Chapter 5
Concurrency and Synchronization in Structured Cyber Physical Systems

Jitender Grover and Ram Murthy Garimella

Abstract Cyber Physical System (CPS) involves the integration of the Cyber World and the Physical World. Structured Cyber-Physical Systems such as Wireless Sensor Networks (WSNs), Medical Monitoring, Smart Grids, Process Control Systems, Robotics Systems, Auto-Pilot Avionics, etc. are subjected to intense research and development. Efforts are underway to build various special purpose CPSs, without any disciplined effort to innovate theoretical principles required in their analysis and design. In the past few years, control theorists began to intensify research efforts to model and analyse special purpose, structured CPSs. When Concurrent Cyber Physical Systems evolve on multiple time scales, synchronizing such systems is an important problem. The literature on dynamical systems shows that the modeling of concurrent systems received limited interest. It is clear that if concurrent CPSs need to be analysed then taking Concurrency and Synchronization into account is essential. This chapter represents one interesting solution to this problem, i.e. Tensor State Space Representation (TSSR) for modeling and analysis of Concurrent Cyber Physical Systems. Furthermore, the synchronization and temporal semantics are absolutely essential for the control of CPSs evolving on Multiple Time Scales. Thus, the concepts for handling concurrency and time-scales are inevitable in the design and implementation of Cyber Physical Systems. This chapter basically presents an analytical approach to design, monitor, and maintain dynamical concurrent systems. Along with Concurrency and Synchronization, importance and management of Temporal Semantics are also presented in this chapter. Effective methods like Temporal Division of Labor and Edge Computing are proposed to deal with the issue of sensors' data fusion in multi-level servers. An Agent-based Framework is proposed for Reliable and Fault-tolerant CPS architecture. The type of modeling and analysis presented in the chapter will help in designing and implementing linear and distributed Cyber Physical Systems in a better way.

J. Grover (✉) · R. M. Garimella
Signal Processing and Communication Research Center (SPCRC), International Institute of Information Technology, Hyderabad, India
e-mail: jitender.grover@research.iiit.ac.in; rammurthy@iiit.ac.in

S. Guo, D. Zeng (eds.), *Cyber-Physical Systems: Architecture, Security and Application*, EAI/Springer Innovations in Communication and Computing,
https://doi.org/10.1007/978-3-319-92564-6_5

5.1 Introduction

Cyber Physical System (CPS) [1] involves the integration of the Cyber World and the Physical World. The evolution of Cyber Physical Systems began with the embedded systems [2]. Embedded systems are information processing systems that are embedded [3] into a larger product like machines or devices [4, 5]. Some examples of embedded systems can be found in the automotive sector when it comes to safety systems like anti-lock braking system (ABS) to regulate the brake force or the automatic four-wheel drive to increase traction or the airbag system that has a sensor that notices the crash and an actuator triggers the release of the airbag. Another example can be the traffic lights where pedestrians press a button to demand green light. So the term embedded system is used for a hardware and software system which is connected with the outside world through sensors and actuators to handle a distinct task [6, 7]. Embedded systems are followed by networked embedded systems. In the networked embedded system, a number of embedded systems get connected with each other and integrated into a wider context [8]. For example, a car, for instance, can be seen as an integrated control and information system where ABS, climate control, gearbox, speed control information and the motor control get connected to a system. Another example would be autonomous aviation. When the pilot switches on the autopilot, then the engine, the position and the speed of the plane need to be controlled. Also the actuators trigger adjustments to meet the settings of the autopilot [9, 10]. These early form of Cyber Physical Systems can be seen as networked embedded systems that are connected with each other through the internet. In this way, the physical world and the virtual world are merging as shown in Fig. 5.1. That's why they are called Cyber Physical Systems [11]. So, for example, cars in a city which can be seen as embedded systems share physical data like speed or distance with other cars through the internet, we could use these data to make road transport much more intelligent. It would be possible to improve traffic flow or to reduce accidents.

Cyber Physical Systems consist of physical components [12] and cyber components as shown in Fig. 5.1, that's why we call them cyber physical [13]. Actually, all Cyber Physical Systems are based on an information processing computer system which is embedded into a product like in a car, plane or other devices. These embedded systems do not stand alone anymore, they share their data via a communication network like the Internet with cloud computing [14]. This way the data from many embedded systems can be collected and processed in the cloud [15]. Connected embedded systems can be controlled decentralized via computational unit where data can be processed automatically or by the human–computer interface.

CPSs and IoTs have become the technology of future [16]. It is the base of Industry 4.0. The expansion of connected CPSs with us over the years is shown in Fig. 5.2. In 2005, when Mobile Era started, humans were connected to only/mostly three devices like Mobile, Computer, and Tablet through the Internet or other technologies.

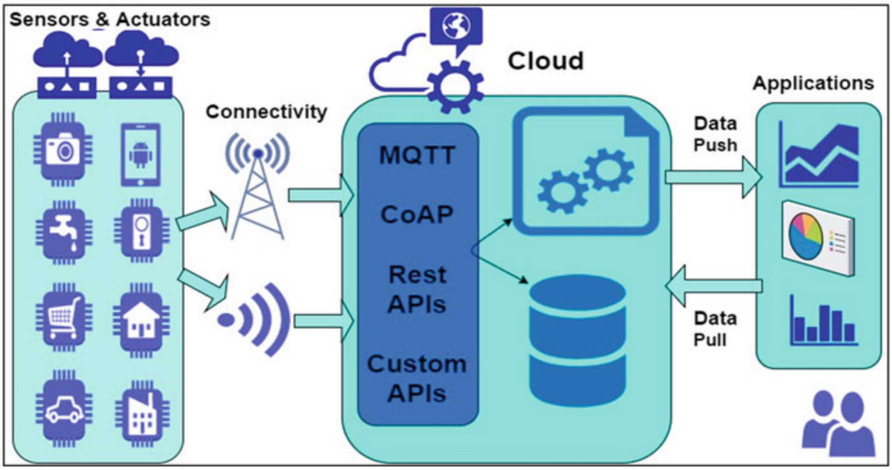
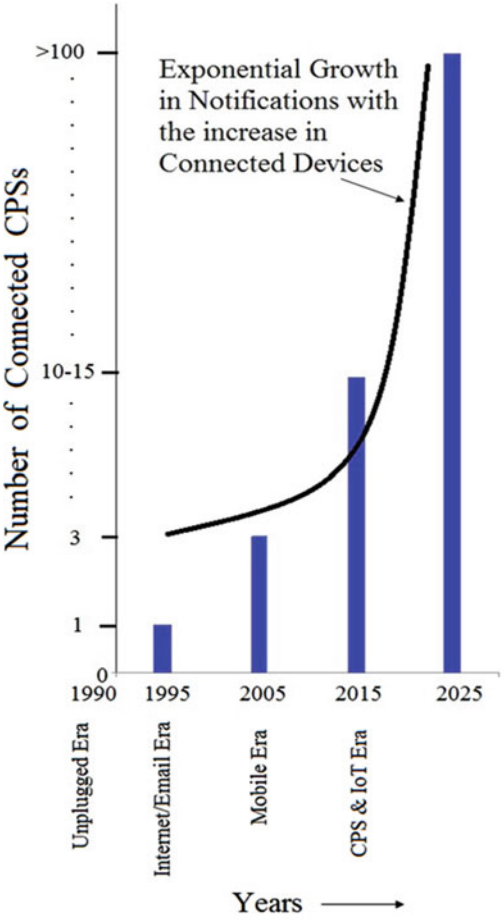


Fig. 5.1 IoT-Cloud infrastructure

In the IoT-Era, started in 2015, we own as many as 10–15 electronic devices ranging from wearables to non-wearables. Now if we go towards future, there can be more than 100 connected and concurrent Cyber Physical Systems and all over the world 100 Billion concurrent Cyber Physical Systems & sub-systems can be connected in the year 2025 and will keep on increasing over the years. There is a high probability that all the sensor devices won't work individually and will be dependent on other devices as well. For the sake of simplicity, we can consider any number of examples of a big Cyber Physical System like Airplane or Factory, where the Cyber Physical System is the combination of a lot of sub-systems or sensors. All the sensors/sub-systems need to work in tandem to produce an appropriate input for the control server. If there is any error in that input, then the decision taken by the control system will go wrong and can create any big damage to the whole system. Apart from errors, if the sensed data is not synchronized and properly interleaved considering time as a prime factor then it may create a wrong input for the processing systems and feedback results can be severe.

There are many research areas and challenges related to those areas that are associated with Cyber Physical Systems [17, 18]. Some broad areas are shown in Fig. 5.3. Sensors Network (Wired or Wireless), their networking, writing software on different levels, controlling sensor devices with actuators, building information theories for data manipulations or aggregations, and last but not the least writing formal methods in computer science to model and analyse the CPSs in a better way [19] are some major research areas and challenges related to CPSs [20]. All the areas are merged into each other and research required careful effort while solving one issue. For example, while solving issues related to ground level implementation, issues like coordination between different sub-systems [13, 21], synchronization between different sub-systems [22], aggregation of data depending upon the capacity of network, requirement of controlling through feedback loops,

Fig. 5.2 Expansion of connected devices



delay-tolerance of system in case of real-time system [23] must be taken into consideration as failing in any of the other related part can make the whole system collapse [5]. From the next section onwards, many of the fine-tuned research issues and solutions will be discussed through analytical modeling and intuitive ideas.

For Cyber Physical Systems, there are many fields of applications [24]. Cyber Physical Systems can be used for the energy provision in manufacturing to optimize production processes, for assisted living and also in healthcare, etc. Structured Cyber Physical Systems such as Wireless Sensor Networks (WSNs), Medical Monitoring, Smart Grids, Process Control Systems, Robotics Systems, Auto-Pilot Avionics, etc. [25, 26] are subjected to intense research and development. Efforts are underway to build various special purpose CPSs, without any disciplined effort to innovate theoretical principles required in their analysis and design. In the past few years, control theorists began to intensify research efforts to model and analyse special purpose, structured CPSs [27]. When CONCURRENT Cyber

Fig. 5.3 Formal areas and challenge in CPSs

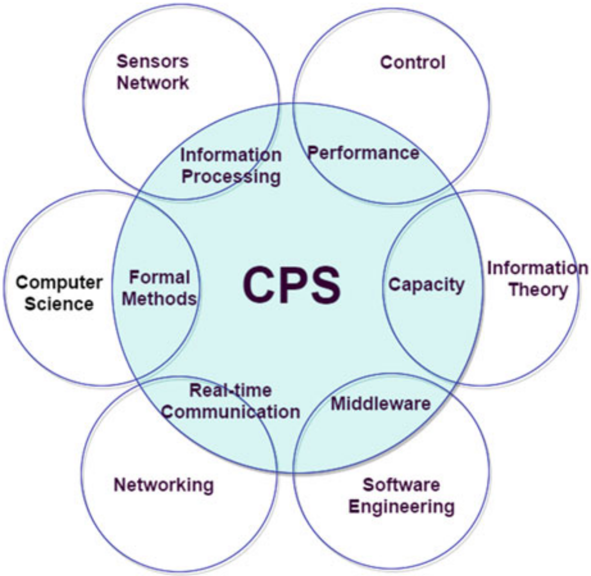
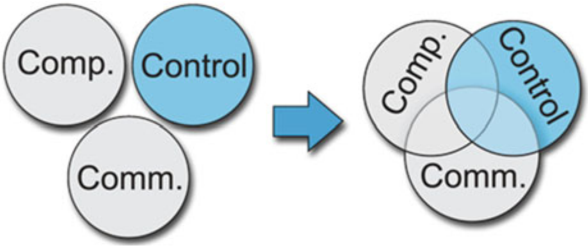


Fig. 5.4 Joint design of communication, control and compute systems in CPSs



Physical Systems evolve on multiple time scales, synchronizing such systems is an important problem. One interesting solution to this problem is Tensor State Space Representation (TSSR) for modeling and analysis of CONCURRENT Cyber Physical Systems [28]. The literature on dynamical systems [29], modeling of concurrent systems received limited interest. It is clear that if concurrent CPSs need to be analysed, taking CONCURRENCY & SYNCHRONIZATION into account is essential.

The analysis of concurrent and structured CPSs requires tractable models of interconnected dynamical systems [30, 31]. Thus, theoretical efforts (modeling and analysis) related to CPSs are actively pursued by many researchers [32]. Also, design and implementation of prototype CPS are underway at many institutions across the world [33]. The unification of CONTROL, COMMUNICATION and COMPUTATION (Fig. 5.4) functions in multi-dimensional linear dynamical systems have been discovered and formalized. Specifically, optimal control tensors, optimal codeword tensors, optimal logic function tensors are synthesized as the stable states of multi-dimensional neural networks. Furthermore, the synchronization and temporal semantics are absolutely essential for the control of CPSs evolving on Multiple Time Scales. Thus, the concepts for handling concurrency and time scales are inevitable in the design and implementation of Cyber Physical Systems. The unified theory provides new insights into the global research activity for the design and analysis of concurrent multi-dimensional linear systems evolving on multiple timescales. Also, it will help the research organizations and industries to build special case CPSs.

From systems' viewpoint, we have an interconnected network of linear/nonlinear/hybrid/periodic/aperiodic/distributed/hierarchical dynamical CPSs evolving on multiple time scales [34] (with several feedback loops from control theory viewpoint) [35]. These dynamical systems could be loosely coupled (Buildings' Sensor Networks) or tightly coupled (Wireless Body Area Networks). Some approaches to representing [33] concurrent linear dynamical systems and synchronization of such systems evolving on multiple time scales were discussed in the next section. For instance, a Wireless Body Area Network (monitoring temperature, blood pressure, heart's electrical activity (ECG), brain's electrical activity (EKG), ultrasound, brain imaging-CAT/MRI/FMRI) involves current periodic/aperiodic, linear/nonlinear dynamical systems [36] evolving on multiple time scales. Thus, such a network of Cyber Physical Systems naturally presents important challenges for Smart Hospitals [37, 38]. It is realized that in most such Cyber Physical Systems, in-network distributed computation takes place. Decision fusion takes place at various levels of hierarchical network of Cyber Physical Systems.

Analytical approach to design, monitor, maintain such dynamical systems exists to some extent in the available literature [39]. But, several interesting new challenges naturally arise and need extensive research efforts. For instance, fuzzy logic-based decision support systems present new research challenges. Coming sections in the chapter are going to address some of the important issues related to Concurrency and Synchronization in Structured Cyber Physical Systems.

5.2 Concurrent Cyber Physical Systems: Modeling

5.2.1 Concurrency in Cyber Physical Systems

Concurrency is the phenomenon of things to happen in parallel in a system [40]. Cyber Physical System can have many subsystems running in parallel. If they are non-interactive, then there is no need for concurrency control. But if they interact and depend on each other, then serious concurrency control is required. In most Cyber Physical Systems, several physical/physiological processes are concurrently evolving in time on different timescales. For instance, Wireless Body Area Networks (W-BAN) (e.g. sensors monitoring temperature, blood pressure, glucose level, ECG, EKG, etc.) are evolving on different time scales and the information fusion occurs on different time scales. Coordinating and analysing data generated by such concurrent Cyber Physical Systems is extremely important. Research related to Concurrency Control in Computer Science will provide useful insight into designing and coordinating Concurrent CPSs. It should be kept in mind that several new research issues naturally arise in designing and analysing Concurrent CPSs.

Classifications of Concurrent CPS:

1. Discrete/Continuous Time
2. 1-D/M-D: Input–Output state space representation

Section 5.2.2 explains the solution for maintaining concurrency in Cyber Physical Systems and Sub-systems based on these classifications.

5.2.2 Coordination and Maintenance of Concurrent Cyber Physical Systems

The basic characteristic of Cyber Physical and Embedded Systems is dependency. The key reason is that whatever we do in our information processing, it has an immediate impact on the physical environment. So in case, we do something wrong, we might cause harm to some objects, we might cause harm to people, and that has to be avoided under all circumstances. So, it's not like the office automation where in case there is a problem with some program, we just start the computer again. We have to make sure there will be no harm caused due to CPS.

Cyber Physical and Embedded Systems must be dependable and this comes in a variety of flavours and in a variety of aspects. One of the important aspects is the maintainability which takes into account that systems might fail, and if they fail it should be possible to repair these systems and it shouldn't take too long. So, therefore, we are defining maintainability where maintainability is defined as the probability of a system working correctly t time units after an error occurred. Then as a third aspect, we have availability. Availability is the probability of a system working at a particular time. So, for example, let's suppose that we have a system which very predictively works for 999 h and then fails for 1 h and then works again for 999 h and then fails again. So, such a system would have an availability of 99.9%.

Coordination is must in [41] Concurrent Cyber Physical Systems to make them maintainable and available 24 $\times$ 7. If the concurrent systems or sub-systems lack coordination between them, then the probability of failure increases and on the other hand probability of availability decreases.

5.2.3 Design Issues: Concurrent CPSs

Digital integrated circuits are used to realize Cyber Physical System which can be a combinational, sequential or hybrid circuit as per requirement. Combinational integrated circuits can have processing delay at each gate and propagation delay as well. Depending upon the application and its tolerable delay, the design of the system can be verified for meeting the daily requirement of the system. In the sequential Integrated circuit as well, the delay can be calculated at design time and it can be checked that it is meeting the delayed threshold or not? In the sequential circuit, the current output depends upon the past Inputs and outputs and finite state machine associated with such circuits help in finding average tolerable delay. For meeting the requirement of delay and other constants analytical or simulation tools should be used and processor competition power should also be taken into consideration. So we require a tool for digital circuits' performance prediction.

From the present Cyber Physical Systems' implementation, it can be easily observed that temporal semantics are not taken into consideration while designing the software system for any CPS. Most of the available processors run the procedure sequentially so to achieve parallelism in concurrent Cyber Physical Sub-systems, we require careful interleaving of different procedures together [42]. Having multiple hardware processes makes the concurrency task easy but it makes interactions very complex. Involvement of several layers of software, communication between running tasks, separate timing and clock in different processes always make it hard to achieve concurrency on the hardware level. So it's always better that we do this task in software. In software subsystem, it is required to calculate the worst-case execution time. It gives an idea that the system can meet the critical real-time deadlines while designing the Cyber Physical System.

Further, in this section, we are going to discuss modeling method for linearly interacting concurrent Cyber Physical Systems in one/two/multi-dimensions. It is done by stacking the states of concurrent Cyber Physical Systems in different dimensions. Modeling Method can also be generalized for nonlinear concurrent CPSs as well.

5.2.4 Modeling Linear Concurrent CPS

Equation (5.1) represents the concurrent physical processes happening at the same time. Their state is represented as a scalar function of time.

$$\{p_i(t) = 1 \leq i \leq N\} \tag{5.1}$$

Equation (5.2) represents these physical processes as an N-dimensional vector. It can be seen as the state of all the concurrent coupled scalar valued processes. It is advantageous in examining multiple inputs and outputs in a linear or nonlinear, time-varying or invariant system.

$$P(t) = \begin{bmatrix} p_1(t) \\ p_2(t) \\ \vdots \\ p_{N-1}(t) \\ p_N(t) \end{bmatrix} \tag{5.2}$$

If input of systems is represented as $I(t)$, then

$$\frac{d}{dt} P(t) = f_1 \left(P, I, t\right) \tag{5.3}$$

$$Q(t) = f_2 \left(P, I, t\right) \tag{5.4}$$

Equations (5.5) and (5.6) represent the concurrent systems when they are time-varying and Eqs. (5.7) and (5.8) represent when they are time invariant.

$$\frac{d}{dt} P(t) = X_1(t)\ P(t) + X_2(t)\ I(t) \tag{5.5}$$

$$Q(t) = X_3(t)\ P(t) + X_4(t)\ I(t) \tag{5.6}$$

$$\frac{d}{dt} P(t) = X_1\ P(t) + X_2\ I(t) \tag{5.7}$$

$$Q(t) = X_3\ P(t) + X_4\ I(t) \tag{5.8}$$

For the linearly coupled systems, systems are represented as a vector-valued state rather than scalar valued state. $\bar{M}(t)$ (Eq. 5.9) is the *NxN* state matrix of coupled concurrent systems. Equation (5.10) and $\bar{A}(t)$ given in Equation (5.11) is the output matrix linearly coupled concurrent system. Similarly Eqs. (5.12) and (5.13) show the same for time invariant concurrent systems. Here, concurrent processes' state is shown as vector value. The linear coupling matrices X_1, X_2, X_3, X_4 are homogeneous for $\bar{M}(t)$.

$$\bar{M}(t) = [S_1(t) : S_2(t) : S_N(t)] \tag{5.9}$$

$$\frac{d}{dt} \bar{M}(t) = X_1(t)\ \bar{M}(t) + X_2(t)\ \bar{I}(t) \tag{5.10}$$

$$\bar{A}(t) = X_3(t)\ \bar{M}(t) + X_4(t)\ \bar{I}(t) \tag{5.11}$$

$$\frac{d}{dt}\bar{M}(t) = X_1\bar{M}(t) + X_2\ \bar{I}(t) \tag{5.12}$$

$$\bar{A}(t) = X_3\bar{M}(t) + X_4\ \bar{I}(t) \tag{5.13}$$

Equations (5.14) and (5.15) show the state space representation of discrete time concurrent Cyber Physical Systems/Sub-systems. Same way state space representation of time-invariant Cyber Physical Systems in discrete time can be generated.

$$\bar{M}\ (x+1) = X_1(x)\ \bar{M}(x) + X_2(x)\ \bar{I}(x) \tag{5.14}$$

$$\bar{A}(n) = X_3(x)\ \bar{M}(x) + X_4(x)\ \bar{I}(x) \tag{5.15}$$

Instead of coupling matrices, 3D arrays can also be used to the present homogeneous linear concurrent systems rather than using coupling matrices $\{X_1, X_2, X_3, X_4\}$. Outer product computation using tensor “X” and “S” is given below. It will generate a 5D array “R”.

$$X_{i_1,i_2,i_3}.S_{j_1,j_2} = R_{k_1,k_2,\ k_3,k_4,k_5}$$

Equations (5.16) and (5.17) show the state space representation of heterogeneous linear Concurrent Cyber Physical Systems. They are required to be represented using multi-dimensions rather than single dimension so a matrix or a 3D array or a multidimensional array (tensor) is used to represent such states. Systems can be in discrete or continuous time as well. Here discrete time is taken into consideration.

$$S\ (x+1) = X_1(x)\ S(x) + X_2(x)\ I(x) \tag{5.16}$$

$$T(n) = X_3(x)\ S(x) + X_4(x)\ I(x) \tag{5.17}$$

For multidimensional linear Cyber Physical Systems, Eqs. (5.18) and (5.19) are evolved as shown below. Here $X_1(x)$ is a state coupling multidimensional tensor presented with the order 2r. $S(x+1)$ is the system state at ‘$x+1$’ discrete time index. Similarly $S(x)$ is the state at ‘x’ time index. $X_2(x)$ is the input coupling multidimensional tensor with the order represented as ‘$r+1$’. $T(x)$ is the multidimensional output tensor with the order represented as ‘x ‘. $I(x)$ is the varying multidimensional inputs answer with the order ‘l’. $X_3(x)$ and $X_4(x)$ are the multidimensional tensors with the orders ‘$s+r$’ and ‘$s+1$’ respectively, used as tenses to the output dynamics.

$$S_{i_1,\ldots,i_3}(x+1) = X_{(i_1,\ldots,i_r\,j_1,\ldots,j_r)}(x).S_{j_1,\ldots.,j_r}(x) + X_{(i_1,\ldots,i,\,j_1,\ldots,j_l)}(x).I_{j_1,\ldots.,j_l}(x) \tag{5.18}$$

$$T_{q_1,\ldots,q_3}(x+1) = X_{(q_1,\ldots,q_s\,j_1,\ldots,j_r)}(x).S_{j_1,\ldots.,j_r}(x) + X_{(q_1,\ldots,q_s\,j_1,\ldots,j_l)}(x).I_{j_1,\ldots.,j_l}(x) \tag{5.19}$$

5.3 Synchronization of Concurrent Cyber Physical Systems

5.3.1 Networked Cyber Physical Systems: Synchronization

The approach for synchronizing the Cyber Physical Systems discussed earlier requires complete knowledge of system dynamics. In many [43] CPS, the dynamics are not completely/perfectly known. For synchronization of such systems, the following approach could be utilized.

We need to estimate the maximum delay involved in knowing the output for worst case input (it should be noted that the system could be implemented in software) [44, 45]. The idea is best illustrated by a parallel connection of software/hardware systems. The systems require processing delays (for worst case input) which are all not necessarily equal.

Hence, the system to which the outputs of parallel systems constitute input must receive all the inputs in a synchronized manner. Thus, as shown in Fig. 5.5, delay buffers are necessary. In general, based on the network topology, synchronization is achieved by optimally buffering/delaying the inputs/outputs in an intelligent manner [46, 47]. Such an approach is already utilized in some robotics applications, for instance, Critical Path Method (CPM) is well known. Synchronization of CPS should be concentrated on the following classifications:

- Replicated Decoupled/non-interacting CPSs
- Replicated Coupled/interacting CPSs

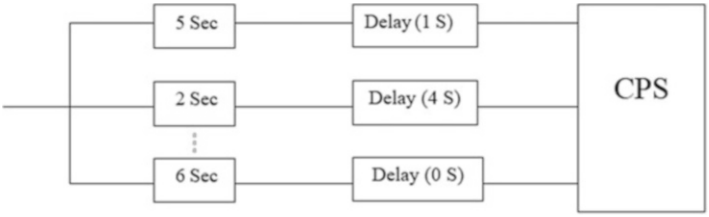


Fig. 5.5 Delay buffers in parallel systems

5.3.2 Clock's Synchronization: Multidimensional CPSs

As most of the physical processes in Cyber Physical Systems happen at multiple time scales [48, 49] the mark distributed coupled linear CPSs on different time scales [50]. Representation of search Cyber Physical Systems and their respective States can be represented as given in Eqs. (5.20), (5.21), and (5.22).

$$\begin{bmatrix} p_1(x) \\ p_2(x) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} p_1\left(\frac{x}{t_1}\right) \\ p_2\left(\frac{x}{t_2}\right) \end{bmatrix} + \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} q_1\left(\frac{x}{u_1}\right) \\ q_2\left(\frac{x}{u_2}\right) \end{bmatrix} \tag{5.20}$$

Dynamics of linearly interacting concurrent Cyber Physical Systems on multiple time scales are represented in Eq. (5.20), where $\{t_1, t_2, u_1, u_2\}$ is a scalar value related to discrete time and happening on separate time scales. The value is >1 and they are coupled.

$$\text{LCM } \{t_1, t_2\} = u \text{ and } x' = \frac{x}{u} \tag{5.21}$$

LCM of $\{t_1, t_2\}$ is calculated in Eq. (5.21).

$$\begin{bmatrix} p_1\left(ux'\right) \\ p_2\left(ux'\right) \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} \\ r_{21} & r_{22} \end{bmatrix} \begin{bmatrix} p_1\left(f_1x'\right) \\ p_2\left(f_2x'\right) \end{bmatrix} + \begin{bmatrix} s_{11} & s_{12} \\ s_{21} & s_{22} \end{bmatrix} \begin{bmatrix} q_1\left(f_1x'\right) \\ q_2\left(f_2x'\right) \end{bmatrix} \tag{5.22}$$

After applying Eq. (5.21) to Eq. (5.20), we got Eq. (5.22). A common global clock can be defined using LCM of $\{u, t_1, t_2\}$. It solves the problem of synchronization in Cyber Physical System by defining the system on a single time scale.

5.4 Temporal Semantics: Design of Cyber Physical Systems

Design issues related to embedded systems naturally provide insights into the choices of hardware and/or software systems employed in CPS [51, 52]. But various emerging CPSs naturally require time-critical guarantees on the processing of the data generated by the Concurrent Cyber Physical Systems evolving on different time scales [53]. The research related to REAL TIME OPERATING SYSTEMS (RTOS) will provide interesting insights into providing time-critical guarantees on results generated by various sub-systems in CPSs. Thus, the design of SOFTWARE SYSTEMS must necessarily ensure that temporal semantics [54] are taken into account. Existing programming abstractions (paradigms) must necessarily be redesigned to ensure that real-time performance guarantees are provided.

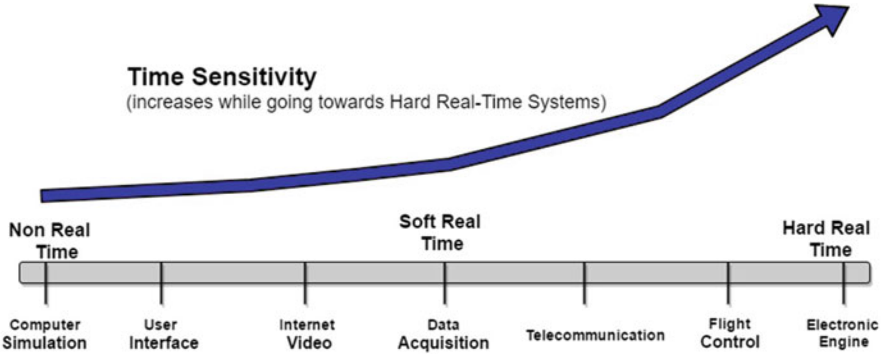


Fig. 5.6 Real-time and non-real-time CPSs

5.4.1 Classification of Cyber Physical Systems (CPSs)

One possible classification of Cyber Physical Systems (CPS) is [55] (Fig. 5.6)

- Real-Time CPS and
- Non-Real-Time CPS

A *hard real-time system* is one way or the computation has no value whatsoever if the time constraint is not met [56, 57]. Examples are control systems for industrial processes, Air Traffic control, and vehicles' subsystem control, etc. [58, 59]. The *soft real-time system* is the one having the property of timeliness of a computation where the value of computation decreases according to its tardiness. Examples are data acquisition systems, telecommunication, internet video, VoIP, etc. *Non-real-time systems* are the one having no time deadline at all to complete any task. If there is one, then failing to meet that deadline doesn't affect the system at all. One example is a computer simulation.

Considering Real-Time CPSs, deadlines are of two types [60]:

- Those which can be met with software/hardware resources.
- Those which can't be met. This can happen and there is no hope, e.g. Disaster.

Since any CPS involves hardware and software sub-systems [61, 62], it is essential that these sub-systems meet the time-critical deadlines. Otherwise, such real-time CPS will be unable to meet the performance guarantees essential for the operation of such systems.

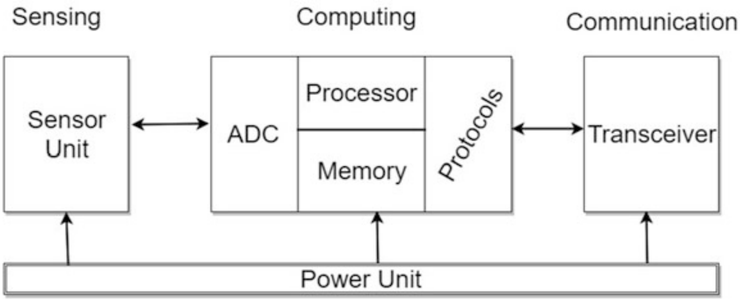


Fig. 5.7 Sensor node architecture

5.4.2 Real-Time CPSs: Temporal Semantics

5.4.2.1 Goal: Optimal Utilization of Software/Hardware Resources to Meet the Time-Critical Deadline

We first consider hardware subsystems. Since any CPS involves the integration of cyber world and physical world, most of the time a sensor/transducer (wired or wireless) is essential. Most sophisticated sensors have (Fig. 5.7) (a) Transducer; (b) Communication unit (transceiver in the case of wireless sensors), (c) Computation unit, (d) Analog-to-Digital Converter, (e) power unit, etc.

The current state of technology determines the speed of A-to-D converter. It imposes units on how fast digital signal is generated from the analog signal. Such technological limitations must be kept in mind when a CPS is designed. In effect, only currently realizable (technology wise) time critical deadlines can be met in current CPS. It should be noted that through "intelligent" utilization of available resources (e.g. time, memory, processors, etc.), sometimes critical deadline can be met. We illustrate an innovative idea which enables meeting temporal deadlines related to computation at the sensor.

We first state the problem of time-critical computation problem [63, 64] at a sensor/embedded system and how it can be solved through intelligent computation. Consider a temperature sensor monitoring a phenomenon and is required to locally compute a function such as mean, min, max, etc. Specifically, minimum fusion function needs to be computed within a "Critical Time", e.g. T_0. It is clear that as the number of sensed samples increases, the computation time with existing computation hardware proportionately increases. We propose the idea of TEMPORAL DIVISION OF LABOUR (Fig. 5.8), i.e. as and when the sensed temperature data becomes available locally the fusion function is immediately computed and stored in memory (so that a large number of samples are not accumulated). Thus, proactive intermediate computation of fusion function and storage of local fusion function values can be utilized to efficiently compute the fusion function over a larger timescale. This approach enables the sensor to meet the critical time deadline. In this case, "efficient" utilization of computing resources potentially enables the deadline to be met.

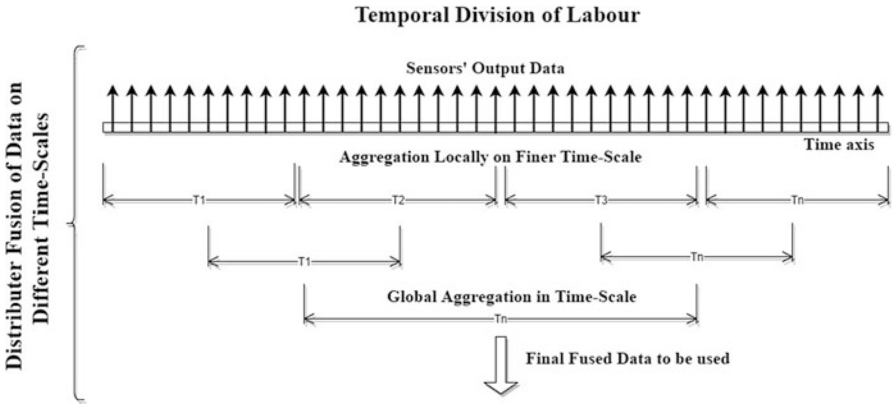

Temporal Division of Labour
Sensors' Output Data
Time axis
Aggregation Locally on Finer Time-Scale
T1
T2
T3
Tn
T1
Tn
Global Aggregation in Time-Scale
Tn
Final Fused Data to be used
Distributer Fusion of Data on Different Time-Scales

Fig. 5.8 Temporal division of labour

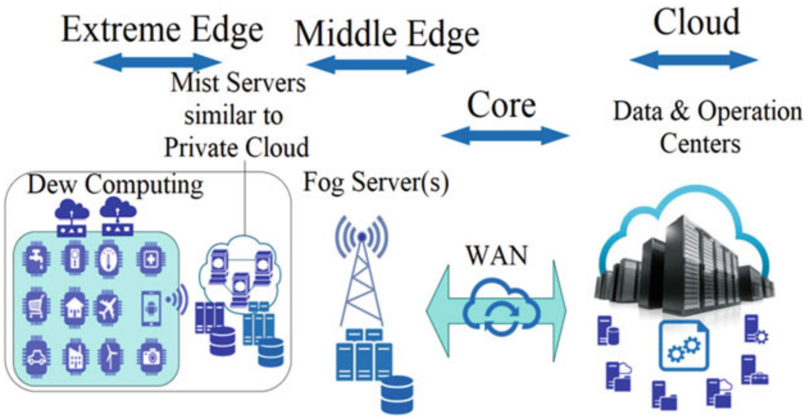


Fig. 5.9 CPSs with edge computing and cloud

Example: MEAN of N samples -> 'T_0' Time. Buffer samples on a finer time scale 'd_0' store the value, i.e. mean on a Finer time scale Nd_0 samples on a Coarser timescale takes more time.

5.4.2.2 Effective Idea: Edge Computing

If we require to manage real-time (with hard and soft deadlines) and non-real-time (or relaxed real-time computation) computation, the hierarchical architecture of cloud-fog-mist-dew computing can be utilized as shown in Fig. 5.9. This is also called Edge Computing architecture. Edge Computing has emerged as an advanced technology for future CPSs and IoTs. It seems to solve the issues of managing delays in hard real-time applications with ease. Along with this, it is going to resolve the issue of excessive bandwidth requirement in core Internet in coming years.

Edge Computing is the computing in Cloud Servers near to the physical sensing environment [65, 66]. Cloud Servers on Edge are also called Fog Servers or Cloudlets [67–69]. They can be just a few hops away from the sensors so decreases the delay dramatically for real-time applications. Even with the lower configuration than a Cloud Server, Fog Server [70, 71] gives better performance by removing the delays of data transmission in Core Internet [72]. This [73] Cloud Hierarchy can be extended further to the Extreme Edge, where computation is done in Mist Servers situated in the vicinity of the sensors just like a private Cloud. There can be many Mist servers available in the boundaries of an organization/institute. Mist servers further manage the delay-sensitive applications better, for example, managing industrial pipeline issues, fire alarms, etc. [74, 75] Computation in Extreme Edge can be done in Sensor Nodes as well [76]. Processors nowadays have the capability to manage some data locally and can do local computations. This is called Dew Computing. There can be local fusion at a sensor on a finer time scale followed by storage locally or fusion over a coarser time scale [77] on stored samples meeting the critical time deadline. Thus, we have a temporal hierarchical fusion of sensed data [78].

A similar idea is applied for opportunistic Communication [79] and control. For this, we can sense all the free channels at any point of time, snatch the band when available and locally transfer to the intermediate node and then to the destination node. This idea can be applied to Edge Computing as well to transmit data over to the Cloud from Fog Server.

5.4.3 Software System: Temporal Semantics

In the design and analysis of algorithms, various sorting algorithms are thoroughly understood/analysed for time complexity as well as memory complexity [80, 81]. Thus, any application (for instance, sorting marks of students) which uses an efficient sorting algorithm can be predicted for the amount of time taken for generating the output (for worst case input). This conclusion is true for various other algorithm based applications. But there are other software systems (e.g. software switches/routers) for which estimating the time required for getting desired output cannot be easily predicted. Formal computational models of special/general purpose software systems are very useful to determine the time to generate an output given worst case input [81–83]. Software testing approaches could also be adopted for estimating the time complexity.

Most software systems are designed to be modular involving "Functions/Subroutines" which can be tested for estimating the time complexity under worst case input. Most Cyber Physical Systems are based on software and/or hardware systems. Thus, temporal semantics must take into account the architecture involving the integration of software and hardware systems.

It should be kept in mind that extensive testing of distributed software/hardware systems may be prohibitive in terms of time required and may not be possible. Thus, new innovative solutions need to be discovered.

5.5 Reliability and Fault Tolerance: Concurrent Cyber Physical Systems

5.5.1 Fault Tolerance

The design of [84] Concurrent Cyber Physical Systems evolving on multiple time scales must be fault tolerant and failure tolerant [85]. Such a design is very critical since failures/faults in sub-systems could be very disastrous in various applications [86]. Fault/failure tolerant system design is investigated by various researchers. Such results provide useful insight in the case of Cyber Physical System design. But the highly interconnected network of Cyber Physical Systems presents various new challenges which are not addressed in the traditional fault-tolerant design.

5.5.2 Fault/Failure Localization:

From a coarse description viewpoint, a graph captures the topology of the interconnected network of Cyber Physical System that potentially spatially distributed and evolving in time on multiple (time) scales.

Diagnostic subsystem embedded into the CPS network must be designed such that fault/failure localization and repair can be done in real-time [87]. In other words, a network of CPSs should be designed for fault/failure tolerance. Well-known approaches such as Design For Testability (DFT), BUILT In Self Test (BIST) in embedded system design provide useful indications/hints.

From a control theoretic viewpoint [88], coupled CPSs involving many feedback loops must be analytically and/or simulation-wise be examined for "stability" issues. In fact, new technologies are required when some of the CPSs are nonlinear. Issues such as positive feedback, resonance must be tested so that small disturbance/noise are not amplified significantly leading to failure of subsystems. The design of the architecture of networked CPS will involve new problems, solutions from a graph/hypergraph theoretic viewpoint.

5.5.3 Architectural Considerations: Cyber Physical Systems

It is essential that the architecture of any CPS is able to localize fault (fault detection and localization) and be designed in such a way that the entire system is fault tolerant.

In the case of distributed, networked Cyber Physical System there are at least two categories with respect to the topology:

- Topology of networked CPS is under the control of user: Controlled Deployment
- Topology of networked CPS is not under the control of user: Uncontrolled Deployment

Example Consider a wireless sensor network deployed in a building to detect and communicate FIRE in the building. In this case, the user can control the topology/deployment of the wireless network.

Suppose the wireless sensor network is deployed in a forest (using say Helicopter) to monitor and communicate FIRE. The topology/deployment is not under the control of the user. We can also run into distributed networked CPS in which the topology of a portion of the system is under the control of the user and some portion is not under the control of the user. It should be noted that those CPS whose topology is under the control of user are easy to monitor/diagnose and operate.

We are thus naturally led to the design of distributed CPS that are easy to diagnose/monitor and maintain. As in digital telephone networks, the CPS is integrated with an Operations Support System (OSS) for diagnosing/ monitoring/maintaining various subsystems. For such purpose, we propose an overlay network. Such operations support system could be based on modeling abstractions of distributed CPS. The main challenge is that the dynamics of the physical/physiological system could only be partially understood.

In some cases, the dynamics are highly nonlinear and not well understood. One usually resorts to simulations to partially understand the dynamics of such systems. Let us consider the example of Wireless Body Area Network (WBAN). Even though diseases of various organs and their functioning/malfunctioning is understood to a limited extent, well-designed WBAN can monitor the health of various interacting organs in real-time and can provide good diagnostic support to the doctor. The equipment such as ECG machine, MRI scanner can be endowed with software to analyse the 1-D/2-D/3-D signal locally and provide diagnostic help. Also, the equipment could be locally networked to monitor the health of various interacting organs.

In reliability theory applied to fault-tolerant computing systems, various innovative ideas are proposed [89]. Some of those ideas are also applicable to Cyber Physical Systems [90]. Various types of redundancy (e.g. 1-in-N redundant systems) are incorporated to make the entire system sufficiently reliable. By means of test inputs fed to various CPS (Wired Communication or Wireless Communication), it could be possible to monitor the health of various subsystems in a large system. Thus Operations Support System can monitor the health of distributed CPS.

The architecture of wired/wireless sensors could be modified to be able to locally compute fusion functions such as Mean, Median, Mode, etc. on a finer timescale. For instance, underwater sensors can be designed to monitor the earthquake/tsunami. Also, existing sensor designs should be modified in such a way that the architecture is tailored to the MAC, routing protocols utilized to network the sensors.

5.5.4 Reliability and Fault Management Using Edge Servers

Internet of Things (IoTs) has emerged as a mechanism for connecting physical world and the cyber world via Internet [91]. So while talking about the reliability and fault tolerance in CPS, the whole mechanism can be represented as IoTs alternatively.

5.5.4.1 Reliability and Backup Policy

One of the important aspects in CPSs is reliability [92]. Reliability refers to the probability of a system working correctly, provided it was working correctly at time t equals zero. That means we are considering the probability over time that the system is working.

To construct a reliable CPS or IoT-Cloud infrastructure the replication of sensed data is very important [93]. The redundancy in data makes sure the feedback for the actuators based on the past sensed data [94, 95]. The backup system can take over the control in case of any failure. For Clouds, we use the concept of AZ (Availability Zone) for backup that can be used in case of any disaster or failure in the Cloud server [96].

Now while using the concept of Edge Computing, the focus is on running CPS Applications from the Edge of the network rather than from the cloud that is so far away from the end sensor devices. So for all the possibilities [97], Fog and Mist level servers should also have backup servers on their levels to run the applications from alternative servers in case of failure of any one of them.

The multi-layered architecture is reliable [98] in the sense that even if there are no alternative servers on the same layer having data replications, there is a guaranty of replication of sensed data on the higher and upper layer. The volume of data for the same sensor may vary on Dew, Mist, Fog and Cloud layers [99] but it would be there on all layers to take over the system in case of absence of any server on any layer.

5.5.4.2 Fault Tolerance and Agent

By definition, Mobile Agent is a special purpose software code that can transfer itself from one machine to another, but practically it is the same code running on all the machines and just data gets transmitted from one system to another. Mobile Agents are very helpful in managing distributed systems as there is no central system to manage them. In this proposed work, the faults [100, 101] in the whole hierarchy will be managed by Mobile Agents.

Here Agent will work as Resource and Network Monitoring Agent. It will share the neighbour information and link state information with other agents on alternative servers [101], if available. Except monitoring, they will also be responsible for assigning the priority to the CPS applications depending upon their delay-tolerance. This priority information will be used at the time of load distribution [102] of a particular server in case of impacted failure or impacted scheduled shutdown activity [103]. It will also help in new path discovery [104] after the load distribution in the case of faults. It will also keep a check on timing intervals of monitoring and backing up the data.

Figure 5.10 shows the recovery process in case of faults and scheduled shutdowns, reactively and proactively in respective case. It is depicting the complete fault tolerance cycle of the hierarchy between Dew, MIST, Fog and cloud. Here the fault tolerance has been achieved by exploiting the capabilities and benefits of an agent, which is basically a software program running on each server. Also at the time of any fault occurrence, the whole responsibility has been assumed/assigned for upper layer agent, not the affected layer's agent.

5.6 Agent Working in Different Conditions

1. As per assumption, the reason behind any fault/shutdown can be either any scheduled activity or any sudden activity.
2. Any sudden or shutdown activity comes with two possibilities that either it's going to affect respective server/device or no effect at all on server/device.
3. So as per represented in the graph if the ongoing activity is effectless then for scheduled event agent will move with proactive actions which are basically to send notifications to the all respective whereas for sudden activity agent will look for reactive action.
4. For any sudden fault, the agent will fetch priority index for all applications of the affected server and immediately it will check if any other server available on the same layer [104, 105] (either in same or different availability zone). After that it will do application migration [106] and connection redirection and then will do the load transfer activity as per the sequence given below:

Total load $= C$
Server S can handle (load) $= T$

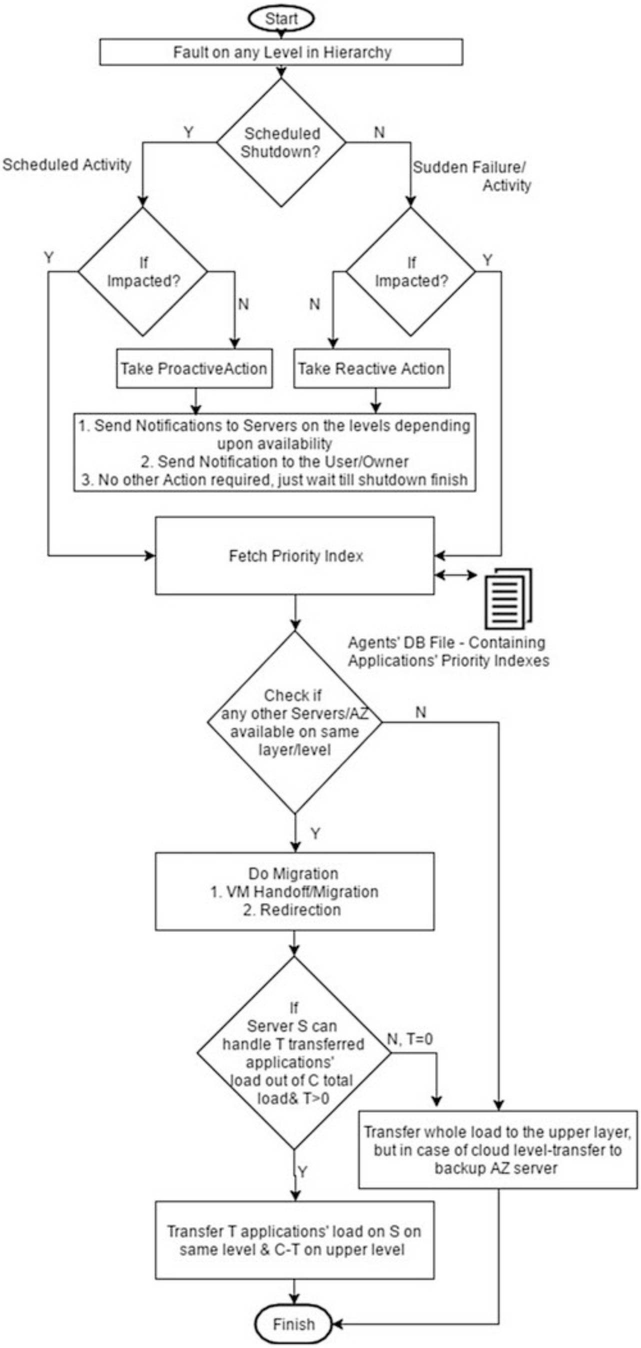

Start
Fault on any Level in Hierarchy
Scheduled Shutdown?
Y
N
Scheduled Activity
Sudden Failure/ Activity
If Impacted?
Y
N
If Impacted?
N
Y
Take ProactiveAction
Take Reactive Action
1. Send Notifications to Servers on the levels depending upon availability
2. Send Notification to the User/Owner
3. No other Action required, just wait till shutdown finish
Fetch Priority Index
Agents' DB File - Containing Applications' Priority Indexes
Check if any other Servers/AZ available on same layer/level
N
Y
Do Migration
1. VM Handoff/Migration
2. Redirection
If Server S can handle T transferred applications' load out of C total load& T>0
N, T=0
Y
Transfer whole load to the upper layer, but in case of cloud level-transfer to backup AZ server
Transfer T applications' load on S on same level & C-T on upper level
Finish

Fig. 5.10 Agents' working in faults

```
if $T > 0$,
then
    Transfer $T$ (load) → $S$
    Transfer ($C$–$T$) load → upper layer
else if $T = 0$
    Transfer whole load → upper layer
```

5. But unfortunately, if no other same layer server is available, then it will point for the upper layer to transfer the whole load.

5.7 Conclusion

This book chapter is all about providing appropriate models and solutions for Concurrent Cyber Physical Systems. The chapter takes into consideration the concurrency and the synchronization requirement of the systems. Tensor State Space Representation (TSSR) is used to model linear concurrent Cyber Physical System. The model works for both homogeneous and heterogeneous type of systems. The chapter also defined the idea of delay buffers for synchronizing multiple subsystems and model for clock synchronization of multidimensional Cyber Physical Systems. Importance of Time or Temporal semantics can't be denied in concurrent systems. Importance of time in real-time Cyber Physical Systems is discussed and a time-oriented solution "Temporal Division of Labour" for data fusion on multiple levels of systems is represented. At last, one advanced technology Edge Computing was discussed as a solution to multiple issues. It has been shown that how the hierarchical Cloud can manage temporal data and how it can also deal with reliability, fault tolerance and fault localization issues. All the formal discussions regarding CPSs are very important to have a theoretical understanding of issues and their solutions.

Acknowledgments The authors gratefully acknowledge the helpful suggestions from Dr. Shivangi Katiyar, Assistant Professor, CGC Landran. This book chapter is entirely written in MS Word and all the figures are created on website https://www.draw.io/.

References

1. Lee EA, Seshia SA (2016) Introduction to embedded systems: a cyber-physical systems approach. MIT Press, Cambridge
2. Kopetz H (2011) Real-time systems: design principles for distributed embedded applications. Springer, Berlin
3. Givargis T, Russo S (eds) (2008) Software technologies for embedded and ubiquitous systems: 6th IFIP WG 10.2 international workshop, SEUS 2008, Anacarpi, Capri Island, Italy, Oct 1–3, 2008, Revised Papers (Vol. 5287). Springer

4. Lee J, Bagheri B, Kao HA (2015) A cyber-physical systems architecture for industry 4.0-based manufacturing systems. Manuf Lett 3:18–23
5. Jensen JC, Chang DH, Lee EA (2011) A model-based design methodology for cyber-physical systems. In: Wireless communications and mobile computing conference (IWCMC), 2011 7th international, IEEE, pp 1666–1671
6. Marwedel P (2010) Embedded system design: embedded systems foundations of cyber-physical systems. Springer, Berlin
7. Lee EA (2006) Cyber-physical systems-are computing foundations adequate. In: Position paper for NSF workshop on cyber-physical systems: research motivation, techniques and roadmap (Vol. 2)
8. Baheti R, Gill H (2011) Cyber-physical systems. Impact Control Technol 12:161–166
9. Xiong G, Zhu F, Liu X, Dong X, Huang W, Chen S, Zhao K (2015) Cyber-physical-social system 5n intelligent transportation. IEEE/CAA J Automatica Sinica 2(3):320–333
10. Camarinha-Matos LM, Falcão AJ, Vafaei N, Najdi S (eds) (2016) Technological innovation for cyber-physical systems: 7th IFIP WG 5.5/SOCOLNET advanced doctoral conference on computing, electrical and industrial systems, DoCEIS 2016, Costa de Caparica, Portugal, April 11–13, 2016, Proceedings (Vol. 470). Springer
11. Khaitan SK, McCalley JD (2015) Design techniques and applications of cyberphysical systems: a survey. IEEE Syst J 9(2):350–365
12. Wirsing M, Banâtre JP, Hölzl M, Rauschmayer A (eds) (2008) Software-intensive systems and new computing paradigms: challenges and visions (Vol. 5380). Springer
13. He J, Chen J, Cheng P, Cao X (2014) Secure time synchronization in wireless sensor networks: a maximum consensus-based approach. IEEE Transac Parallel Distrib Syst 25(4):1055–1065
14. Freris NM (2017) A software defined architecture for cyberphysical systems. In: Software defined systems (SDS), 2017 fourth international conference on, IEEE, pp 54–60
15. Lee EA (2007) Computing foundations and practice for cyber-physical systems: a preliminary report. University of California, Berkeley Technical Report UCB/EECS-2007-72
16. Lee EA (2015) The past, present and future of cyber-physical systems: a focus on models. Sensors 15(3):4837–4869
17. Liu Y, Peng Y, Wang B, Yao S, Liu Z (2017) Review on cyber-physical systems. IEEE/CAA J Automatica Sinica 4(1):27–40
18. Siaterlis C, Genge B, Hohenadel M (2013) EPIC: a test bed for scientifically rigorous cyber-physical security experimentation. IEEE Transac Emerg Topics Comput 1(2):319–330
19. Agha G, Danvy O, Meseguer J (eds) (2011). Formal modeling: actors; open systems, biological systems: essays dedicated to carolyn talcott on the occasion of her 70th birthday (Vol. 7000). Springer
20. Ungureanu G, Sander I (2017) A layered formal framework for modeling of cyber-physical systems. In: 2017 design, automation & test in Europe conference & exhibition (DATE), IEEE, pp 1715–1720
21. Andrade HA, Derler P, Eidson JC, Li-Baboud YS, Shrivastava A, Stanton K, Weiss M (2015) Towards a reconfigurable distributed testbed to enable advanced research and development of timing and synchronization in cyber-physical systems. In: ReConFigurable computing and FPGAs (ReConFig), 2015 international conference on, IEEE, pp 1–6
22. Rajkumar RR, Lee I, Sha L, Stankovic J (2010) Cyber-physical systems: the next computing revolution. In: Proceedings of the 47th design automation conference. ACM, New York, pp 731–736
23. Kim KD, Kumar PR (2012) Cyber–physical systems: a perspective at the centennial. Proc IEEE 100(Special Centennial Issue):1287–1308
24. Shi J, Wan J, Yan H, Suo H (2011) A survey of cyber-physical systems. In: Wireless communications and signal processing (WCSP), 2011 international conference on, IEEE, pp 1–6
25. Wu FJ, Kao YF, Tseng YC (2011) From wireless sensor networks towards cyber physical systems. Pervasive Mobile Comput 7(4):397–413

26. Hu F, Lu Y, Vasilakos AV, Hao Q, Ma R, Patil Y, Xiong NN (2016) Robust cyber–physical systems: concept, models, and implementation. Futur Gener Comput Syst 56:449–475
27. Derler P, Lee EA, Sangiovanni-Vincentelli AL (2011) Addressing modeling challenges in cyber-physical systems (No. UCB/EECS-2011-17)
28. Lee EA (2008) Cyber physical systems: design challenges. In: Object oriented real-time distributed computing (isorc), 2008 11th IEEE international symposium on, IEEE, pp 363–369
29. Chandhoke S, Hayles T, Kodosky J, Wang G (2011) A model-based methodology of programming cyber-physical systems. In: Wireless communications and mobile computing conference (IWCMC), 2011 7th international, IEEE, pp 1654–1659
30. Tan Y, Goddard S, Perez LC (2008) A prototype architecture for cyber-physical systems. ACM Sigbed Rev 5(1):26
31. Artho C, Legay A, Peled D (2016) Automated technology for verification and analysis. Springer, Berlin
32. Lee EA (2010) CPS foundations. In: Design automation conference (DAC), 2010 47th ACM/IEEE, IEEE, pp 737–742
33. Liu Z, Yang DS, Wen D, Zhang WM, Mao W (2011) Cyber-physical-social systems for command and control. IEEE Intell Syst 26(4):92–96
34. Al-Nayeem A, Sha L, Cofer DD, Miller SM (2012) Pattern-based composition and analysis of virtually synchronized real-time distributed systems. In: Cyber-physical systems (ICCPS), 2012 IEEE/ACM third international conference on, IEEE, pp 65–74
35. Kim M, Stehr MO, Kim J, Ha S (2010) An application framework for loosely coupled networked cyber-physical systems. In: Embedded and ubiquitous computing (EUC), 2010 IEEE/IFIP 8th international conference on, IEEE, pp 144–153
36. Gelenbe E, Wu FJ (2013) Future research on cyber-physical emergency management systems. Future Internet 5(3):336–354
37. Sztipanovits J, Koutsoukos X, Karsai G, Kottenstette N, Antsaklis P, Gupta V, Wang S (2012) Toward a science of cyber–physical system integration. Proc IEEE 100(1):29–44
38. Conti M, Das SK, Bisdikian C, Kumar M, Ni LM, Passarella A, Zambonelli F (2012) Looking ahead in pervasive computing: challenges and opportunities in the era of cyber–physical convergence. Pervasive Mobile Comput 8(1):2–21
39. Mosterman PJ, Zander J (2016) Cyber-physical systems challenges: a needs analysis for collaborating embedded software systems. Softw Syst Modeling 15(1):5–16
40. Garimella RM (2014) Concurrent cyber physical systems: tensor state space representation. In: Control & Automation (ICCA), 11th IEEE international conference on, IEEE, pp 1232–1237
41. Xiao K, Ren S, Kwiat K (2008) Retrofitting cyber physical systems for survivability through external coordination. In: Hawaii international conference on system sciences, proceedings of the 41st annual, IEEE, pp 465–465
42. Balani R, Wanner LF, Friedman J, Srivastava MB. Lin K, Gupta RK (2011) Programming support for distributed optimization and control in cyber-physical systems. In: Proceedings of the 2011 IEEE/ACM second international conference on cyber-physical systems, IEEE computer society, pp 109–118
43. Eidson JC, Lee EA, Matic S, Seshia SA, Zou J (2012) Distributed real-time software for cyber–physical systems. Proc IEEE 100(1):45–59
44. Fink J, Ribeiro A, Kumar V (2012) Robust control for mobility and wireless communication in cyber–physical systems with application to robot teams. Proc IEEE 100(1):164–178
45. Deng X, Yang Y (2013) Communication synchronization in cluster-based sensor networks for cyber-physical systems. IEEE Transac Emerg Topics Comput 1(1):98–110
46. Yadav A, Waghmare A, Sairam AS (2014) Exploiting node heterogeneity for time synchronization in low power sensor networks. In: Contemporary computing and informatics (IC3I), 2014 International conference on, IEEE, pp 828–832
47. Yokoyama T, Matsubara A, Yoo M (2015) A real-time operating system with gnss-based tick synchronization. In: Cyber-physical systems, networks, and applications (CPSNA), 2015 IEEE 3rd international conference on, IEEE, pp 19–24

48. Shih CS, Yang CM, Cheng YC (2015) Data alignment for multiple temporal data streams without synchronized clocks on iot fusion gateway. In: Data science and data intensive systems (DSDIS), 2015 IEEE international conference on, IEEE, pp 667–674
49. Lee EA, Matic S, Seshia SA, Zou J (2009) The case for timing-centric distributed software invited paper. In: Distributed computing systems workshops, 2009. ICDCS Workshops' 09. 29th IEEE international conference on, IEEE, pp 57–64
50. Subramanian V, Umbarkar A, Doboli A (2012) Decentralized event detection using distributed interrupts in Cyber Physical Systems. In: Systems conference (SysCon), 2012 IEEE international, IEEE, pp 1–6
51. Lee EA (2009) Computing needs time. Commun ACM 52(5):70–79
52. Tan Y, Vuran MC, Goddard S (2009) Spatio-temporal event model for cyber-physical systems. In: Distributed computing systems workshops, 2009. ICDCS Workshops' 09. 29th IEEE international conference on, IEEE, pp 44–50
53. Broman D, Derler P, Eidson J (2013) Temporal issues in cyber-physical systems. J Indian Inst Sci 93(3):389–402
54. Eidson J, Lee EA, Matic S, Seshia SA, Zou J (2010) A time-centric model for cyber-physical applications. In: Workshop on model based architecting and construction of embedded systems (ACES-MB), pp 21–35
55. Georg H, Muller SC, Dorsch N, Rehtanz C, Wietfeld C (2013) INSPIRE: integrated co-simulation of power and ICT systems for real-time evaluation. In: Smart grid communications (SmartGridComm), 2013 IEEE international conference on, IEEE, pp 576–581
56. Anwar F, D'souza S, Symington A, Dongare A, Rajkumar R, Rowe A, Srivastava M (2016) Timeline: an operating system abstraction for time-aware applications. In: Real-time systems symposium (RTSS), 2016 IEEE, pp 191–202 IEEE
57. Bui D, Lee E, Liu I, Patel H, Reineke J (2011) Temporal isolation on multiprocessing architectures. In: Proceedings of the 48th design automation conference. ACM, New York, pp 274–279
58. Liu K, Lee VCS, Ng JKY, Chen J, Son SH (2014) Temporal data dissemination in vehicular cyber–physical systems. IEEE Trans Intell Transp Syst 15(6):2419–2431
59. Sun H, Liu J, Chen X, Du D (2015) Specifying cyber physical system safety properties with metric temporal spatial logic. In: Software engineering conference (APSEC), 2015 Asia-Pacific, IEEE, pp 254–260
60. Weiss M, Chandhoke S, Melvin H (2015) Time signals converging within cyber-physical systems. In: Frequency control symposium & the European frequency and time forum (FCS), 2015 joint conference of the IEEE international, IEEE, pp 684–689
61. Dai W, Vyatkin V, Pang C, Christensen JH (2015) Time-stamped event based execution semantics for industrial cyber-physical systems. In: Industrial informatics (INDIN), 2015 IEEE 13th international conference on, IEEE, pp 1263–1268
62. Eidson JC, Stanton KB (2015) Timing in cyber-physical systems: the last inch problem. In: Precision clock synchronization for measurement, control, and communication (ISPCS), 2015 IEEE international symposium on, IEEE, pp 19–24
63. Petnga L, Austin M (2013) Ontologies of time and time-based reasoning for MBSE of cyber-physical systems. Procedia Comput Sci 16:403–412
64. Broman D, Zimmer M, Kim Y, Kim H, Cai J, Shrivastava A, Lee EA (2013) Precision timed infrastructure: design challenges. In: Electronic system level synthesis conference (ESLsyn), 2013, IEEE, pp 1–6
65. Patel M, et al (2014) Mobile-edge computing introductory technical white paper. White paper, Mobile-edge Computing (MEC) industry initiative
66. Glikson A, Nastic S, Dustdar S (2017) Deviceless edge computing: extending serverless computing to the edge of the network. In: Proceedings of the 10th ACM international systems and storage conference. ACM, New York, p 28
67. Gao Y, Hu W, Ha K, Amos B, Pillai P, Satyanarayanan M (2015) Are cloudlets necessary? School of Computer Science, Carnegie Mellon University, Pittsburgh Technical Report CMU-CS-15-139

68. Datta SK, Bonnet C, Haerri J (2015) Fog computing architecture to enable consumer centric internet of things services. In: Consumer electronics (ISCE), 2015 IEEE international symposium on, IEEE, pp 1–2
69. Chen S, Zhang T, Shi W (2017) Fog computing. IEEE Internet Comput 21(2):4–6
70. Bonomi F, Milito R, Natarajan P, Zhu J (2014) Fog computing: a platform for internet of things and analytics. In: Big data and internet of things: a roadmap for smart environments. Springer, Berlin, pp 169–186
71. Dastjerdi AV, Buyya R (2016) Fog computing: helping the internet of things realize its potential. Computer 49(8):112–116
72. Satyanarayanan M (2017) The emergence of edge computing. Computer 50(1):30–39
73. Wan J, Xia M (2017) Cloud-assisted cyber-physical systems for the implementation of industry 4.0. Mobile Netw Appl 22:1–2
74. Bonomi F, Milito R, Zhu J, Addepalli S (2012) Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on mobile cloud computing. ACM, New York, pp 13–16
75. Wen Z, Yang R, Garraghan P, Lin T, Xu J, Rovatsos M (2017) Fog orchestration for internet of things services. IEEE Internet Comput 21(2):16–24
76. Yannuzzi M, van Lingen F, Jain A, Parellada OL, Flores MM, Carrera D, Olive A (2017) A new era for cities with fog computing. IEEE Internet Comput 21(2):54–67
77. Satyanarayanan M, Simoens P, Xiao Y, Pillai P, Chen Z, Ha K, Hu W, Amos B (2015) Edge analytics in the internet of things. IEEE Pervasive Comput 14(2):24–31
78. Mahmud R, Buyya R (2016) Fog computing: a taxonomy, survey and future directions. arXiv preprint arXiv:1611.05539
79. Mach P, Becvar Z (2017) Mobile edge computing: a survey on architecture and computation offloading. IEEE communications surveys & tutorials
80. Sorensen A, Gardner H (2010) Programming with time: cyber-physical programming with impromptu. In: ACM Sigplan notices (Vol. 45, No. 10, pp 822–834). ACM
81. Raman V, Donzé A, Maasoumy M, Murray RM, Sangiovanni-Vincentelli A, Seshia SA (2014) Model predictive control with signal temporal logic specifications. In: Decision and control (CDC), 2014 IEEE 53rd annual conference on, IEEE, pp 81–87
82. Wang H, Zhou X, Dong Y, Tang L (2009) Modeling timing behavior for cyber-physical systems. In: Computational intelligence and software engineering, 2009. CiSE 2009. International conference on, IEEE, pp 1–4
83. Tan F, Wang Y, Wang Q, Bu L, Zheng R, Suri N (2013). Guaranteeing proper-temporal-embedding safety rules in wireless cps: a hybrid formal modeling approach. In: Dependable systems and networks (DSN), 2013 43rd Annual IEEE/IFIP International conference on, IEEE, pp 1–12
84. Wan J, Yan H, Suo H, Li F (2011) Advances in cyber-physical systems research. KSII Transac Internet Info Syst (TIIS) 5(11):1891–1908
85. Saha I, Roy S, Ramesh S (2016) Formal verification of fault-tolerant startup algorithms for time-triggered architectures: a survey. Proc IEEE 104(5):904–922
86. Crenshaw TL, Gunter E, Robinson CL, Sha L, Kumar PR (2007) The simplex reference model: limiting fault-propagation due to unreliable components in cyber-physical system architectures. In: Real-time systems symposium, 2007. RTSS 2007. 28th IEEE international, IEEE, pp 400–412
87. Abdelwahed S, Kandasamy N, Gokhale A (2007) High confidence software for cyber-physical systems. In: Proceedings of the 2007 workshop on automating service quality: held at the international conference on automated software engineering (ASE). ACM, New York, pp 1–3
88. Wang X, Hovakimyan N, Sha L (2013) L1Simplex: fault-tolerant control of cyber-physical systems. In: Proceedings of the ACM/IEEE 4th international conference on cyber-physical systems. ACM, New York, pp 41–50
89. Zhang Y, Xie F, Dong Y, Zhou X, Ma C (2013) Cyber/physical co-verification for developing reliable cyber-physical systems. In: Computer software and applications conference (COMPSAC), 2013 IEEE 37th Annual, IEEE, pp 539–548

90. Zhang S, Vittal V (2014) Wide-area control resiliency using redundant communication paths. IEEE Trans Power Syst 29(5):2189–2199
91. Satyanarayanan M, Schuster R, Ebling M, Fettweis G, Flinck H, Joshi K, Sabnani K (2015) An open ecosystem for mobile-cloud convergence. IEEE Commun Mag 53(3):63–70
92. Schoitsch E (ed) (2010) Computer safety, reliability, and security. In: 29th international conference, SAFECOMP 2010, Vienna, Austria, Sept 14–17 2010, Proceedings (Vol. 6351). Springer
93. Kang W, Kapitanova K, Son SH (2012) RDDS: a real-time data distribution service for cyber-physical systems. IEEE Transac Ind Info 8(2):393–405
94. Martin A, Fetzer C, Brito A (2011) Active replication at (almost) no cost. In: Reliable distributed systems (SRDS), 2011 30th IEEE symposium on, IEEE, pp 21–30
95. Shen M, Kshemkalyani AD, Hsu TY (2015). Causal consistency for geo-replicated cloud storage under partial replication. In: Parallel and distributed processing symposium workshop (IPDPSW), 2015 IEEE international, IEEE, pp 509–518
96. Mittal D, Agarwal N (2015) A review paper on fault tolerance in cloud computing. In: Computing for sustainable global development (INDIACom), 2015 2nd International conference on, IEEE, pp 31–34
97. Yusuf II, Thomas IE, Spichkova M, Androulakis S, Meyer GR, Drumm DW, Schmidt HW (2015) Chiminey: reliable computing and data management platform in the cloud. In: Proceedings of the 37th international conference on software engineering-Vol. 2, IEEE Press, pp 677–680
98. Madsen H, Burtschy B, Albeanu G, Popentiu-Vladicescu FL (2013) Reliability in the utility computing era: towards reliable fog computing. In: Systems, signals and image processing (IWSSIP), 2013 20th international conference on, IEEE, pp 43–46
99. Elmroth E, Leitner P, Schulte S, Venugopal S (2017) Connecting fog and cloud computing. IEEE Cloud Comput 4(2):22–25
100. Wei Y (2014) Auto-configurable, reliable, and fault-tolerant cloud storage with dynamic parameterization. In: Services (SERVICES), 2014 IEEE world congress on, IEEE, pp 295–300
101. Gan H, Chen L (2014) An efficient data integrity verification and fault-tolerant scheme. In: Communication systems and network technologies (CSNT), 2014 fourth international conference on, IEEE, pp 1157–1160
102. Skobelev P, Trentesaux D (2017) Disruptions are the norm: cyber-physical multi-agent systems for autonomous real-time resource management. In: Service orientation in holonic and multi-agent manufacturing. Springer, Cham, pp 287–294
103. Jennings B, Stadler R (2015) Resource management in clouds: survey and research challenges. J Netw Syst Manag 23(3):567–619
104. Taylor B, Abe Y, Dey A, Satyanarayanan M, Siewiorek D, Smailagic A (2015) Virtual machines for remote computing: measuring the user experience
105. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for vm-based cloudlets in mobile computing. IEEE Pervasive Comput 8(4)
106. Ha K, Abe Y, Chen Z, Hu W, Amos B, Pillai P, Satyanarayanan M (2015) Adaptive vm handoff across cloudlets. Technical report, Technical Report CMU-C S-15–113, CMU School of Computer Science