Time Task Scheduling for Simple and Proximate Time Model in Cyber-Physical Systems

Yuan FANG[1,2], Sian En OOI[1], Yuto LIM[1], and Yasuo TAN[1]

[1] Japan Advanced Institute of Science and Technology (JAIST),
1-1 Asahidai, Nomi, Ishikawa, 923-1292 JAPAN,
{yfang, sianen.ooi, ylim, ytan}@jaist.ac.jp,
WWW home page: http://www.jaist.ac.jp
[2] Dalian Polytechnic University (DPU),
No.1 Qinggongyuan, Dalian, Liaoning, CHINA

Abstract. Modeling and analysis play essential parts in a Cyber-Physical Systems (CPS) development, especially for the system of systems (SoS) in CPS applications. Many of today's proposed CPS models rely on multiple platforms. However, there are massive reusable components or modules in the different platform. And also, the model had to be modified to meet the new system requirements. Nevertheless, existing time model technologies deal with them, but it leads to a massive time consuming and high resource cost. There are two objectives in this paper. One is to propose a new simple and proximate time model (SPTimo) framework to the practical time model of hybrid system modeling and analysis. Another is to present a time task scheduling algorithm, mix time cost and deadline first (MTCDF) based on computation model in the SPTimo framework. Simulation results demonstrate that the MTCDF algorithm achieves the priority the scheduling of tasks with a time deadline, and match with optimal scheduling in time requirement and time cost.

Keywords: cyber-physical systems, time model, scheduling, deadline first, optimal scheduling index

1 Introduction

Cyber-Physical Systems (CPS) in [1] and [2] is usually defined as tight integration of computation, communication, and control with deep interaction between physical and cyber elements in which embedded devices, such as different sensors and actuators, are wireless or wired networked to sense, monitor and control the physical world. With CPS becoming more popular, many types of research have devoted to their development. The modeling and analysis play an essential part of the safety and mission critical system of systems (SoS) development in CPS. Researches in [3] have contributed to the modeling of CPS. Some of the modelings of the discrete and continuous behavior of heterogeneous systems has been developed recently, such as Ptolemy II [4] and Programming Temporally Integrated Distributed Embedded Systems (PTIDES)[5], those models for a deterministic modeling paradigm suitable for CPS application at any scale. All most of the current modeling structure is platform to platform, it follows discrete-event (DE) semantics. Furthermore, time is the first parameter because it is the interaction requires physical world and cyber world in CPS.

The model-driven development method first needs to solve the structure of the model, then to solve the task scheduling. Existing CPS modeling method is the procedure and initial process of the modeling are still highly complicated. Furthermore, time is very consuming because it requires a full and in-depth understanding of the details of the physical environments. Its the weakness which it is difficult, even impossible, to adapt some existing scheduling algorithms different systems or to different scheduling targets. When we design a scheduling algorithm for a newly developed system, it is difficult to benefit from the existing scheduling algorithms, which usually results in a high cost. For many existing CPS scheduling algorithms, to enhance them to support some practical requirement is difficult, even impossible. Due to system complexity, simple and proximate modeling purpose method for CPS application.

The objectives of this paper are to propose a simple and proximate time model (SPTimo) framework for CPS applications to meet the requirements of modeling and analysis with minimum cost and time firstly. This model based on a proximity method, which is used to model each component of sensing, fabric network, and actuating in the reality multiple platforms model as computation model, control model, and communication model, respectively. Second, Mixed Time Cost and Deadline-First (MTCDF) algorithm was proposed to addressed the problem of multi-programming scheduling in the SPTimo computation model. At last, the successful ratio and optimal scheduling index were shown in the simulation results.

2 Related Works

2.1 Modeling of CPS

CPS modeling requires portrayal of how interactions between the process and physical processes are calculated and how they behave when they are merged [5]. The model-based analysis provides a better understanding of CPS behavior, and model-driven design can improve design automation and reduce errors in refinement. Edward Lee proposed CPS is integrated computing power and physical process of the system[6], which uses embedded computers and networks to monitor the physical processing process, and with the feedback loop, physical process and computation process affect each other. For a system modeling, simulation and verification, computational science and control science have different ways. However, in the control science, the research on the physical world is often based on time, abstracting the system as a continuous time model, and time is the most critical factor in the model this will result in collisions and random failures in the interaction between the computational unit and the physical entity model. There are various interactive entities involved in CPS system. It can be a natural environment, building, machine, a physical device, human beings, etc. The situation can be real-time sensing part (such as the physical entity) can also be controlled.

The current reality model of CPS in Fig. 1, most of them are multiple platforms structure. There are sensors, actuators, computation units in the platform. If there is a *Service* need all the platforms 1, 2 and 3. The *Service* will be in computation time $\sum_{i=1}^{4} com_i$, and the all the delay is $\sum_{i=1}^{5} d_i$. There is two main problem with this model: *i)* the platform usually uses a single algorithm to schedule. For services with different time requirements, time efficiency is not optimal.*ii)* some computation time is repeated. From the perspective of the entire system, different computing capabilities have different effects on the time of service. Accumulating these effects together will cause the time and task of the next service to miss deadlines and cannot be implemented, which reduces the efficiency of the system.

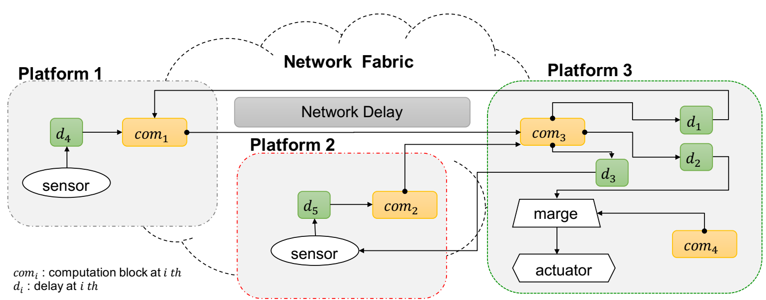


Fig. 1: General time model for CPS

2.2 Time Model

Discrete-Event (DE) is used to model time, discrete interactions between concurrent actors. [6] There *event* is in each communication, conceptually understood to be an instant message sent from one actor to another. The ***time model*** has encompassed the overall approach to handling time sequencing.

3 Simple and Proximate Time Model Frameworks

3.1 Structure of the SPTimo Framework

The SPTimo framework is illustrated in Fig. 2. There are three sub-models included in the frame. The first sub-model is ***Computation Model***. All the components with computational requirements from the different platform are approximated by this model. A service request requires connecting different platforms in a specific order. Each platform has different devices that complete the tasks. The operation generated by each device is called a *Time Task*. In a CPS service, time tasks are required to be completed before their deadlines. To satisfy this requirement, their required computation resources should be allocated to tasks at the right time. The allocated result is called *schedule*, while the allocating process is called *scheduling* which is conducted by a scheduler equipped in the system. The second sub-model is ***Control Model***. All the components of the approximate control operation are integrated into this model. The control model is mainly proposed in the control model, especially the feedback control algorithm, which includes online and offline forms. ***Communication Model*** follows the existing communication protocols. It mainly includes the network synchronization unit and the time offset of the SPTimo framework.

In fig. 1, there are four computation units, and they will feedback the result to the other parts. If the case in the SPTimo framework, there is one computation time and get the optimal schedule for one service.

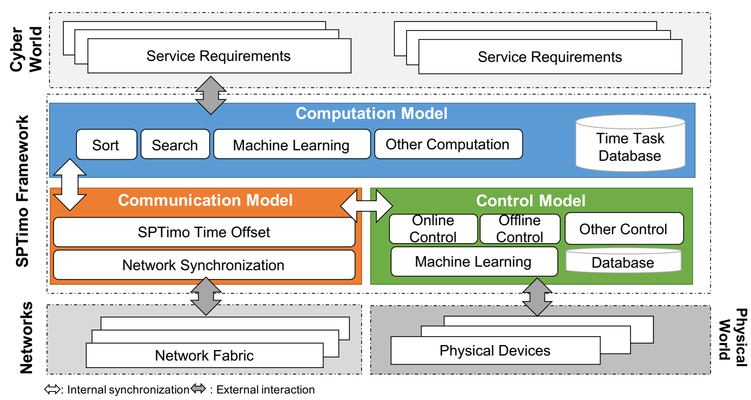


Fig. 2: SPTimo framework

3.2 Time Task Model

Definition 1 (Time Task Dependency Graph) A time task dependency graph is a directed non-cyclic graph. $G = (P, E)$, where P is a platform set, $E \subseteq P \times P$ is a dependency realation (edge) set, with $(p_i, p_j) \subset E$,$i \neq j$,where $p_i, p_j \subset P$. An edge (p_i, p_j) in the task dependency graph means platform p_i can start to execute only after platform p_i has been completed. $p_i \prec p_j$ is used to illustrate this dependency relation, and the this relation is transitive.

Definition 2 (Time Task) A time task is a multidimensional set $TT = (T, S, V)$. It includes a subset of time T, a subset of states S, and a subset of values V. The j time task of i platform is $T_{i,j} = \{A_{i,j}, E_{i,j}, D_{i,j}\}$, $A_{i,j}$ is the arrival time in a task scheduling, $E_{i,j}$ is the execute time, and $D_{i,j}$ is the deadline. Each task $S_{i,j} = \{s_{i,j}\}$, $s_{i,j}$ is the state of the elements. $V_{i,j}$ is value of the time task $T_{i,j}$ with state $S_{i,j}$. i is means the i time task, and j is means the j platform, $(i, j) \subseteq N$.

Definition 3 (Service) A service is defined by the operation of the platforms with the order of execution in the CPS application. $Service = (P_i \prec P_j)$. We assume that the execution time task $T_{i,j}$ inside the platform P_i are disordered. In order to meet the highest proportion of successful service execution, the time task sequence within the platform can be adjusted.

Eqn. (1) is used to calculate the waiting time of the time task. Equ. (2) for determining whether the time tasks can be executed.

$$W\left(_{i,j}\right) = A_{i,j} + \sum E\left(_{i',j'}\right) \tag{1}$$

where i', j' is means the total task before i, j.

$$D\left(_{i,j}\right) \geq W\left(_{i,j}\right) \tag{2}$$

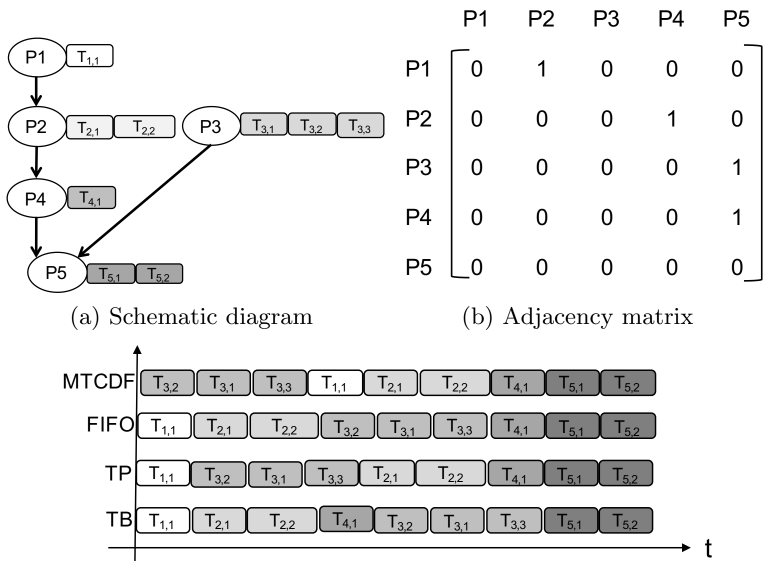
$\forall T\left(_{i,j}\right)$ satisfied the $Equation\,(2)$, the service entire time task is executable.

4 Time Task Scheduling

4.1 Scheduling Procedure

There are many scheduling algorithms used in various real-time systems. In this subsection, through an example described in Fig. 3, the performance of three widely used scheduling algorithms, Tree Based (TB), Task Proirity (TP), and First In First Out (FIFO) are studied. In the case, the lengths of the indivisible fragments in the tasks are shown the execute time length. The parameters of the platform time task are shown in Table 1. Schedules the time task with the direct way according to the following adjacency matrix. Each row corresponds to the starting point, and each column corresponds to the ending point. For example, $P_1 \prec P_2$, the intersection of the first row and the second column has a value of 1 in the adjacency matrix.

Dependency the adjacency matrix, fig. 3 (b), Table 2 shows that the four scheduling results.



(a) Schematic diagram (b) Adjacency matrix

(c) Scheduling algorithms results

Fig. 3: Example for scheduling

Table 1: Time task parameters

Platform	$T_{i,j}$	$A_{i,j}$	$E_{i,j}$	$D_{i,j}$
P1	$T_{1,1}$	0	1	5
P2	$T_{2,1}$	1	1	10
	$T_{2,2}$	1	4	10
P3	$T_{3,1}$	0	1	5
	$T_{3,2}$	0	1	1
	$T_{3,3}$	0	1	5
P4	$T_{4,1}$	2	1	10
P5	$T_{5,1}$	3	1	15
	$T_{5,2}$	3	1	15

Table 2: The scheduling result of example

Algorithm	*Service* Scheduling	$\sum_i^{\forall j} W\ (_{i,j})$	Successful Ratio	Unable to Meet Deadline
TB	$\{P_1 \prec P_2 \prec P_4 \prec P_3 \prec P_5\}$	44	0.6667	$T_{3,1}$, $T_{3,2}$, $T_{3,3}$
TP	$\{P_1 \prec P_3 \prec P_2 \prec P_4 \prec P_5\}$	35	0.8889	$T_{3,2}$
FIFO	$\{P_1 \prec P_2 \prec P_3 \prec P_4 \prec P_5\}$	44	0.6667	$T\ _{3,1}$, $T_{3,2}$, $T_{3,3}$
MTCDF	$\{P_3 \prec P_1 \prec P_2 \prec P_4 \prec P_5\}$	44	1.0000	None

Algorithm 1 Mixed Time Cost and Deadline First (MTCDF)

A *Service* is processing schedule defined by an adjacency matrix $|AJ\,(P_i \prec P_j)|$,P_i,P_j is platform, for all $i, j, k \subseteq N$
$S_{MTCDF} = AJ.\texttt{MTCDFSort}(T_{i,j})$
for all $1 \leq i \leq n$ **do**
  for all $1 \leq j \leq n$ **do**
    $sum_i = T_{i,j}^{deadline}$
    sort($T_{i,j}$) with deadline first
  end for
end for
for all $1 \leq i \leq n$ **do**
  if $sum_i <= sum_{i+1}.and.P_i.AJ == P_{i+1}.AJ$ **then**
    $S_{k++} = P_i$
  else
    $S_{k++} = P_{i+1}$
  end if
end for
Scheduling list is $S_{MTCDF} = list(S_k, 1 \leq k \leq n)$

4.2 Mixed Time Cost and Deadline-First (MTCDF) Algorithm

Dynamic scheduling of tasks in an overloaded real-time system was proposed by [7]. Cheng et al. propose SMT (*satisfiability modulo theories*)-based scheduling method [8]. Lim et al. propose time delay model for smart home [9]. Those pursues to focus on maximizing the total number of tasks that can be completed before their deadlines and time delay model with experiment. In the paper, we propose one navel scheduling algorithm, mixed time cost and deadline first (MTCDF) base the time task database of SPTimo Framework. This method is used to priorities the scheduling of tasks with a time deadline, and records the computation time.

Some assumptions were applied to the proposed scheduling algorithms, e.g., *1*) The requests for all time tasks for which strict deadlines exist are random; *2*) Time tasks are independent for each platform; *3*) Run-time for each time task is constant and does not vary with time.

According to the time task example, there are two main parts that affect the efficiency of the algorithm.*i*) The order of execution of the platforms in the same priority situation. *ii*) The time cost to compute the time task schedule in the deadline-first platform. Such as the scheduling $T_{3,2} \prec T_{3,1} \prec T_{3,3}$.
The relationship between deadline-first and time cost is defined by the following equation:

$$\mathit{\Gamma}_k = \mu \cdot \sum_{i=1}^{n} P_i^{deadline} + \nu \cdot \sum_{j=1}^{m} T_{i,j}^{timecost} \quad (3)$$

where $\mathit{\Gamma}$ is means the optimal scheduling index, k is the algorithm, μ is the coefficient of deadline-first part, ν is the coefficient of time cost part. $i, j, n, m \subseteq N$. The process of schedule synthesis is summarized in Alg. 1.

5 Simulation and Results

5.1 Simulation

In order to evaluate the successful ratio among the TP, TB, FIFO, and MTCDF scheduling algorithms in different λ and service including platforms' number. The input *Service* is generated according to the adjacency matrix AJ. There is one time task database, with 10 times number of platform random time tasks. The input functions random distribute the time tasks to platform from time task database. Each time task has three parameters, in the Def. (2). For each arrival time $A_{i,j}$ is generated according to Poisson distribution with arriving rate λ to every platform. In this simulation, the average inter arrival time of task is fixed at 100 ms per frame, where as the average inter arrival time of task varies from 10 to 1000 ms per frame. All the simulation results is collected using a 64bit Intel Core i7 CPU 2.4 GHz with 16GB of memory.

5.2 Results

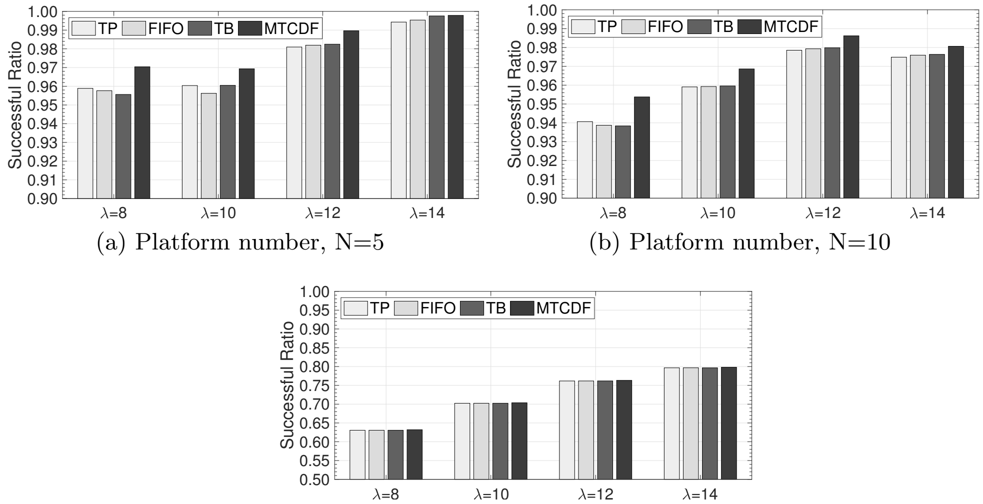
A service is performed by different time tasks of multiple platforms according to the scheduling result. In this simulation, the number of time tasks in the time task database is 10 times the number of platforms, and randomly matched to the platform as the service is generated. The arrival time and deadline of the time task are with the Poisson distribution. With $\lambda = \{8, 10, 12, 14\}$, the time task number is the the successful ratio and the optimal index Γ_k was observed. The results presented in Fig. 4 and Fig. 5 are averaged over 100 simulation runs with different random numbers.

Successful Ratio We have measured the effect of successful ratio on different platform number with different λ. As seen from Fig. 4 (a), when the λ is changed, the successful ratio are increased. Especially, more MTCDF can improve more succesful ratio in the same λ comparing with TP, TB, and FIFO. As Fig. 4 (b), we show the successful ratio with 10 platform. In Fig. 4 (c), we show the 100 platform. As the result, the successful ratio is over 0.648. And the four scheduling algorithms results are very close.

Optimal Scheduling Index The optimal scheduling index refers to the more appropriate scheduling was found. We use the min-max normalization method to process the results. In Fig. 5 shows the optimal scheduling index Γ with different λ different platform number N. The results show that the optimal scheduling index can give to optimization scheduling results, with meeting the time requirements first and time cost.

6 Conclusion and Future Work

In this paper, we have addressed the platform to platform scheduling issues of the CPS time model. We define a new time model the SPTimo framework in

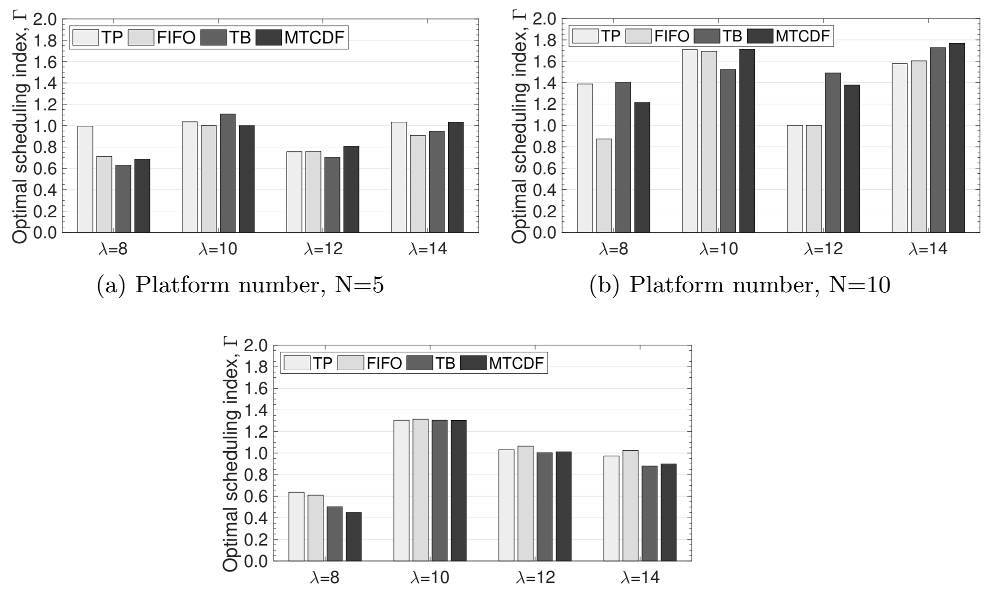


(a) Platform number, N=5

(b) Platform number, N=10

(c) Platform number, N=100

Fig. 4: Successful ratio



(a) Platform number, N=5

(b) Platform number, N=10

(c) Platform number, N=100

Fig. 5: Optimal scheduling index, Γ

CPS. The contributions of the SPTimo framework are it reduce the repetition rate of the components and the elements in the same model can be approximated, to enhance the efficiency of computation and control. According to the SPTimo framework design principle, we proposed the MTCDF algorithm for computation model scheduling algorithm. The simulation results show that the MTCDF time task scheduling method can improve service success rate. The SPTimo framework can generality of the matching optimal scheduling index for time requirement service.

As future work, we will extended the machine learning algorithm of computation model in SPTimo framework. Then we will integrate the control model to SPTimo framework.

References

1. E. A. Lee, "CPS foundations," *ACM/IEEE Des. Automation Conf. (DAC)*, pp.737742, 2010.
2. B. Sean, S. Shankar, A. John, S. Roundtable, "Reliability of embedded and cyber-physical systems," *IEEE Security and Privacy*, vol.8,no.5 : pp27-32, 2010
3. J. C. Eidson, E. A. Lee, S. Matic, S. A. Seshia and J. Zou, "Distributed Real-Time Software for CyberPhysical Systems," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 45-59, Jan. 2012.
4. P. Derler, E.A. Lee, and A.S. Vincentelli, "Modeling cyber-physical systems," *Proceedings of the IEEE*, vol.100, no.1, pp.1328, 2012.
5. P. Derler,T.H. Feng,E.A. Lee, et al. "PTIDES: A programming model for distributed real-time embedded systems," *California Univ Berkeley Dept of Electrical Engineering and Computer Science*, 2008.
6. E. A. Lee, "The Past, Present and Future of Cyber-Physical Systems: A Focus on Models," *Sensors (Basel, Switzerland)* , vol.15, no.3, pp. 4837-4869. PMC. Web, 2017.
7. A. Marchand, M. Chetto, "Dynamic scheduling of periodic skippable tasks in an overloaded real-time system," *IEEE/ACS International Conference on Computer Systems and Applications*, Doha, Qatar, pp. 456464, Apr. 2008.
8. Z. Cheng, H. Zhang,Y. Tan, et al. "SMT-Based Scheduling for Overloaded Real-Time Systems," *IEICE Transactions on Information and Systems*, 2017, 100(5): 1055-1066.
9. Y. Lim, Y. Tan. "Time Delay Modeling for Energy Efficient Thermal Comfort Control System in Smart Home Environment," *International Conference on Computational Science and Technology.* Springer, Singapore, 2017: 42-52.
10. P. Palensky, E. Widl, A. Elsheikh. "Simulating cyber-physical energy systems: Challenges, tools and methods," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, 2014, 44(3): 318-326.
11. A. Murugesan, S. Rayadurgam, M.W. Whalen, et al. "Design Considerations for Modeling Modes in Cyber-Physical Systems," *IEEE Design & Test*, 2015, 32(5): 66-73.
12. M.D. Ilic, L. Xie, U.A. Khan, et al. "Modeling of future cyberphysical energy systems for distributed sensing and control," *IEEE Transactions on Systems, Man, and Cybernetics-Part A: Systems and Humans*, 2010, 40(4): 825-838.