The Shortcomings of Models
(Cyber-Physical system a study case)

Dr. Mohammed Molhem *

(Received 12 / 6 / 2024. Accepted 15 / 4 / 2025)

☐ ABSTRACT ☐

System design starts with specifications as a first step; hence, the correct specifications need a valid model that can capture the aspects of the system. The fundamental difference between systems has led to a variety of Models of Computation (MoCs), therefore, no model can specify all systems individually and more than one model is needed to achieve that. The Cyber-Physical Systems (CPSs) is a special case of a complex system, for several reasons; it is a heterogeneous, interactional system and time is a critical parameter in it. On the other hand, CPS is an edge system in new-generation information technologies such as IoT, big data, cloud computing, artificial intelligence (AI), etc. Therefore, for CPS system design, a special kind of model is required which needs to be more flexible and have a great ability to specify heterogeneous systems. This paper points out the shortcomings in MoCs through a proposed criterion based on the system's mathematical fundamentals. The results of applying this criterion to most common MoCs show the structural defect in MoCs, and the reasons for their limitations in designing CPS systems. Finally, some examples are introduced as study cases to confirm the results of this criterion.

Keywords: MoC, CPS, edge system, Heterogeneous system, SLDL, Hybrid Model, VSIA.

* **Associate Professor, Department of Computer and Electronic Engineering, Tartous University, Tartous, Syria. m.molhem@tartous-univ.edu.sy.**

القصور الذاتي في النماذج
(النظام الفيزيائي المحوسب CPS كحالة دراسية)

د. محمد ملحم *

(تاريخ الإيداع١٢ / ٦ / ٢٠٢٤. قُبِل للنشر في ١٥/ ٤ / ٢٠٢٥)

□ ملخّص □

الخطوة الأولى في تصميم أي نظام هو توصيفه specification، والتوصيف الصحيح يحتاج إلى نموذج حاسوبي Model of Computing (MoC) مناسب يتوافق مع طبيعة النظام. تتعدد أدوات النمذجة لأسباب أهمها، الإختلاف الجوهري بين الأنظمة من حيث الخصائص والمواصفات بالتالي عدم قدرة أي نموذج منفرداً على توصيف كل الأنظمة، لهذا السبب يتم استخدام أكثر من نموذج لتحقيق ذلك. تُعد الأنظمة الفيزيائية المحوسبة (CPSs) حالة خاصة من الأنظمة المعقد، وذلك لعدة أسباب؛ فهي أنظمة غير متجانسة وتفاعلية، والزمن مُعامل حاسم فيها. من ناحية أخرى، يُعد نظام CPS نظام حافة edge system في الجيل الجديد لتكنولوجيا المعلومات مثل إنترنت الأشياء IoT والبيانات الضخمة والحوسبة السحابية والذكاء الاصطناعي. يتطلب تصميم نظام CPSنوعاً خاصاً من النماذج التي تتمتع بالمرونة والقدرة على توصيف الأنظمة غير المتجانسة heterogeneous. تُبين هذه الورقة أوجه القصور والعيوب في النماذج الحاسوبية MoCs من خلال اقتراح معيار تقييم يستند إلى الأسس الرياضية للنظام. تُشير نتائج تطبيق هذا المعيار على معظم النماذج الحاسوبية الشائعة إلى الخلل بنيوي فيها، وتُبين أسباب محدوديتها في تصميم أنظمة CPSs. وأخيراً، يُقدَّم البحث بعض الأمثلة كحالات دراسية لتأكيد نتائج هذا المعيار.

الكلمات المفتاحية: MoC, CPS, edge system, Heterogeneous system, SLDL, Hybrid Model, VSIA.

*أستاذ مساعد، كلية تكنولوجيا المعلومات والاتصالات، جامعة طرطوس، طرطوس، سورية.

ايميل: m.molhem@tartous-univ.edu.sy ، رقم الأوركيد: https://orcid.org/0000-0001-9935-6258

Introduction:

Systems are viewed differently according to researcher's perspectives and their ways of thinking. However, all references indicate that the system is an interaction of different groups of components to achieve a logical end, to form an integral of a complex whole, to form a unified whole, or to perform a function that cannot be done by any component individually [1]. The system is seen through a range of terms, such as inputs/outputs and the relationship between them.

The CPS system is an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system. It is not sufficient to separately understand the physical components and the computational components, we must instead understand their interaction. The term CPS is the conjunction of physical processes, computation, and communication, and relates to the currently popular terms Internet of Things (IoT), Industry 4.0, the Industrial Internet, Machine-to-Machine (M2M), the Internet of Everything, and the Fog (like the Cloud, but closer to the ground). All of these reflect a vision of a technology that deeply connects our physical world with our information world.

The system is represented by a model, and this model is a simplification of the real system and contains exactly those characteristics and properties. So, a model is not an identical copy of the system, and moving from system to model needs a deep understanding and a good definition of the system. Modeling is a conceptual process that reflects system specifications. The first step of designing the model is a mathematical definition level, then in the next step the programming languages capture these mathematical formalisms and transform their semantics into MoCs as shown in figure (1).

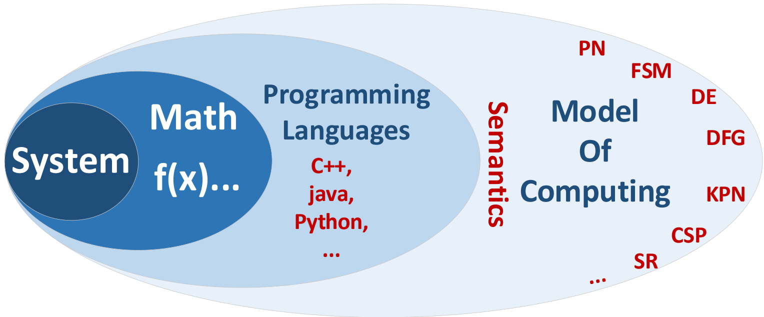


Figure 1. Abstract Levels of Model

The multiplicity of modelling tools reflects the inability of any model individually to specify the heterogeneous system [2,3]. This inability can be understood through the mathematical fundamentals, which is the first level of model construction. Mathematical system theory is a good tool that can capture the basic concepts of representation and construction of systems, and this theory allows us to have a deep understanding of these systems [4, 5].

The main contributions of this work are:

- Focuses on the term CPS, because it is more foundational and durable than all-new generation information technologies, it does not directly reference either implementation approaches (e.g،. the "Internet" in IoT) or particular applications (e.g., "Industry" in Industry 4.0).
- Proposes an analytic methodology to illustrate the mathematical abstraction levels of the system, which reflects the fundamental intellectual problem of CPS.

- The proposed analytic methodology provides that the systems are based on four mathematical bases.
- Clarifies the shortcomings in Models of Computing and presents the methods to recover them.

To prove our view, Ptolemy II (Version 11.0.1_20180619) is used, which is a graphical modeling and simulation environment, and it is suitable for designing and testing various types of models. The rest of this paper is organized as follows.
Related works are presented in Section 2. In Section 3, analysis and classifications of the system. The systematic approach to determine the shortcoming in models is illustrated in Section 4. Section 5, presents an Anti-Braking System (ABS) as an example of a Cyber-Physical System. The conclusion is presented in Section 6.

1. Related works

Most research has focused on the general classification of models and their uses. One of the researches that analyzed the important details of the model is [6, 7], where a group of specialists at George Washington University created a mechanism to classify models called VSI (Virtual Socket Interface) Alliance Taxonomy. This mechanism is done by highlighting the accuracy of the important details in the model according to several axes (temporal detail axis, data value detail axis, function detail axis, and structural detail axis) as shown in Figure (2). In addition to the previous axes, the Programming Abstraction level axis is added to show the concept of hardware and software design in the model.

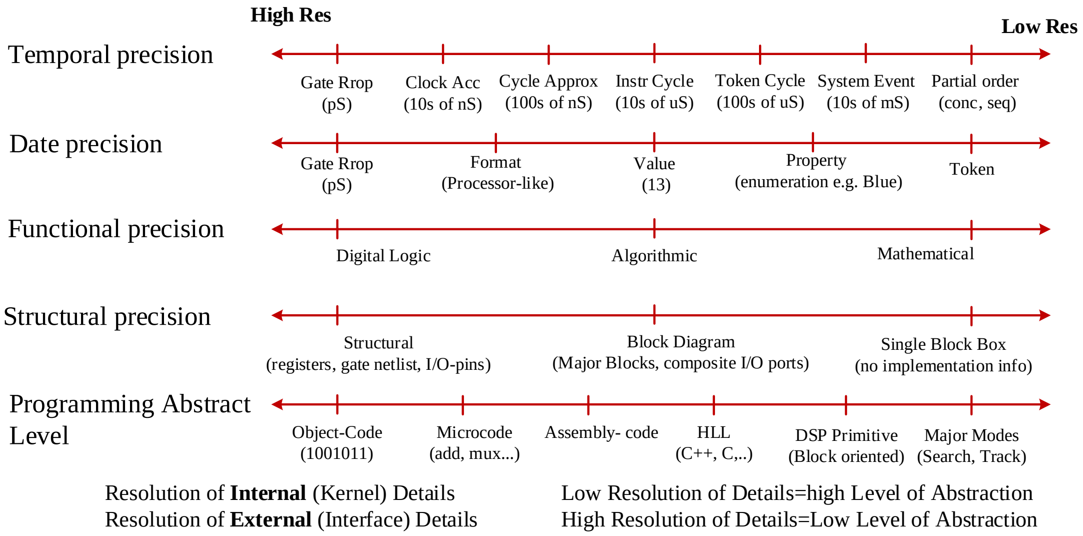


Figure 2. VSIA model classification axes

The Time Accuracy axis highlights details from the highest time accuracy (High Res) picoseconds, to the lowest accuracy (Low Res) which is dependent on events ordered (Partial order). The axis of data accuracy starts from the highest resolution (bit) to the lowest resolution (Token). The axis of functional accuracy begins with Boolean logic and ends with mathematical equations. The axis of structural precision starts with information of big details and ends with blocks with no internal details. The software axis starts with machine language and ends with system-level design language. As an example, figure (3) shows the evaluation of FSM and Petri Net using VSIA mechanism [3, 5, 8, 9].

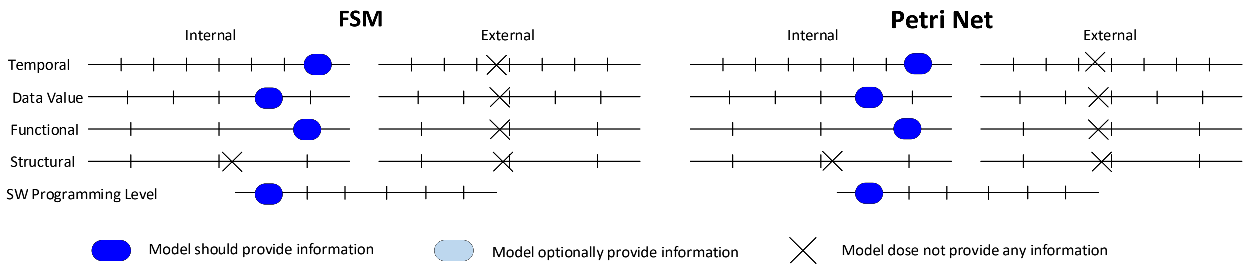


Figure 3. Evaluation of FSM and Petri Net using the VSIA rating mechanism

In [10], M. Burgin and G. D Crnkovic present taxonomy of existing MoCs according to the general notion of natural computation, intrinsic to physical systems, and particularly cognitive computation in cognitive systems (e.g, Existential taxonomy, Temporal taxonomy, Organizational taxonomy, Representational taxonomy, Operational, Process-oriented, and Computation levels taxonomy). In [11] P. A. Fisgwick presents a uniform model design taxonomy whose categories are inspired by categories in programming language principles, the taxonomy includes a set of primitive model types (conceptual, declarative, functional, constraint, spatial) and a way of integrating primitive model types (multimodeling). The authors in [12] survey various approaches to the formal modelling and analysis of the temporal features of computer-based systems, with a level of detail that is also suitable for non-specialists.

2. System Analysis and Classifications

Every system consists of homogeneous partial components. Dynamic components are an example of homogeneous components, they specify by differential equations, and if these components work in a discrete time domain, they specify by difference equations which are another type of mathematical method. The computing systems introduced other types of specifying methods such as discrete events. Discrete events are an advanced form of modelling and simulation, are associated with the existence of simulation languages and algorithms, and support the design of computing systems. This type of polymorphism in modelling is very important, it introduces new concepts such as embedding and quantizing [13].

We mean by embedding; which component is embedded in the other? And what is the relationship between them? For example, discrete-time systems are a subclass of discrete event systems, but it does not mean that any discrete-time system is a discrete events system. On the other hand, quantizing allows for finding the interfacing between the polymorphisms of the system. For example, discrete events are most appropriate in distributed and synchronized simulations, especially when joining a system of discrete events with a system of differential equations. On the other hand, event paths are used to wrap differential equations models and other models, and by quantizing we get a discrete signal on the time axis [2, 4, 14].

Many factors and several methods are used to classify the systems, like time, orientation, modelling techniques, and properties. Some systems are classified according to interaction with the environment, like cyber-physical systems (CPS) or according to their complexity [15]. System classification is not an easy task, Figure (4) summarizes the classification of systems according to several orientations [1, 16, 17].

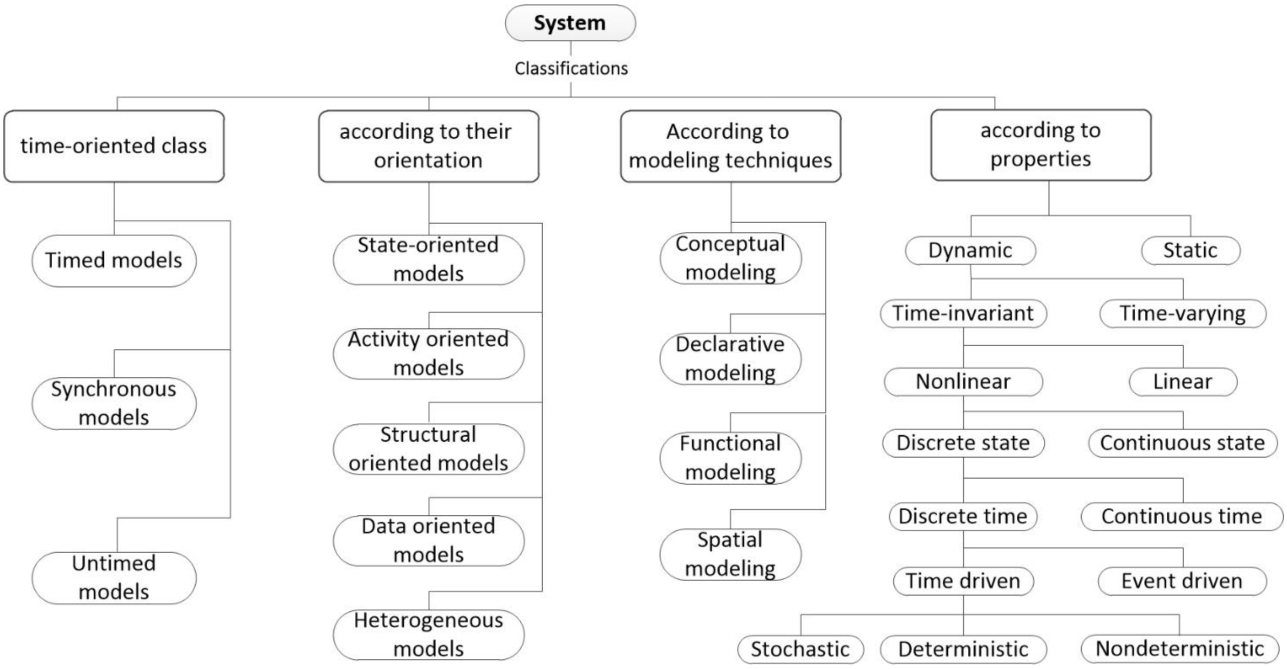


Figure 4. System classifications

2.1 Mathematical Abstraction of the System

The objective of mathematical abstraction is to obtain the hierarchical structure of the system, get enough mathematical knowledge about it, and find out the types of its components. By mathematical abstraction the task of choosing an appropriate model becomes easier [4, 18, 19]. In the following the mathematical abstraction levels were presented as we proposed.

1- Description Level (Level 0): A system is a collection of different subsystems; these subsystems interact to perform a specific function. In this example, we assume the system consists of n components (such as Control, DSP, Communication, Biologic, and Other Components) as shown in figure (5).

2- Formal Level (Level 1): Each component in the system is a process, and all processes interact by signals. In this level, the processes $\{P_i: i = 1,2 \ldots\}$ interact with each other by signals (s_i) as follows:

$$S_{i.out} = P_i(S_{i.in}) \quad (1)$$

3- Input-Output Relation Level (Level 2): To understand the type of input/output relation, the system' data (Time, Variables) types must be well defined [20, 21].

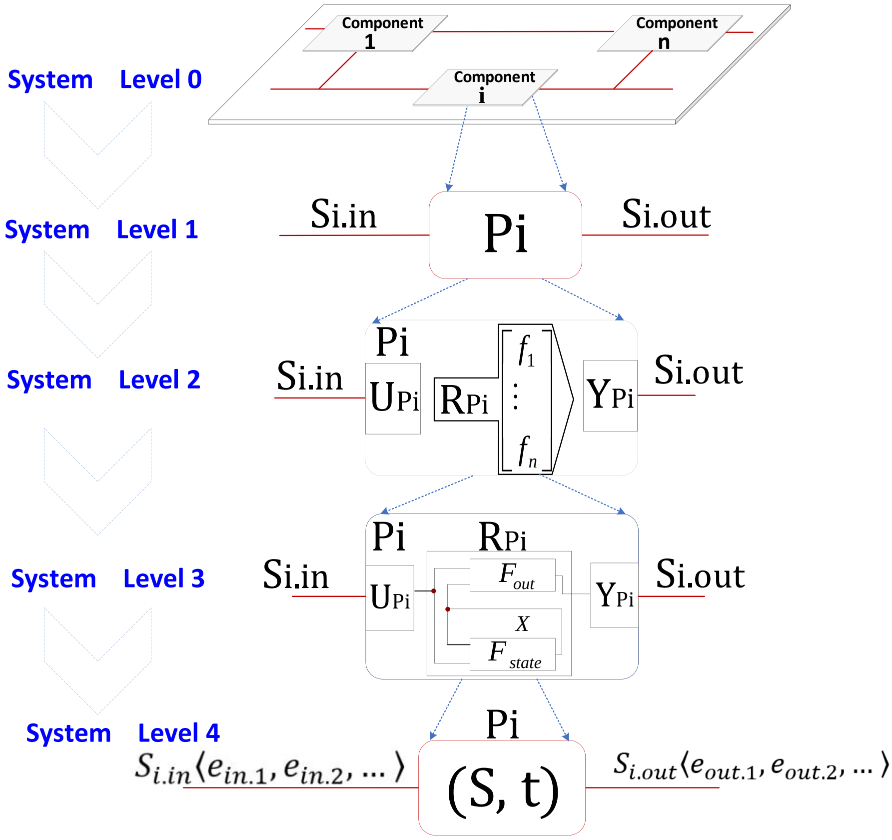


Figure 5. Mathematical Abstraction levels of the system

At the input/output level, there is no information about the properties of the system, it interacts with the environment through set of inputs $U_{Pi} = [u_1, u_2, \ldots, u_r]$ and a set of outputs $Y_{Pi} = [y_1, y_2, \ldots, y_m]$. We assume that each signal s_i whether is output or input $\in (V_{base}, T_{base})$ where T_{base} is set of time, and V_{base} set of values [22,23]. The I/O relation R is changed according to system states. We suppose a set of functions $F = \{f_1, f_2, \ldots, f_n\}$ associated with the states i, and depending on the system's state and input the output change. The system presented at this level as follows:

$IO = (S, U, Y, F)$

U: inputs, Y: Otput, $S(V_{base}, T_{base})$

$F = \{f_1, f_2, \ldots, f_i, \ldots\}$: set of I/O functions

$$F: U \rightarrow Y, \qquad (2)$$

4- State Level (Level 3): At this level appear the concepts of past, present, and future. The current output was produced from the last state and current input. If we propose X is the set of states, F_{state} is the state functions, and F_{out} is the output functions, then the system presented as follows:

$System = (S, U, Y, X, F_{state}, F_{out})$

U: inputs, Y: Otput, X: states set, $S(V_{base}, T_{base})$

$F_{state}: X \times U \rightarrow X$ state transition functions (3)

$F_{out}: X \times U \rightarrow Y$ output functions

5- Variables Level (Level 4): It is the basic level of the system, where all the sets $\{S, U, Y, X\}$ are sets of variables. All variables change as function of time and are presented

mathematically as a pair $\{s(v,t): v \in \mathrm{V_{base}}, t \in \mathrm{T_{base}}\}$ where v is the value of s at time t [5, 24, 25, 26].
After we defined the time relation of our proposed system, we defined our variables.
It's clear from the last level, that the base of any system is a multipair of variables $(v,t) \in (\mathrm{V_{base}}, \mathrm{T_{base}})$ and these variables take multi form as shown in figure (6)

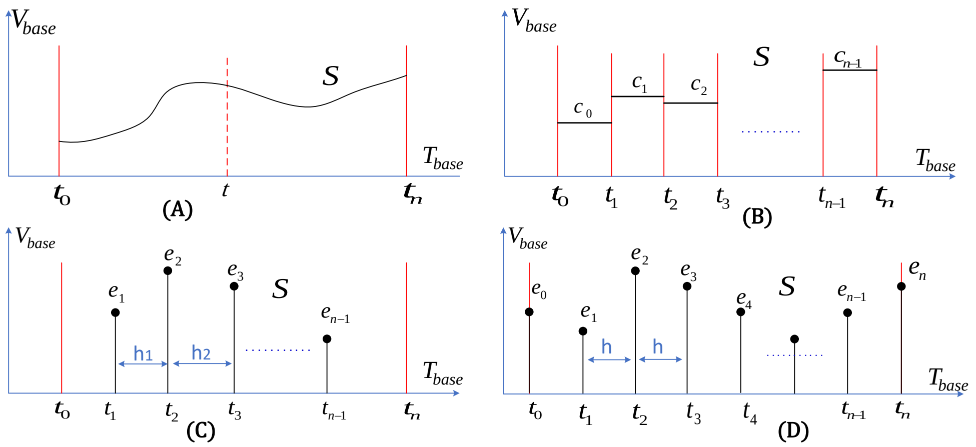


Figure 6. Segments types (A): Unifying continuous pieces within a continuous time range, (B): Unifying fixed pieces at a series of time intervals, (C): Pieces of events within a continuous time range, (D): Pieces of events within a dashed time range.

In the figure (6.A) The segment S has a value at all points $\mathrm{t} \in [\mathrm{t_0}, \mathrm{t_n}] \in \mathrm{T_{base}} \in \mathbb{R}$. Mathematically, the set $\mathrm{V_{base}}$ is n-dimension space of continuous segments, i.e. $\mathrm{V_{base}} = \mathbb{R}^{\mathrm{n}}$ and $\mathrm{S} : [\mathrm{t_1}, \mathrm{t_n}] \to \mathrm{V_{base}}$. Then the system is continuous.
The type of segment shown in figure (6.B) is a subclass of the continuous type shown in figure (6.A). where the segment S taken different constant values $\mathrm{c_0}, \mathrm{c_1}, \dots, \mathrm{c_{n-1}} \in \mathrm{V_{base}} = \mathbb{R}^{\mathrm{n}}$at time points $\mathrm{t_1}, \mathrm{t_2}, \mathrm{t_3}, \dots, \mathrm{t_{n-1}} \in [\mathrm{t_0}, \mathrm{t_n}]$.
The type of segment shown in figure (6.C) is a sampled version of the type in figure (6.A). where the segment S takes different values $\mathrm{e_1}, \mathrm{e_2}, \dots, \mathrm{e_{n-1}} \in \mathrm{V_{base}}$ at arbitrary time points $\mathrm{t_1}, \mathrm{t_2}, \mathrm{t_3}, \dots, \mathrm{t_{n-1}} \in [\mathrm{t_0}, \mathrm{t_n}]$ where the distance between time points not same.
The last type shown in figure (6.D) is a subclass of the type shown in figure (6.C) where the distance between time points is the same equal $\{h: t_i = t_{i-1} + h : h \in \mathbb{R}_{+}\}$.
Summarizing the previous, the system's variables are based on the pair $S(\mathrm{s}, \mathrm{t}) \in (\mathrm{V_{base}} \cup \{\emptyset\} \times \mathrm{T_{base}})$ is taken the forms shown in Figure (7) which is the basic level of the system.

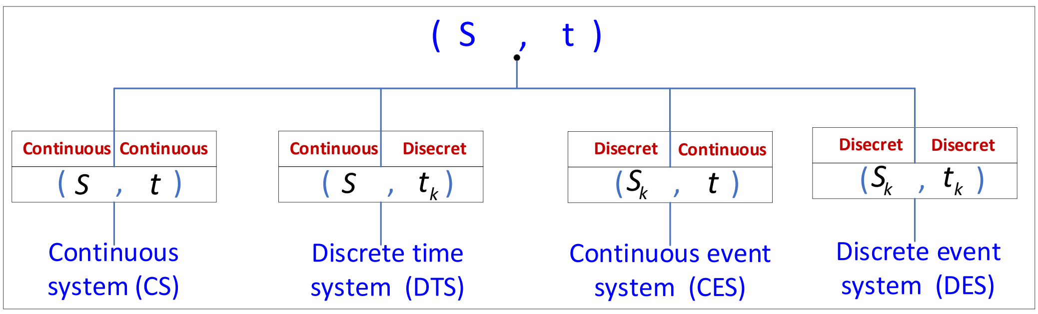


Figure 7. The forms taken by pair (s, t)

The aforementioned multipair of variables represents the mathematical foundation of the system, and to obtain a valid model each pair should be specified by a well-defined method as follows [27, 28]:

- For continuous system (CS), the Differential Equations are a good classical method.
- For discrete time system (DTS), which is a sampled version of a continuous system, so we can use the Difference Equations to solve these system functions.
- For continuous events system (CES), we determine the system functions by using arbitrary Probabilistic Methods and as a special case the Continuous Time Markov Processes.
- For discrete events system (DES), we determine the system functions by using Deterministic Methods (constant time) and as a special case the Discrete Time Markov Processes.

The Time play the main rule in determine the type of methods as shown in the table1.

Table1: Type of methods (mathematical techniques)

Variable / Time	Continuous	Discrete
Continuous	Differential Equations	Probabilistic Methods
Discrete	Difference Equations	Deterministic Methods

The discrete-events is a special method introduced by the computing world. And all the components in computing systems are connected through signals, which are a series of chronological events defined as follows:

signal: $s(e_1, e_2, \dots, e_i, \dots)$
event: $\mathrm{e}_i(\mathrm{v}_i, \mathrm{t}_i)$. (4)
where: $\mathrm{v}_i \in \mathrm{V}_{\mathrm{base}}, \quad \mathrm{t}_i \in \mathrm{T}_{\mathrm{base}}$

So, the system's components are wrapped by an environment of discrete events. As a result, the model is valid when it is based on the aforementioned mathematical bases as shown in Figure (8), which is the main conclusion of the flow diagram.

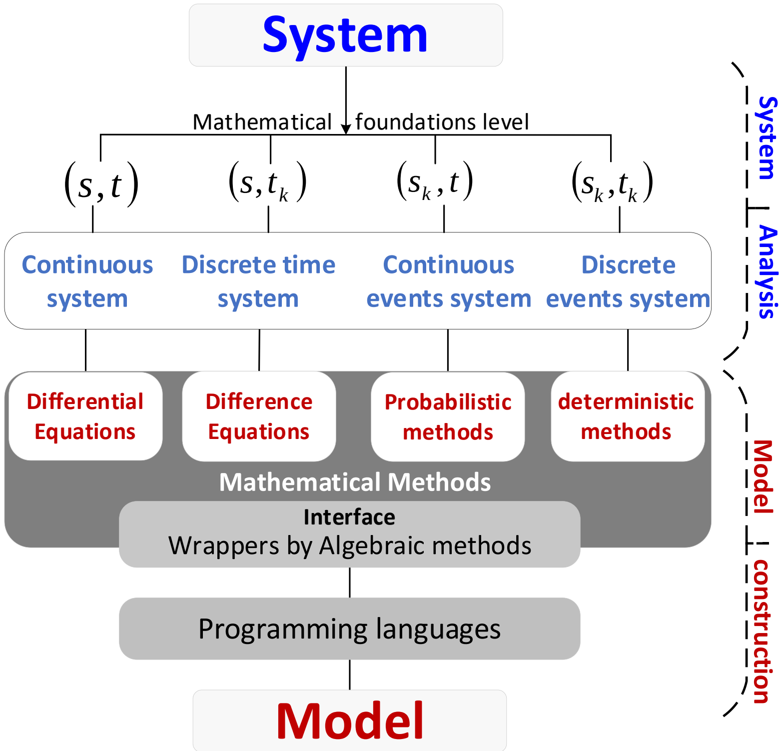


Figure 8. Flow diagram summarizes the analytical approach to system from the descriptive level to the basic mathematical level, and the model design requirements.

2.2 The proposed Criterion

After the mathematical analysis of the system, we obtain the system based on four mathematical foundations as shown in the figure (8). The proposed criterion based on these mathematical foundations, the object of this criterion is to show the ability of the model to capture the heterogeneities in the system, in our approach if the model covered all mathematical foundations of the system (see Figure (8)), then the model has the best chance to be a valid and robust model, therefore the better model is who based on the aforementioned mathematical foundations. Each type of pair of variables reflects one mathematical foundation, and each mathematical foundations presents one type of system, so the valid model **M** is one that fully covers the four foundations as shown in Figure (9).

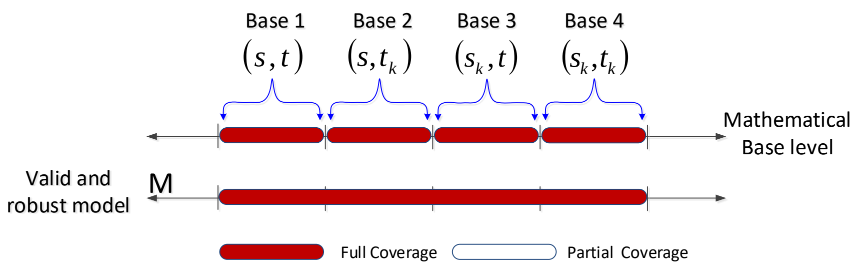


Figure 9. The valid model VM according to our approach

3. Systematic Approach to Determine the Shortcoming in Models

After studying the most important MoCs, and comparing their mathematical foundations with the aforementioned mathematical foundations of proposed criterion, we notice that no model individually can cover all the foundations as shown in figure (10), which indicates a lack in designing the models, and according to our approach that is the essential reason for the models' weaknesses and inability to specify the system. For example, Petri Net and FSM models are one of the most important models used as a systems level design model, and according to their formal definition shown in table 2 they cover Base3 and partially Base4 but they aren't able to cover Base1 and Base2, then they are unable to specify the continuous systems or its sampled version. When using Petri Net and FSM as timed models (Timed Petri Net, Timed FSM) the time become continuous and the models able to cover the Base3 only [9, 29, 30].

Table2: Formal Definition of Petri net and FSM

Petri Net	FSM
$PN = (P, T, A, w, \vec{x}_0)$,	$M = (\Sigma, X, g, x_0, F)$
P is a finite set of places	Σ is a finite alphabet
T is a finite set of transitions	X is a finite set of States
A is a set of arcs, $A \subseteq (P \times T) \cup (T \times P)$	g is a state transition function, $g: X \times \Sigma \to X$
w is a weight function, $w: A \to \mathbb{N}$	x_0 is the initial state, $x_0 \in X$
$\vec{x}_0$ is an initial marking vector, $\vec{x}_0 \in \mathbb{N}^{\|P\|}$	F is the set of final ststes, $F \subseteq X$

By analyzing the formal definition of other models, we obtain the evaluation according to the proposed criterion as shown in Figure (10), it's clear no model individually covers all mathematical foundations which indicates a shortcoming in models. And on the other hand, most of models designed based on the mathematical foundation Base3, which reflect the programming language effects on the MoCs.

To recover the shortcoming in models, should be follow one of these rules

– Merging more than one model into a single design (multimodeling),

– Expanded some models to cover the heterogeneity as much as possible in the system (e.g, Petri Net and color PN (CPN)).

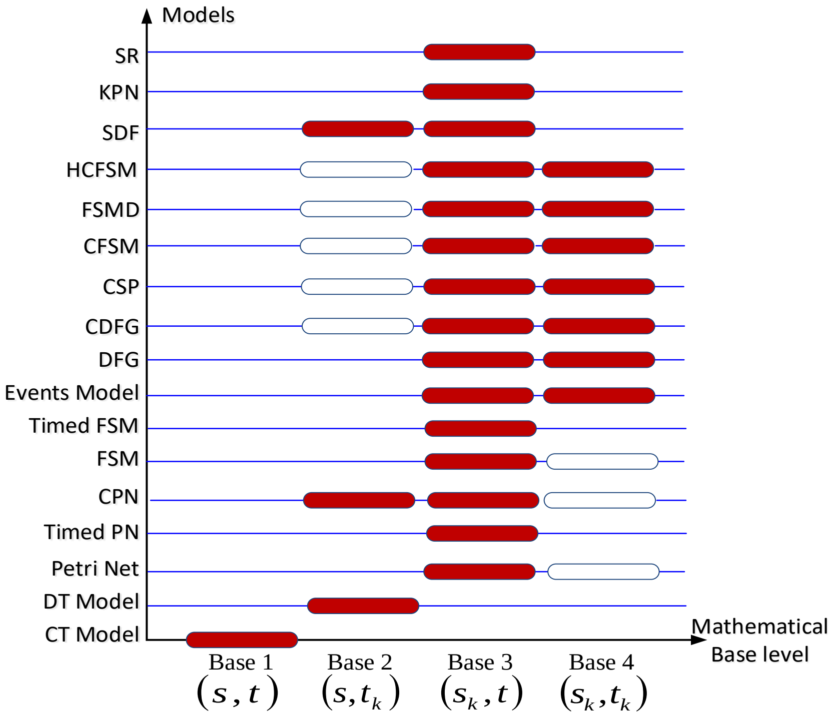


Figure 10. Evaluation of models according to the proposed criterion [3, 5, 9, 19]

4. Study Case (Cyber Physical System)

A cyber-physical system (CPS) is the conjunction of physical processes, computation, and communication as shown in figure (11.a). CPS is an integration of computation with physical processes whose behavior is defined by both cyber and physical parts of the system. The equivalence mathematical foundations of CPS are shown in figure (11.b). Then to modeling the CPS according to the proposed criterion we need a model M_{CPS} shown in figure (11.c). It's clear There is no model in figure (10) fulfills the conditions of specification of the CPS [31].

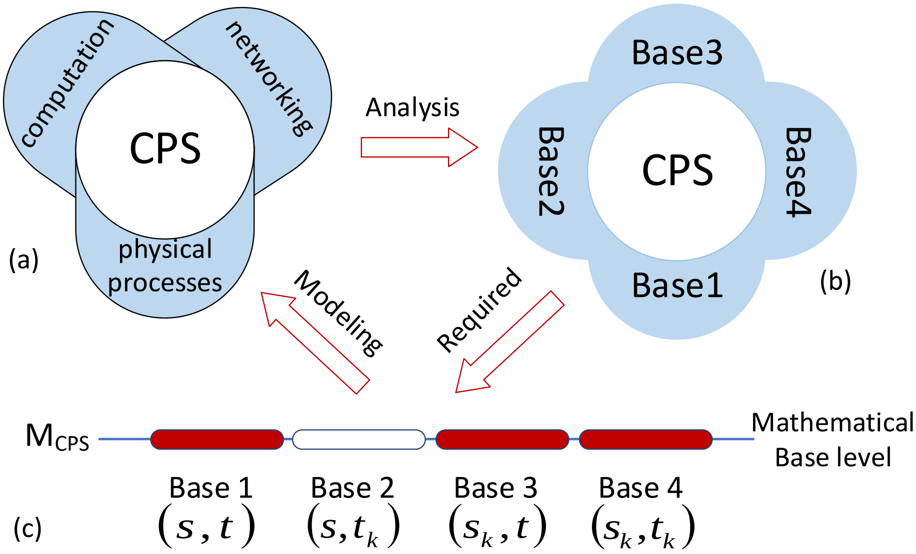


Figure 11. A cyber-physical system (a), the equivalence mathematical foundations of CPS (b), CPS Modeling requirements based on a proposed criterion (c)

- **Example: Anti-Lock Braking System (ABS)**

The present ABS is controlled by the slip ratio and the wheel acceleration, When the braking action is initiated, a slippage between the tire and the contacted road surface will occur, which make the speed of the vehicle to be different from that of the wheels. Through breaking the wheel effected by several forces and torques as shown in Figure (12).

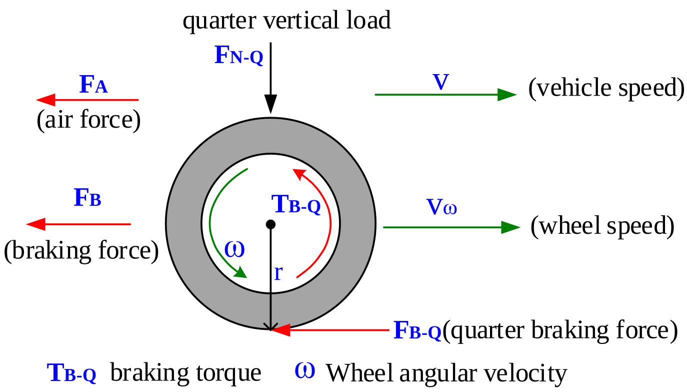


Figure 12. Wheel system parameters during braking action

The slip ratio $\lambda \in [0,1]$ indicates the state of wheel speed, if $\lambda = 0$ the wheel rotates freely, and if $\lambda = 1$ the wheel stopped.

$$\lambda = \frac{v - \omega r}{v}, \quad (5)$$

The braking force given by

$$F_{B-Q} = f(\lambda)\mu F_{N-Q} \quad (6)$$

Where $f(\lambda)$ is the function of Road-Wheel Coefficients friction, that is given as follows:

$$f(\lambda) = c_1\left[1 - e^{-\lambda c_2}\right] - \lambda c_3 \quad (7)$$

Where c_1, c_2, c_3 are the Coefficients that determine the road kind (e.g, dry, wet, snowy).
The angular acceleration of wheel during Braking given by

$$J_w\dot{\omega} = rF_{B-Q} - T_{B-Q} \quad (8)$$

The acceleration of vehicle during Braking given by

$$m\dot{v} = -F_B - F_A \quad (9)$$

Air resistance force given by

$$F_A = c_A A_c \frac{\rho}{2} v^2 \quad (10)$$

Braking torque given by

$$T_B = 2r_d\gamma_d A_p P_B \quad (11)$$

At the beginning the ABS OFF and the vehicle run at normal speed v, and oil braking pressure $P = 0$. When the driver presses the braking pedal ABS ON the vehicle system goes into two states as follows:

States		Vehicle system
Initial	ABS OFF	braking oil Pressure $P = 0$, vehicle run at normal speed
State1	ABS ON	Increasing Pressure $P(t)$, vehicle and wheel velocity decreasing, and $V > V_w$
State2	ABS ON	Reducing Pressure $P(t)$ at initial value, vehicle speed decreasing, and $V = V_w$

ABS is a cyber-physical system, combine two behaviors. First (State1), oil braking pressure increases, and the linear velocity of the vehicle (V) and wheels (V_w) decreases continuously in two different manners $(V > V_w)$. Secondly (State2), when the difference between the wheel's velocity and the vehicle's velocity increases a reduce event appears and becomes $(V = V_w)$ for some moments, then the system moves to a new state of $(V > V_w)$. Neither model shown in figure (10) can be modelling ABS individually, so we need to combine the continuous system (CT Model) and continuous events system (FSM) as a hybrid model. The hybrid model combines the continuous model with FSM [30].
The angular acceleration of wheel and The linear acceleration of vehicle during braking taken in account the testing values shown in table (2) given by

$$\dot{\omega} = 2069.297\mu\left[1.3\left(1 - e^{-10\lambda}\right) - 0.8\lambda\right] - 1.25T_B$$

$$\dot{v} = 14.715\mu\left[1.3\left(e^{-10\lambda} - 1\right) + 0.8\lambda\right] - 0.00024v^2$$

The ABS system move from State1 to State2

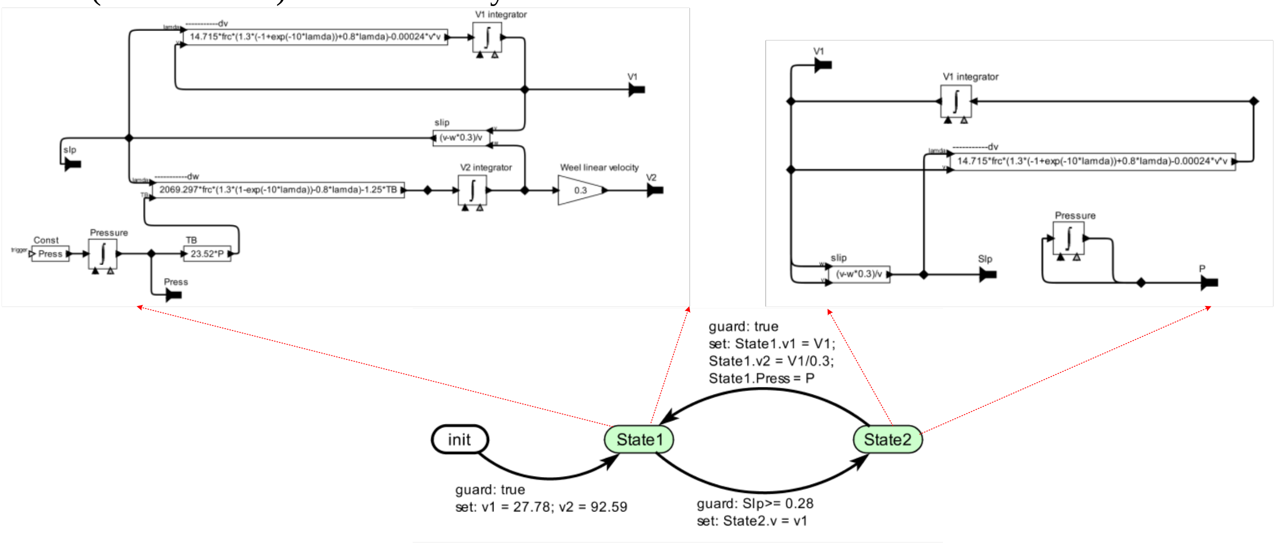
$$S_1 \to S_2 = \begin{cases} 1 & \lambda \geq 0.28 \\ 0 & \lambda < 0.28 \end{cases}$$

Table (2): Vehicle system parameters and testing values

$r[m] = 0.3$	Wheel radius	$A_c[m^2] = 2$	vehicle front face
$\lambda \in [0,1]$	Longitudinal slip ratio	$\rho[kg/m^3] = 1.2$	Air density
$c_1 = 1.3, c_2 = 10, c_3 = 0.8$	Constants	$m[kg] = 1500$	Vehicle mass
$F_{B-Q}[N]$	Quarter of braking force	$\omega[rad/s]$	Wheel angular

			velocity
$\mu \in [0,1]$	Road wheel fraction coefficient	$v[m/s]$	Vehicle speed
$F_{N-Q}[N]$	Quarter of weight force	$T_{B-Q}[Nm]$	Quarter of braking torque
$F_N[N](4 \times F_{N-Q})$	Vehicle load	$F_B[N]$	Total braking force
$c_A = 0.3$	Air force coefficient	$F_A[N]$	Air resistance force
$J_w[kgm^2] = 0.8$	Wheel Inertial moment	$g[m/s^2] = 9.81$	Gravity acceleration
$\gamma_d = 0.4$	Braking disk fraction coefficient	$r_d[m] = 0.15$	Radius of braking disk
$A_p[m^2] = 1.96\text{E} - 3$	Cross-section of braking piston	$P_B[bar], [N/m^2]$	Hydraulic pressure

We test the model using Ptolemy II [32] as shown in figure (13.a), the linear velocity of the vehicle (V) and wheels (V_w) shown in figure (13.b), and the figure (13.c) the sequence events (reduce events) that move the system from State1 to State2 and vice versa.



(a)

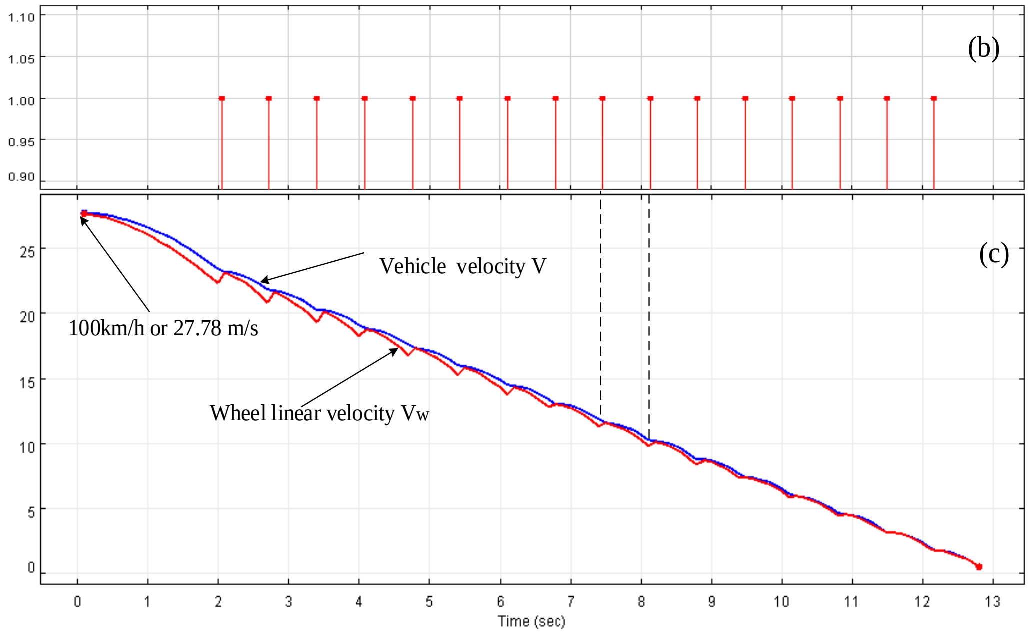


Figure13: (a): Hybrid system model for the ABS system, (b): Reduce Events that move system from state1 to state2, (c): the linear velocity of the vehicle V and wheels V_w during braking

Conclusions and Recommendations:

The paper began with a fact that the model is not a law, it is only a scientific fact that has not been proven correct, it can demonstrate its limitations or disprove only. Therefore, in this research, we analyze the system to reach the foundations which are used to specify any system, and at the same time what foundations should be used in model design. As we said in the introduction the first abstract level in the model is the mathematical level then the programming language transforms it semantically into models. So, the mathematical foundation is the basic building block in the design of any model. The research introduces four types of foundations, and through them the shortcomings are shown. To prove our idea, we presented the ABS as a Cyber-Physical system, and we illustrate that no Model can be modeling ABS individually due to the lack of mathematical foundation. In future research, the work will continue to determine the role of programming languages in this shortcoming, and then introduce a new criterion in classification and evaluating the models.

List of Abbreviations

MoC	Model of Computing	CS	Continuous System
VSIA	Virtual Socket Interface	DTS	Discrete Time System
SLDL	System Level Design Level	CES	Continuous Events System
CPS	cyber-physical systems	DES	Discrete Events System
DSP	Digital Signal Processing	DFG	Data Flow Graph
CSP	Communicating Sequential Process	CPN	Colored Petri Nets
PSM	Program State Machine	PN	Petri Nets
SDF	Synchronous Data Flow	CM	Continuous Model
CFG	Control Flow Graph	DE	Discrete Event
SPN	Stochastic Petri Net	FSM	Finite State Machine
KPN	Kahn Process Network	SR	Synchronous/Reactive
CFSM	Codesign Finite State Machine	FSMD	Finite State Machine with Datapath

References:

[1]. Axel J. Modeling Embedded Systems and socs/Concurrency and Time in MoC. Chapter (3, 4, 5, 6) Royal Institute of Technology. Stockholm, Sweden 2004.

[2]. Alberto S V. Quo Vadis SLD: Reasoning about Trends and Challenges of System-Level Design. Proceedings of the IEEE; 2007.

[3]. Gabor K. Unification or integration? The Challenge of Semantics in Heterogeneous Modeling Languages. Vanderbilt University. Nashville, TN 37235, USA 2014.

[4]. Arun K T. Principles of System Identification. https://doi.org/10.1201/9781315222509 . CRC Press 2018.

[5]. Bernard P Z, Alexandre M, Ernesto K. Theory of Modeling and Simulation/ Integration Discrete Event and Iterative System Computational Foundations; 3ed. Chapter 5. Elsevier 2019.

[6]. Ralf S. Special Session/ Virtual Soket Interace Alliance. FZI. SLD DWG. http://www.vsi.org.

[7]. Mark G, VSI Alliance. VSI System Level Design Model Taxonomy. Version 1. 1998

[8]. Richard Z. Embedded Systems Design and Verification. ISA Corporation; San Francisco, California, USA. Chapter 3. Industrial Information Technology Series, Taylor, 2009.

[9]. Adnan S, Ali H M, Khalid M.Specification and Modeling of HW/SW CO-Design for Heterogeneous Embedded Systems. Vol I, Proceedings of the World Congress on Engineering 2009.

[10]. M. Burginand G. Dodig-Crnkovic, "A taxonomy of computation and information architecture." *Proceedings of the 2015 European conference on software architecture workshops*. 2015.

[11]. P. A. Fishwick, "A taxonomy for simulation modeling based on programming language principles." IIE transactions 30.9 (1998): 811-820.

[12]. C. A. Furia, et al, "Modeling time in computing: a taxonomy and a comparative survey." *ACM Computing Surveys (CSUR)* 42.2 (2010): 1-59.

[13]. Surajit C, Tamal R, Samarjit S, Christian B V. Modelling and Simulation in Science, Technology and Engineering Mathematics. Springer 2019.

[14]. Felice B, Luciano L, Claudio P, Alberto S V. Modeling and Designing Heterogeneous Systems. Concurrency and Hardware Design, LNCS 2549, pp. 228–273, 2002; Springer-Verlag Berlin Heidelberg 2002.

[15]. Goran D P,Luis F, Nuno L, Zlata P. What is a Cyber-Physical System:Definitions and Models Spectrum. FME Transactions (2019) 47, 663-674.

[16]. Gabriel A W. Discrete-Event Modeling and Simulation/ A Practitioner's Approach. Chapter 2. Taylor & Francis Group 2009.

[17]. Daniel D G, Samat A, Andreas G, Gunar S. Embedded System Design/ Modeling, Synthesis and Verification. University of California, Irvine 2010.

[18]. Claudius Ptolemaeus, editor. System Design, Modeling, and Simulation using Ptolemy II. Ptolemy.org, 2014,

[19]. Richard Z. Embedded Systems Design and Verification. ISA Corporation; San Francisco, California, USA. Chapter 3. Industrial Information Technology Series, Taylor, 2009.

[20]. Marten L, Christian M, Soroush B, Edward A L. Toward a Lingua Franca for Deterministic Concurrent Systems. pp20-23.ACM 2021.

[21]. Edward A L. Models of Timed Systems. International Conference on Formal Modeling and Analysis of Timed Systems. pp 17–33. Springer 2018.

[22]. Alfred D, John G. Discrete sets definable in strong Expansions of ordered abelian groups.arxiv:2208.06929. //doi.org/10.48550/arXiv.2208.06929, 2022.
[23]. Edward. L. "The past, present and future of cyber-physical systems: A focus on models." *Sensors* 15.3 (2015): 4837-4869.
[24]. Spencer B, Ram D S, Eswaran S. Compositional Models for Complex Systems. https://doi.org/10.1016/B978-0-12-817636-8.00013-2 .Elsevier 2019.
[25]. Xiaojun L, Eleftherios M, Edward A L. Modeling Timed Concurrent Systems. University of California, Berkeley 2006.
[26]. Gabriel A W, Pieter J M. Discrete-Event Modeling and Simulation/Theory and Applications. Taylor & Francis Group 2011.
[27]. Steven C C. Applied Numerical Methods with MATLAB for Engineers and Scientists. McGraw-Hill 2018.
[28]. Ali U K. Ordinary Differential Equations for Engineers. Springer 2019.
[29]. Edward A L, Sanjit A S. Introduction to Embedded Systems - A Cyber-Physical Systems Approach. 2ed. Part I. University of California, Berkeley 2017.
[30]. Peter M. Embedded System Design, Foundations of Cyber-Physical Systems, and the Internet of Things. 2ed. Chapter(2). Springer 2018.
[31]. Mohammed M. A novel adaptive sampling algorithm for cyber-physical systems. International Review of Applied Sciences and Engineering 15, no. 2 (2024): 161-170.
[32]. Claudius Ptolemaeus, Editor: System Design, Modeling, and Simulation using Ptolemy II; Http://ptolemy.org/systems, 2014.