_u})$, find the corresponding output domain $(\boldsymbol{\mathcal{D}_y})$, such that all the predicates are satisfied and all probabilities are consistent. The decision making problem is analogous and is defined as follows: given all the knowledge for all the models in the environment $(\boldsymbol{F_i}, \boldsymbol{\Pi_i})$ and a required output domain $(\boldsymbol{\mathcal{D}_y})$, find the corresponding input domain $(\boldsymbol{\mathcal{D}_u})$, such that all the predicates are satisfied and all probabilities are consistent.

The structural equivalence is depicted in Fig. 5. where the goal is to find an equivalent model which behaves as the initial environment in terms of input-output.

The multi resolution and concurrent modeling is supported by allowing to simultaneously define models at four distinct levels of detail:

- **inter – relational**: define the models and the connections between them
- **intra – relational**: define the internal causality and the correlations between the attributes of a model
- **logical**: define the behaviour of a model using logical predicates and trust factors
- **numerical**: define certain inputs and outputs that are treated numerically for increased modeling fidelity and precision

Use case scenarios appropriate for harnessing the relational modeling framework include any application where multiple heterogeneous entities are interacting and where the local behaviour is known but the emergent behaviour, due to the interactions, cannot be estimated in a straight-forward manner. Such applications include: design of IoT architectures, management of critical infrastructures, control of large scale complex workflows or management of cyber enterprises.

The first step in applying this formalism is to decompose the complex context considered and identify atomic entities. Each entity is abstracted as a model inside the modeling environment. Further, the interactions between the entities are represented through connections between models. For each model, the input and output interfaces and internal parameters are defined (as input, internal and output parameters and attributes). The input – output behaviour of each entity is defined using sets of logical predicates with attached trust factors. The analysis problem at the environment level is useful for assessing and estimating the emergent behaviour and outcome of the complex system in a specific context defined by a certain input domain. The decision making problem is useful for designing control strategies, i.e., determine the admissible input domain given a required outcome. The structural equivalence is suitable for assessing if the interaction of multiple components is neutral (the behaviours compensate each other) or for eliminating redundancies.

5. Conclusions

In this paper we have proposed a new modeling framework which provides an efficient tool for describing and analyzing the behaviour and the structural correlations in highly heterogeneous contexts and complex systems. Its main disadvantage compared to other methods is the lack of dynamics. However, multiple advantages are provided which include: the modeling language is intuitive, simple and universal, being close to human reasoning, semantics are included, the approach is very flexible and is suitable in any context, complex behaviours can be defined with a high degree of fidelity, resulting models are easily composable for modeling emergent phenomena, uncertainties and nonlinearities can be embedded and the modeling can be approached in a multi-resolution and concurrent manner.

This formalism can be used in applications where multiple entities are interacting and the emergent behaviour is the result of both the local behaviours and the interactions. It can also be used for systems under development where some of the particularities are undefined, unclear or variable.

Future work to further extend this approach includes the development of efficient algorithms and software modeling platforms required to validate this formalism.

R E F E R E N C E S

[1]. *D. Braha, A. A. Minai, and Y. Bar-Yam, Eds.*, Complex Engineered Systems: Science Meets Technology. Berlin, Heidelberg: Springer, 2006.

[2]. *L. B. Rainey and A. Tolk*, Modeling and Simulation Support for System of Systems Engineering Applications, 1 edition. Hoboken, New Jersey: Wiley, 2015.

[3]. *E. Geisberger and M. Broy, Eds.*, Living in a networked world: Integrated research agenda Cyber-Physical Systems. München: Utz Verlag GmbH, 2015.

[4]. *F. Lamnabhi-Lagarrigue et al.*, “Systems & Control for the future of humanity, research agenda: Current and future roles, impact and grand challenges,” Annual Reviews in Control, vol. 43, pp. 1–64, Jan. 2017, doi: 10.1016/j.arcontrol.2017.04.001.

[5]. *Y. Bar-yam*, Dynamics Of Complex Systems, 1 edition. Reading, Mass: CRC Press, 1999.

[6]. *S. Wolfram*, Cellular Automata And Complexity: Collected Papers, 1 edition. Reading, Mass: Westview Press, 2002.

[7]. I. Dumitrache, “From model-based strategies to intelligent control systems,” WSEAS Transactions on Systems and Control, vol. 3, no. 6, pp. 569-575, 2008.

[8]. *L. Esterle and R. Grosu*, “Cyber-physical systems: challenge of the 21st century,” Elektrotech. Inftech., vol. 133, no. 7, pp. 299–303, Nov. 2016, doi: 10.1007/s00502-016-0426-6.

[9]. *I. Dumitrache, S. I. Caramihai, M. A. Moisescu, and I. S. Sacala*, “Neuro-inspired Framework for cognitive manufacturing control,” IFAC-PapersOnLine, vol. 52, no. 13, pp. 910–915, 2019.

[10]. *E. A. Lee*, “The Past, Present and Future of Cyber-Physical Systems: A Focus on Models,” Sensors, vol. 15, no. 3, Art. no. 3, Mar. 2015, doi: 10.3390/s150304837.

[11]. *K. H. van Dam, I. Nikolic, and Z. Lukszo*, Agent-Based Modelling of Socio-Technical Systems. Springer Netherlands, 2013.
[12]. *P. Nuzzo*, Compositional design of cyber-physical systems using contracts, EECS Department, University of California, Berkeley, 2015.
[13]. *D. B. West*, Introduction to Graph Theory, 2 edition. Upper Saddle River, N.J: Pearson, 2000.
[14]. *W. Zamojski and J. Sugier*, Eds., Dependability Problems of Complex Information Systems. Springer International Publishing, 2015.
[15]. *N. Kottenstette*, "Digital Control of Multiple Discrete Passive Plants Over Networks," International Journal of Systems, Control and Communications (IJSCC), 2011.
[16]. *A. Platzer*, Logical analysis of hybrid systems: proving theorems for complex dynamics. Springer Science & Business Media, 2010.
[17]. *A. Platzer*, Logical foundations of cyber-physical systems. Springer, 2018.
[18]. George N.Saridis. "Analytic formulation of the principle of increasing precision with decreasing intelligence for intelligent machines," Automatica, vol. 25, no. 3, pp. 461-467, 1989.
[19]. *I. Dumitrache, S. I. Caramihai, M. A. Moisescu, I. S. Sacala, L. Vladareanu, and D. Repta*, "A perceptive interface for intelligent cyber enterprises," Sensors, vol. 19, no. 20, p. 4422, 2019.
[20]. *Z. Bubnicki*, Modern Control Theory. Berlin Heidelberg: Springer-Verlag, 2005.
[21]. *Z. Bubnicki*, Analysis and Decision Making in Uncertain Systems. London: Springer-Verlag, 2004.