Research Directions in Energy-Sustainable Cyber-Physical Systems [1]

Sandeep K. S. Gupta *, Tridib Mukherjee, Georgios Varsamopoulos, Ayan Banerjee

Impact Lab
School of Computing, Informatics, and Decision Systems Engineering
Arizona State University
Tempe, Arizona, USA

Abstract

An overview of sustainable computing is provided and different approaches towards design and verification of energy-sustainable computing (i.e. sustainable computing from energy consumption perspective) are discussed for Cyber-Physical Systems (CPSs), i.e. systems with strong coupling between computing components and non-computing processes in physical environment. A major issue in this regard is the inter-dependencies of the non-computing processes on the computing components and vice versa, and the verification of the CPSs' sustainability without real deployment. The trends and dependencies of energy consumption for both computing and non-computing components are conceptualized. Based on this conceptualization, CPS resource management algorithms are categorized according to: (i) computing workload execution and arrival profiles supported, (ii) knowledge of workload profiles during management decision making, (iii) support of power management in the computing components, and (iv) assumptions on non-computing process behavior. These categories are then discussed along with their pros and cons for two representative CPSs: data centers and Body Sensor Networks (BSNs). Model based engineering is used to verify CPS sustainability before real deployment. Several research directions and open problems are further discussed for design and verification of sustainable CPSs.

Key words: cyber-physical systems, sustainability, model-based engineering

1. Introduction

With the ongoing focus on environmental sustainability proliferating in different domains, sustainable computing, a.k.a. green computing, has been getting increased attention in recent years. There are three principal aspects of sustainable computing: (i) reduction of the energy required for running any computing infrastructure [1], e.g., energy-efficient management of data centers; (ii) ensuring longevity of computing equipment to reduce need for their replacement [1], e.g., avoid server breakdown in data centers by maintaining safe operating temperature; and (iii) ensuring energy consumption within the energy available from the renewable energy sources in the environment [2], e.g.,

* Corresponding author.
Email address: sandeep.gupta@asu.edu (Sandeep K. S. Gupta).
URL: http://impact.asu.edu/ (Sandeep K. S. Gupta).
[1] This work was funded in parts by NSF (CNS#0855277, CSR#0834797, CNS#0831544).

sensors on human body being powered by the energy generated from respiration, ambulation, and sunlight. This paper gives an overview on sustainable computing in general and focuses specifically on the *energy-sustainable* computing, i.e. sustainable computing from energy perspective. A major issue in addressing the different aspects of sustainable computing is the need for awareness of the non-computing processes in the physical environment, e.g., the dependency of the equipment longevity on environmental factors and the availability of energy from the environment.

Computing systems having strong coupling with the physical environment are referred as the Cyber-Physical Systems (CPSs). These systems usually monitor, coordinate, and control non-computing processes. Recent advances in the sensor technologies and embedded computing systems have seen a surge of CPSs being investigated [3–5]. ; Examples of CPSs include: *Body Sensor Networks (BSNs)* (i.e. network of medical sensors worn on implanted in human body) that interacts with the human physiology (i.e. a non-computing process) to monitor physiological conditions (e.g., heart rate, pulse rate, blood glucose level), *medical devices* that interacts with the human physiology to control physiological conditions (e.g., maintaining certain level of drug concentration using infusion pumps [5]), *autonomous vehicles* that interact with the vehicle mechanics (i.e. a non-computing process) to monitor and control vehicle's trajectory and dynamics, *disaster response systems* that interacts with various non-computing processes (e.g., human behavior, environment) to monitor critical events and coordinate proper response actions, etc.

In addition to the functional interactions with the non-computing processes as demonstrated by the previous examples, interactions with the non-computing processes can aid in the sustainability from an energy perspective. In this regard, energy scavenging (i.e. a type of interaction) can be performed from various sources in the physical environment such as body heat, sunlight, ambulation, vibration, respiration, and so on [2, 6]. Powering the computing components from these sources reduces the demand for grid power or battery power; thus reducing the carbon emissions and improving the environmental sustainability in general.

Ideally, the CPS operations should be designed such that the required power can be always supplied from the scavenging sources. Towards this objective, one option is to reduce the power demand of the computing operations so that it is always within the available power (or reduces the grid power demand as far as possible). In this regard, strategies for sustainable computing have focused on processor level power management schemes such as frequency control, voltage control, or sleep state scheduling [7], Medium Access Control (MAC) sleep scheduling of the wireless radio [8, 9] communication among the sensors, and amortization of the wireless communication energy in sensor networks with less expensive computation [10]. However, design of sustainable CPSs requires a holistic approach which is aware of the limited available energy from the scavenging sources.

The holistic cyber-physical perspective further helps in a proper awareness of the non-computing processes (other than the ones from which energy can be scavenged), which indirectly affects the sustainability of any computing infrastructure. For example, the cooling energy required to maintain safe operating temperatures in data centers can

be reduced if the impact of the computation on the cooling need is properly understood [11]. Indeed, the Total Cost of Ownership (TCO) of the data centers can be enormous because of a large amount of recurring energy cost, about half of which can be attributed to cooling [12]. As such, it is imperative to transcend the current sustainable computing practices of power management and server provisioning in data centers to a more holistic approach coordinating with the management of the cooling equipment.

Another major challenge in designing sustainable CPSs is the potential infeasibility of real life experimental evaluation. For example, building a data center to verify the holistic management strategies can be cumbersome in terms of both time and resources. Further, many CPSs, e.g., BSNs and autonomous vehicles can be hazardous to test in real situations due to risks associated with their malfunction. Therefore, automated sustainability verification need to be facilitated for CPSs. A well established methodology for such verification is the Model Based Engineering (MBE) [13]. Application of MBE in CPS however will require modeling of the both computing and non-computing processes along with their inter-dependencies. Any computing strategy would have to be analyzed for their effects on the sustainability, e.g., analysis of energy needs of both computing and non-computing processes for verification of energy-sustainability. Such modeling and analysis have to capture possible *spatio-temporal* dynamics of the inter-dependencies, e.g., variation of the available energy from scavenging sources (such as respiration) with respect to time (since respiration rate may depend on physical activities [14]) and space (since different portions of the body may extract different amount of energy [2]).

This paper intends to:

(i) conceptualize the trends and inter-dependencies of the power consumption in both computing and non-computing processes in a CPS;

(ii) categorize holistic resource management in CPSs that considers computing workload management, power management, and non-computing process management; and

(iii) identify research directions and open problems to design holistic resource management and facilitate model-based analysis of CPSs for sustainability verification.

Section 2 gives a brief overview on sustainable computing by discussing different perspectives towards sustainable computing and surveying different research directions taken in regards of these perspectives. Design and verification of energy-sustainable CPS is then discussed in Section 3. The power characteristics of CPSs is theoretically conceptualized based on the dependencies of the power consumption among the computing and non-computing components. To achieve any level of energy-sustainability, resource management algorithms in CPSs have to be aware of the behavior of non-computing processes and the impact of computing components on these processes. Such resource management algorithms are classified based on the workload arrival and execution profiles supported, knowledge of the workload during management decision making, support of power management in the computing components, and

assumptions on the behavior of the non-computing processes. The theoretical conceptualization of the power characteristics and pros and cons of different resource management classes are discussed for two representative CPSs: data centers (in Section 4) and BSNs (in Section 5). Section 6 discusses how model-based analysis can be performed for energy-sustainability verification of CPSs followed by various open problems and research directions in designing and verification of sustainable CPSs (in Section 7). Finally, Section 8 concludes the paper.

2. Sustainable computing

Sustainable computing can be defined from two different perspectives: (i) energy perspective; and (ii) equipment recycling perspective. These two perspectives are described below.

2.1. *Energy perspective*

From the energy perspective, sustainable computing, a.k.a. energy-sustainable computing, can be described as the balance between the power required for computation and the power available from renewable or green sources (i.e. sources in the environment such as solar power). For example, as shown in Figure 1, if the power available from the energy sources is higher than the required power, then the computation can be performed without any power from the grid (or battery). However, both available and required power may vary over time (e.g., solar power is not available during night and the power requirement depends on the time-varying computing operations performed). Computing operations become unsustainable if the required power is higher than the power available from the green sources (as indicated by the shaded regions in Figure 1), in which case the extra (or remaining) power needs to be extracted from the grid or battery. Sustainability of computing operations from the energy perspective (i.e. energy-sustainability) can be defined as follows:

Energy-sustainability *is the average percentage of energy used from the green sources to power all computing units.*

In other words, energy-sustainable computing needs to ensure minimum energy requirement from the grid or battery (i.e. minimizing the areas of the shaded regions in Figure 1). There can be different manifestation of the definition depending on the availability of energy from green sources. For example, if there is no energy available from the green sources (i.e. all the computing operation has to be run from the grid or battery), then energy-sustainability boils down to reducing the average energy required for the computing operations. In all other cases, the best energy-sustainable computing solution would be to ensure that all the computing operations can be powered by the energy generated from the green sources at all time. In such a case, *energy-sustainability can also be measured as the number of computing units which can be completely powered by the energy from green sources*. There are different research directions in achieving energy-sustainable computing such that the grid and battery power need is minimized:

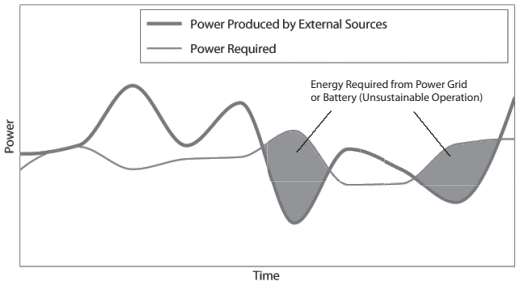


Fig. 1. Profile of power required and power available from external (green) sources. Unsustainable operation can be caused by imbalance of available and required power, in which case energy needs to be supplied from the power grid or battery.

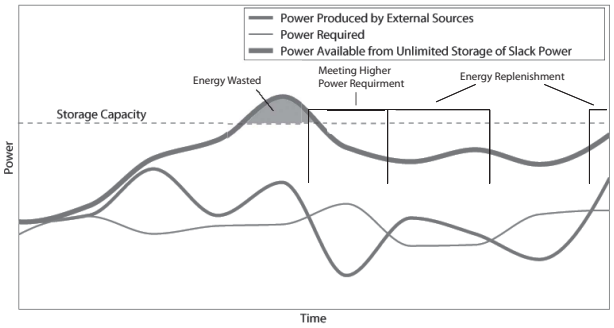


Fig. 2. Power imbalance in Fig. 1 can be addressed by through storage of energy. Stored energy accumulates from the slack between available and required power over time. Energy from the external sources can be wasted because of the limit on the storage capacity.

(i) **Energy storage:** Energy storage devices can store (and replenish) energy whenever available from the green sources. There has been several energy storage techniques such as ultra-capacitors, compressed air storage, batteries, fuel cells, and flywheels [15–17]. Figure 2 shows the variation of stored power for the available and required power profiles in Figure 1. The energy in storage device is accumulated over time depending on the slack between the available and required power. The stored energy can be used when power requirement is higher than the power available from the green sources. During this period, the stored energy reduces. This energy can get replenished when the slack between available and required power increases. Storage and replenishment of energy are constrained by the energy capacity limit of the storage device. This limit may incur wastage of energy generated by the green sources (as shown by the shaded region in Figure 2). The wastage can be significant if there are no replenishment in the later stages. Such a situation can lead to unsustainable operation if the power requirement is higher than the power available for long periods.

(ii) **Reducing energy requirement:** Another major research direction is to reduce the energy requirement. Reducing the energy requirement can either avoid unsustainable operation (when the energy requirement becomes always less the energy available) or reduce the energy need from the power grid or battery (reducing the shaded areas in Figure 1). Following are the different research requirements in this regard:

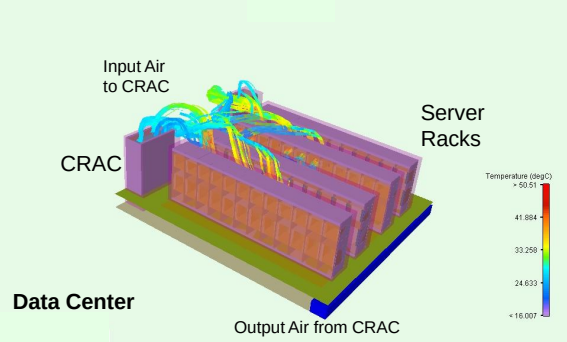
– *Spatio-temporal distribution of operation:* One way to achieve the reduction is through distribution of the computing operation in a spatio-temporal manner. Spatial distribution ensures that the computing operation are distributed in multiple computing units such that not a single unit gets overloaded. Spatial distribution is necessary to avoid high energy requirement in the bottleneck units (i.e. the ones being overloaded). Temporal distribution of computing operation is geared towards delaying the operations until the available power increases. However, such delaying of operations can undesirably affect the performance of the computing operations and are therefore constrained by the performance requirements (e.g., Service Level Agreements (SLAs)

in data centers). Spatio-temporal distribution of operation is widely known as workload (or job) management, i.e. the decision making to determine when (job scheduling) and where (job assignment or dispatching) to execute the computing workloads. Job scheduling and assignment problems are in general NP-hard [18]. In the case of online job scheduling, fast heuristic algorithms and policies are extensively used, such as *first-come first-serve* (FCFS) augmented with *back-filling* [19]. With respect to energy-sustainability, previous research has focused on: (i) including economical models for job schedules [20]; (ii) avoiding or even preventing excessive heat conditions in data centers through job assignment algorithms [3,21,22] thus improving the sustainability (through reduction in cooling power requirement); and (iii) performing spatio-temporal job scheduling (i.e. integrated job scheduling and assignment decision making) in data centers [11, 23, 24].

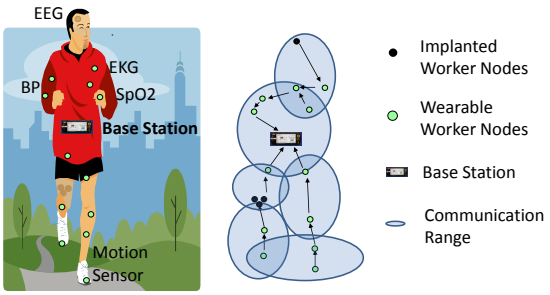
– *Computing power management:* A widely used method to reduce the computing power requirement is by running the computing units at different power modes depending on the operations to performs. For example, a processor not performing any operation can be kept at sleep or hibernate mode to reduce power requirement. At the same time it is important to make enough computing units available so that the required computation can be performed. In this regard, server provisioning and consolidation have been a well known and widely used approach in modern data centers. For example, Freon-EC is an extension to the Freon power-aware management software which adds power control [25,26]). In the case of Internet data centers, there are server provisioning schemes [23] that estimate the anticipated workload and use a small *active server set* while suspending the remaining servers. Similar concept exists in the domain of wireless communication where the wireless radio is turned on and off depending on whether there is any communication to be performed or not, respectively. For example, radio sleep scheduling has been considered for BSNs [8,9].

– *Non-computing system management:* Power requirement by the computing units is often complimented with the requirement from some associated non-computing processes. For example, the cooling power requirement in data centers is driven by the heat dissipated to run computing workloads in the servers. Sufficient cooling of the data center is needed to maintain a safe temperature (often determined by the redline temperature as indicated by the manufacturer) for server longevity (which in turn is an essential factor for sustainable computing from equipment recycling perspective as indicated in Section 2.2). Further, the scavenging of power from the non computing processes, e.g., solar power scavenging through photo-electric effect, can determine how much power requirement can be sustained (as per Figure 1).

(iii) **Scavenging energy from different sources:** Apart from reducing the power requirement, one other complimentary option for energy-sustainable computing is to increase the power available. This requires identification of different potential energy sources and investigating different ways to scavenge energy from these sources. Roundy et. al. [6] and Paradiso et. al. [2] provide a comprehensive list of energy scavenging from body heat,



(a) Data centers



(a) Body Sensor Networks (BSNs)

Fig. 3. Example CPSs.

sunlight, ambulation, vibration, respiration, and so on. Recent work by Sharma et.al. at the HP labs [27,28] has shown how the cow manure from dairy waste can be used to power data centers.

This paper investigates energy-sustainable computing for CPSs. Before continuing in these directions in Section 3, the following subsection describes sustainable computing from equipment recycling perspective.

2.2. *Equipment recycling perspective*

From equipment recycling perspective, sustainable computing is defined as the reusability and longevity of the computing units. In this regard, the following are the different research directions:

(i) **Maintaining safe operating condition:** One way to achieve the longevity of the computing equipment is by maintaining safe operating condition. For example, as mentioned previously, the data centers need to ensure a operating temperature within the equipment redline temperatures. This is particularly important to increase (or maintain) the manufacturer specified Mean Time Before Failure (MTBF), and hence minimize the requirement for replacing the equipment. A common practice over the years in data centers has focused on provisioning the cooling for worst case temperature scenarios [22,29,30], thus undesirably consuming high energy. More recent approaches have focused on dynamic control of the cooling units depending on the variation of the generated heat in the data center [24].

(ii) **Designing sustainable computing platforms:** Computing platforms have been developed to: (i) ensure sustainable and energy-efficient operations [31,32], and (ii) using eco-friendly materials [33–35].

The following section describes energy-sustainable computing of CPSs.

3. Energy-Sustainable Computing in CPS

3.1. *CPS*

There are two types of components in any CPS: (i) computing and (ii) physical. Figure 3 shows the deployment of the two representative CPSs: data centers (Figure 3(a)) and BSNs (Figure 3(b)). For example, data centers use raised floors and lowered ceilings for cooling air circulation, with the computing equipment (i.e. the computing components) organized in rows of racks arranged in an aisle-based layout, with alternating cold aisles and hot aisles. The cooling of the data center room is done by the Computer Room Air Conditioner (CRAC), which supply cool air into the data center through the raised floor vents. The cool air flows through the chassis inlet and gets heated up by convection from the computing equipments and hot air comes out of the chassis outlet. The hot air goes to the input of the CRAC which cools it down. The CRAC along with the hot and cold air constitute the physical components.

Body Sensor Network (BSN) is a network of heterogeneous set of medical devices that can sense, actuate, compute, and communicate with each other through a wireless channel. The architecture of BSN is shown in Figure 3(b). The nodes (i.e. the devices) in a BSN can be broadly classified into two categories: 1) *worker nodes*, which are implanted or wearable medical devices with a low computing capability interfaced with sensors, actuators, and wireless transceivers (e.g. a PPG sensor interfaced with TelosB motes); and 2) *base station*, which has higher computation and communication capabilities (e.g. PDA) to disseminate and collect information to and from the worker nodes, respectively. Each node in a BSN has a set of neighboring nodes with which it can communicate through an one-hop wireless link. The worker nodes, base station and the inter-communication among them form the computing components, whereas the human body along with its physiology form the physical component.

In general, Figure 4 shows the functional architecture of a CPS. The computing components are responsible for executing the workload of a CPS. For example, in a data center, jobs submitted by the user constitute the workload. For a BSN, workload includes sensing and communication of physiological signals. Both the computing and the physical components are powered by a set of energy sources, which themselves can be part of the physical environment. In a CPS, there are strong interactions between the computing components and the physical environment. The interactions can be bidirectional. Interactions from the computing components to the physical environment normally involve controlling of certain non-computing processes. An example of such interaction can be the control of blood glucose levels by an insulin pump. Interactions from the physical environment to the computing components can be of two types: direct and indirect. While indirect interactions mean adapting the computing operations depending on the behavior of the non-computing processes, direct interactions put an immediate dependency of the computing components on the the physical environment (e.g., energy scavenging from replenishable sources in the physical environment). Ideally, *an **energy-sustainable** CPS needs to minimize the energy requirements for its operations or at least make sure that the*

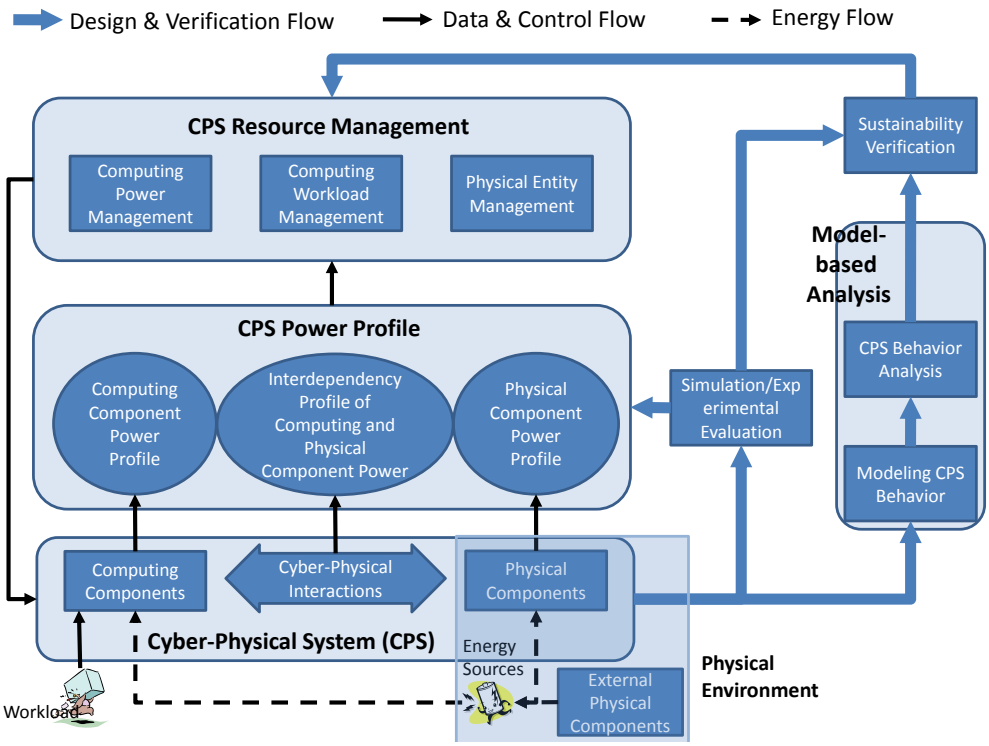


Fig. 4. CPS functional architecture.

energy requirements do not exceed the available energy from the green sources in the physical environment. Therefore, it is important to understand the trends and dependencies of the power consumption in all the CPS components.

3.2. *CPS power characteristics*

The power consumption of a CPS, p^{cps}, depends on the power consumption of both computing and non-computing (i.e. physical) components. The total power required by the computing components is referred as the *computing power*, whereas the total power required by the non-computing components is referred as the *non-computing power*. Based on these power requirements, p^{cps} can be given as follows:

$$p^{cps} = \text{computing power} + \text{non-computing power} = \sum_{i \in C} p_i^c + \sum_{j \in NC} p_j^{nc}, \tag{1}$$

where C and NC are the sets of computing and non-computing components, respectively, p_i^c is the power consumption of computing component i, and p_j^{nc} is the power consumption of the non-computing component j. The dependencies of both p_i^c and p_j^{nc} are described below.

(i) *Computing power*: The power consumption of any computing component, $i \in C$, depends on two factors: (i) the workload being executed at i (i.e. w_i); and (ii) the power mode of i (i.e. β_i). If the function $\mathcal{G}_i^c : \mathcal{W} \times \mathcal{M} \rightarrow \Re$ returns the power consumption, p_i^c, of computing component i (where $\mathcal{W}$ is a set of all possible workloads and $\mathcal{M}$ is the set of all possible computing power modes) then p_i^c can be obtained as follows:

$$p_i^c = \mathcal{G}_i^c(w_i, \beta_i), \tag{2}$$

where $w_i \in \mathcal{W}$ and $\beta_i \in \mathcal{M}$.

(ii) *Non-computing power*: The power consumption of non-computing components (e.g., cooling unit in data centers) depends on a set of non-computing parameters (e.g., air temperature at the input of cooling unit) of the component and a property set of the non-computing process (e.g., amount of heat extracted during the cooling process) performed by the component. If these sets are denoted by $\mathcal{P}$ and $\mathcal{S}$, respectively, then p_j^{nc} of any component $j \in NC$ can be given as:

$$p_j^{nc} = \mathcal{G}_j^{nc}\big(\mathcal{P}, \mathcal{S}\big), \tag{3}$$

where $\mathcal{G}_j^{nc}$ is a function such that $\mathcal{G}_j^{nc} : \mathfrak{R}^{|\mathcal{P}|} \times \mathfrak{R}^{|\mathcal{S}|} \rightarrow \mathfrak{R}$.

3.2.1. *Impact of cyber-physical interactions on CPS power consumption*

The non-computing parameter set, $\mathcal{P}$, is affected by the computing operations because of the cyber-physical interactions. For example, the air temperature at the cooling unit depends on the heat generated in a data center room; which itself depends on the power consumed by the computing servers. Further, this dependency has a spatio-temporal dynamics. For example, at a given time instant the impact from a server at one location (to the input air temperature of the cooling unit) may be different than that of a server with same power characteristics but at different location. This variation is driven by the recirculation pattern of the air in the data center room. Similarly, at a given instant the control logic of an insulin pump may have different effect in different parts of the body depending on the drug diffusion rate. Further, at a given location the impact may vary with time. We denote as $F : (\mathcal{W} \times \mathfrak{R}^3)^n \rightarrow \mathcal{F}^{|\mathcal{P}|}$ the impact function that maps the workload running on a computing component at an Euclidean location to the non-computing parameters in $\mathcal{P}$, where n is the total number of computing components and $\mathcal{F}$ is the set of all possible non-computing parameter functions $f(t, x, y, z)$.

3.2.2. *Function characterization requirement*

For any CPS, it is required to characterize the $\mathcal{G}_i^c$ and $\mathcal{G}_j^{nc}$ functions $\forall i \in C$ and $\forall j \in NC$, respectively, as follows:

– The *characterization of* $\mathcal{G}_i^c$ *function* can be performed based on experimental profiling with different workload and computing modes.

– The *characterization of* $\mathcal{G}_j^{nc}$ *function* has three basic steps:

  · identification of the different elements in the sets $\mathcal{P}$ and $\mathcal{S}$,

  · characterization of the impact function F, and

  · experimental profiling of $\mathcal{G}_j^{nc}$ with different workload and computing modes.

Given the power characteristics of CPS, the following subsection discusses how different resource management strategies can impact the CPS power consumption.

3.3. *CPS resource management*

Given a deployment of the computing components, the CPS power consumption depends on three major types of resource management decision making:

(i) *workload management*, which determines the amount of workload in each computing component, thus affecting the computing power (as per Eq. 2);

(ii) computing *power management*, which determines the power modes of the computing components, thus affecting the computing power (as per Eq. 2); and

(iii) *non-computing component management*, which determines the property set $\mathcal{S}$ of the non-computing processes, thus affecting the non-computing power (as per Eq. 3).

All the management decision making further has to ensure that the service requirements (e.g., job throughput and turnaround time in data centers) meet the user expectations. The determination of workload at each computing component further affects the set of non-computing parameters $\mathcal{P}$ as per the impact function F; this parameter set in turn affects the non-computing power as per Eq. 3. Thus, workload management can have indirect effect on the non-computing power because of the cyber-physical interactions. Such effects impose fundamentally different considerations in the decision making of workload management in order to reduce energy consumption in CPSs. *A more coordinated resource management is required where the workload management and non-computing component management need to be aware of their impact on the CPS power consumption*.

3.4. *Resource management algorithm classification*

Resource management algorithms, which are aware of the non-computing processes and the impact of the computing units to these processes, can be classified based on: (i) support and knowledge of workload; (ii) assumptions on non-computing process; and (iii) support of power management. The following subsections discuss these different categories and identifies the distinctive algorithm classes in each of these categories.

3.4.1. *Support and knowledge of different workload characteristics*

As described in the previous section, workload plays an important role in the CPS power requirements. Workload can be categorized based on their arrival and execution profiles. Workload can execute for a long duration in the scale of seconds, minutes, hours, or even days (e.g., long running scientific jobs in HPC data centers, signal processing and cryptographic operations in sensors); or they can be a stream of short requests (in the scale of milliseconds) such as web transactions and database queries. Further, the arrival of any of these workload can be periodic or aperiodic. For example, in a BSN the workload on the sensors are mostly periodic in nature requiring the same operations (e.g., sensing, communication, and cryptographic operations) in predefined periods. An aperiodic workload on the other

hand has no predefined period of arrival but arrives in an ad hoc manner.

Depending on the knowledge of the workload, a workload management algorithm can be online and offline. An offline algorithm has complete future knowledge of the workload arrival and execution profiles, whereas an online algorithm makes decision based only on the current knowledge and does not have any information on the future workloads. A prediction mechanism can be employed regarding the future workload. However, the accuracy of the prediction mechanism hugely depends on the repeating pattern of the workload. As such, in the rest of the paper we assume that any algorithm that only supports periodic workload (with the exact repetition of arrival rates and execution times) is inherently offline in nature since it can use the past information of the workload as a future knowledge. Any workload that is not exactly periodic but has some periodic pattern (e.g., web requests in Internet data centers) can be thought of as aperiodic; however online algorithms can have better estimation of the future for such workload making them closer to the offline counterparts.

Resource management algorithms can be classified depending on the different types of workload they support and the knowledge of the workload's arrival sequence. In this regard, there are three types of classification categories: (i) workload arrival profile (periodic or aperiodic), (ii) workload execution profile (long running or short running), and (iii) workload knowledge (online or offline). Table 1 summarizes these classes as a support matrix indicating the different workload arrival and execution profiles supported and whether the algorithms are online or offline. These different classes are further described as follows:

– *Specific algorithms*: This class of algorithms supports only a particular type of workload arrival and execution profile. Further these algorithms can be either online or offline and are not flexible in their decision making when future workload information becomes available or is not available, respectively. The algorithms are named depending on the specific cases supported as shown in Table 1. For example, oNline algorithms supporting Long running and Aperiodic workload are referred as LAN algorithms. Note that specific algorithms suffer from general applicability to different workload since the awareness of the CPS power, p^{CPS}, is based on a specific knowledge of the workload. An example algorithm in the LAF class is data centric routing algorithms for BSN that attempts to minimize the communication energy [36]. The algorithms are devised apriori and assume the knowledge of the workload and are hence offline. They are mainly used for medical applications featuring long running aperiodic signal processing jobs. However, many workload for medical monitoring can be periodic in nature. Further, as mentioned previously, since periodic workload (where the arrival and execution of workloads are repeated) are inherently offline in nature there is no classification made for online algorithms supporting such workload. An example algorithm for LPF class supporting periodic offline workload is a MAC protocol for BSNs [37]. An important aspect in designing offline algorithms for aperiodic workload is the higher complexity because of the higher knowledge of the workload.
– *One✶ algorithms*: These algorithms support all possible cases in any one of the classification categories mentioned

Table 1
Classification of CPS resource management algorithms based on their support and knowledge of workload characteristics. The capitalized letters in the sub-categories of supported workload, supported workload arrival, and workload knowledge are used for the abbreviated nomenclature of the algorithm classes. The symbol '✶' is used to denote that all cases in a category is supported. The last two columns show the specific algorithms in different classes from the two representative example of BSN and data centers (a '–' means no algorithm in the corresponding class).

Algorithm Classes		Supported Workload		Supported Workload Arrival		Workload Knowledge		Algorithms for BSN	Algorithms for data centers
		Long running	Short running	Periodic	Aperiodic	oNline	oFfline		
Specific Algorithms	LAN	✔			✔	✔		–	–
	SAN		✔		✔	✔		–	–
	LAF	✔			✔		✔	Data-centric routing [36]	–
	SAF		✔		✔		✔	–	–
	LPF	✔		✔			✔	BSN MAC [37]	–
	SPF		✔	✔			✔	–	–
One✶ Algorithms	✶AN	✔	✔		✔	✔		–	–
	✶AF	✔	✔		✔		✔	Minimum Communication [38]	–
	✶PF	✔	✔	✔			✔	P-M, NP-M	–
	LA✶	✔			✔	✔	✔	–	–
	SA✶		✔		✔	✔	✔	–	–
	L✶F	✔		✔	✔		✔	–	SCINT [11], TASA [39]
	S✶F		✔	✔	✔		✔	–	–
	L✶N	✔		✔	✔	✔		–	ECTC, MaxUtil [40], MinHR [41], Proportional-Share [42]
	S✶N		✔	✔	✔	✔		–	GentleCool [43]
Two✶ Algorithms	L✶✶	✔		✔	✔	✔	✔	–	FCFS-LRH,EDF-LRH FCFS-XInt, EDF-XInt FCFS-HTS, EDF-HTS [11]
	S✶✶		✔	✔	✔	✔	✔	–	TAWD,TASP+TAWD [23]
	✶A✶	✔	✔		✔	✔	✔	–	–
	✶✶N	✔	✔	✔	✔	✔		–	Mercury [25]
Three✶ Algorithm	✶✶✶	✔	✔	✔	✔	✔	✔	NP-NM, drug delivery, reconfiguration [44]	–

previously. The symbol '✶' is used to denote that it can support all the possibilities in a category. Following are the different types of one✶ algorithms:

· ✶*AN* and ✶*AF* : Online algorithms that support aperiodic arrival of both long running and short running workload are referred as ✶AN class of algorithms. Similarly, offline algorithms supporting aperiodic arrival of both long running and short running workload are referred as ✶AF algorithms. An example ✶AF algorithm for BSNs incorporates more computation in the sensors to minimize communication which can consume orders of magnitude higher energy than computation [38]. A major challenge for these algorithms is to provide unified decision making for stream of short requests and individual long running jobs.

· ✶*PF* : Offline algorithms supporting periodic arrival of both long running and short running workload are referred as ✶PF class of algorithms. Since periodic workload are inherently offline in nature, the online version of these algorithms are not categorized. The challenges to address for the ✶PF are similar to that of the ✶AF algorithms. Since the applications in BSNs are mostly periodic in nature (e.g., periodic monitoring of the physiological signals), resource management algorithms for BSN fall in this class (Table 1). These algorithms will be discussed in further detail in Section 5.3.

· *LA*✶ and *SA*✶ : Algorithms supporting aperiodic long running workload with or without the complete knowledge of future workload are referred as LA✶ algorithms; similar algorithms supporting only short running workloads are called SA✶ algorithms. The principal challenge for these algorithms is to make independent decisions based on whatever workload information is available. These algorithms can manifest higher benefits when more information is available on the workload.

· *L✶F, S✶F, L✶N, and S✶N*: Algorithms supporting long running aperiodic or periodic workload with complete knowledge of future workload are referred as L✶F algorithms; similar algorithms supporting only short running workloads are called S✶F algorithms. TASA and SCINT algorithms in the L✶F class have been developed for data centers running long running High Performance Computing (HPC) jobs. The principal challenge for these classes of algorithms is to support both aperiodic and periodic workload with same decision making. This challenge also persists for the online versions of these classes of algorithms, i.e L✶N and S✶N. Several algorithms have been developed in these classes for data centers (see Table 1).

– *Two✶ algorithms*: These algorithms support all possible cases in any two of the three classification categories mentioned previously. Following are the different types of Two✶ algorithms:

· *L✶✶* and *S✶✶* : Algorithms supporting both periodic and aperiodic long running workload irrespective of the future knowledge are referred as L✶✶ algorithms; similar algorithms support only short running jobs are referred as S✶✶ algorithms. The principal challenge is to make independent decisions based on whatever workload information is available. The problem becomes exacerbated since the workload can be either periodic or aperiodic.

Distinguishing the periodic workload and then being aware of that while doing decision making for the aperiodic workload is essential in these algorithms. Many resource management algorithms have been developed for data centers (Table 1). These algorithms will discussed in Section 4.5.

- ✶*A*✶ : Algorithms supporting both long running and short running aperiodic workload irrespective of the future knowledge are referred as the ✶A✶ algorithms.
- ✶✶*N*: Algorithms supporting both long running and short running periodic and aperiodic workload without any future knowledge are referred as the ✶✶N algorithms. Mercury software suite for data centers falls under this category.

– *Three✶ algorithms*: These algorithms support all the possible cases of the three classification categories mentioned previously. The goal for a CPS designer is to employ a Three✶ algorithm. ✶PF class of resource management algorithms for BSNs are extended to support aperiodic workload in an online fashion to design Three✶ algorithm. Further, any automated drug delivery is supported both periodic and aperiodic online manner. These delivery can be performed for long duration or short duration. So any drug delivery falls under Three✶ algorithm class (Table 3.4.1). These algorithms will be discussed in further detail in Section 5.3. Another Three✶ algorithm for online reconfiguration of BSNs [44] allows automatic redistribution of computation and communication among sensors and the base station.

Apart from the workload based classification schemes, the resource management algorithms can be classified based on the assumption on non-computing state and support of the power management.

3.4.2. *Assumption on non-computing state*

Assumptions on the non-computing state determines the behavior of the impact function F. In this regard, there can be two states: steady-state and transient. A steady-state behavior of the non-computing process assumes that the parameter set $\mathcal{P}$ is stabilized to a particular value (e.g., the steady-state temperature at the input of the cooling unit in a data center). A transient behavior is more dynamic in nature where the continuous variation of the parameters in $\mathcal{P}$ over time and space is considered. Transient behavior assumptions often provide more accurate predictions. It should be noted here that transient behavior encompasses steady-state for certain limiting conditions.

3.4.3. *Power management support*

Power management determines the mode of operation β_i of each computing component i. These modes can impact the computing power (see Eq. 2). Resource management algorithms are classified based on whether a constant mode is assumed or power management, i.e. dynamic variation of the modes in the computing components is performed. Overall notational convention for the algorithm classes follows the three letter notation (for workload based classification) followed by hyphen and two letters denoting the non-computing state assumption and power management support

(in sequence). For example, a LAN class of algorithm in Table 1 that assumes Steady-state behavior and does Not perform power management is referred as LAN-SN algorithm. For algorithms that consider transient behavior, a '✷' is used since transient behavior encompasses steady-state behavior. Similarly, '✷' is used when power management is employed. For example, a LAN class of algorithm that assumes transient behavior and supports power management is referred as LAN-✷✷ algorithm.

Given the classification and the notations of different classes of resource management algorithms, the goal should be to design a Five✷ algorithm for resource management, i.e. the algorithm should: (i) be of Three✷ class when categorized based on workload support (and knowledge), (ii) assume transient behavior of the non-computing processes, and (iii) support power management. Sections 4 and 5 discuss the challenges in designing a Five✷ algorithm for specific CPSs such as data centers and BSNs, respectively. Also, the pros and cons of various algorithm classes will be discussed.

3.5. *Verification of CPS for energy-sustainability*

Verification of CPSs in terms of sustainability involves: (i) determining the energy consumption for long-term CPS operations; and (ii) ensuring that the energy consumption is within the energy supplied from the sources in the environment. An ideal way to perform the verification is through experimentation on actual deployment of a CPS or through accurate simulation of the system. Simulation based verification is widely used since the resources required to build experimental test-bed may not be affordable. Both simulation and experimentation can also be used to characterize the various functions such as the $\mathcal{G}_i^c$, $\mathcal{G}_j^{nc}$, and F functions (see Section 3.2). Sections 4 and 5 discuss how these functions can be characterized through real measurement and simulation based profiling for data centers and BSNs, respectively. Further, in many critical CPSs such as BSNs, verification may be required at the design time (without real deployment). Early design time verification has two advantages: (i) it avoids creating real test-scenarios putting lives at risk; and (ii) it provides a way to guarantee and certify the CPS behavior. Such certification methodology can be useful for the various regulating agencies (e.g., FDA approval of the medical devices). One way to perform early design time verification is through Model Based Engineering (MBE). MBE is the method of developing behavioral models of real systems and analyzing the models for requirement verification. There are two main phases in MBE: 1) model development, and 2) model analysis. In the model development phase, a set of *expected properties* of the system is determined from the *system requirements*. An abstract modeling is further performed that generally involves capturing appropriate parameters whose variations can reflect the *system behavior*. Mathematical analysis (*model analysis*) is then performed on the abstract model to *evaluate* the expected properties and *verify* the system requirements. In this paper, we discuss MBE in verifying the CPSs' energy-sustainability at design time.

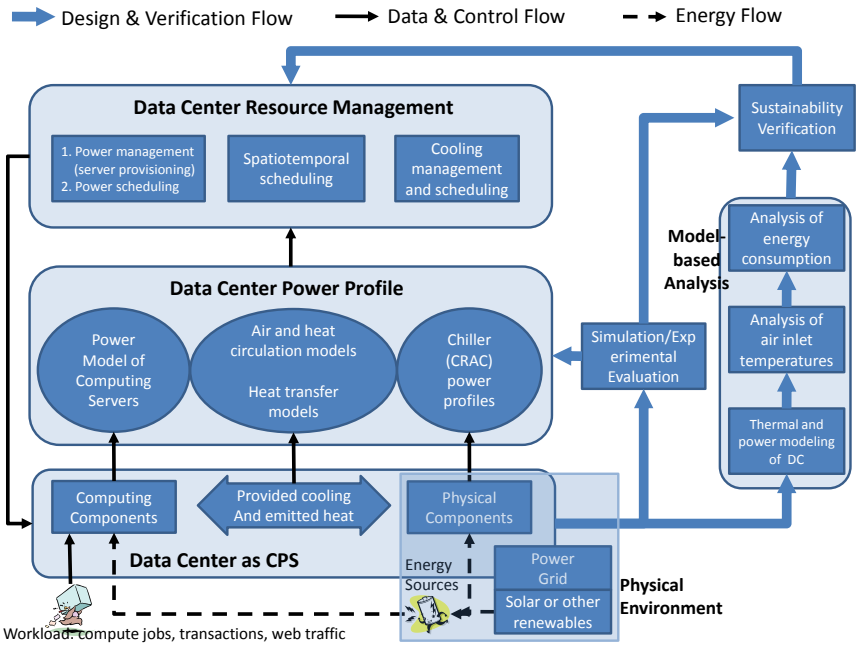


Fig. 5. Data centers can be modeled using the abstract holistic view of CPS as in Fig. 4.

4. Resource management to ensure energy-sustainability of data centers

A data center is a manifestation of the CPS abstraction laid out in the previous sections. A data center consists of computing components (servers), non-computing components (power distribution and chillers), an energy supply most of which comes from the power grid, and of course it is immersed in a practically closed environment in which physical thermal phenomena take place, including the cooling cycle of the servers. Figure 3a shows a typical layout of a data center along with the air input to and output from the cooling unit (CRAC). The mapping between a data center and the CPS functional architecture (in Figure 4) is shown in Figure 5.

4.1. *Characterizing $\mathcal{G}_i^c$ function*

The $\mathcal{G}_i^c$ function describes the power consumption of the equipment with respect to their utilization. Power profiling of computing equipment is a standard practice and there are several well-established methodologies to documenting the power consumption of a system with respect to its utilization.

Power profiling usually yields a "power curve" which consists of averaged power measurements at sample utilization points (idle, 10%, 20% etc). There are research studies that try to map the yeilded curve to a polynomial, most usually a linear function. Although a linear model is not always accurate [45] it has been extensively used due to its simplicity.

Under a linear model, a compute server *i* consumes power with respect to its CPU utilization as follows:

$$\mathcal{G}_i^c = a_i U_i + b_i,$$

where b_i is the idle power of the server and a_i is the slope of the line to the maximum power. The term U_i denotes the CPU utilization ($0 \leqslant U_i \leqslant 1$). The total computing portion of the data center power is the sum of the individual components:

$$P_{\mathrm{c}} = \sum i\mathcal{G}_i^c = \sum i a_i U_i + b_i$$

4.2. *Characterizing $\mathcal{G}_j^{nc}$ function*

The $\mathcal{G}_j^{nc}$ function describes the power consumption of the non-computing equipment with respect the rest of the data center. For the case of the power distribution equipment, then its power consumption depends on the power drawn by the computing equipment, i.e. $\mathcal{P} = P_{\mathrm{c}}$. If we assume a constant efficiency ratio of ϕ, then the power consumption of the power distribution equipment is:

$$\mathcal{G}_j^{nc}(\mathcal{P}, \mathcal{S}) = \mathcal{G}_j^{nc}(\{P_{\mathrm{c}}\}, \{\alpha\}) = \alpha P_{\mathrm{c}}.$$

For the case of the *chillers*, a.k.a. *computer room air conditioners* (CRACs) or *heating ventilation air conditioners* (HVACs), the power consumption depends on the input heat to be extracted divided by the *coefficient of performance* (CoP) of the chiller at the operating temperature

$$\mathcal{G}_j^{nc} = P_{\mathrm{c}} / CoP(T_{\mathrm{input}}),$$

where T_{input} is the input (sensed) temperature. The coefficient of performance denotes the cooling efficiency of the chiller, and ideally is governed by the Carnot efficiency:

$$CoP = \frac{T_{\mathrm{input}}}{T_{\mathrm{input}} - T_{\mathrm{return}}}.$$

For heat extractors that remove a roughly constant amount of heat, this efficiency translates into a quadratic curve (Figure **??**(a)). From the above, we can denote $\mathcal{G}_j^{nc}$ as

$$\mathcal{G}_j^{nc}(\mathcal{P}, \mathcal{S}) = \mathcal{G}_j^{nc}(\{T_{\mathrm{c}}, T_{\mathrm{input}}\}, \{CoP(T)\})$$

Moreover, CRACs feature multiple modes of cooling, i.e. they can cool at different compression ratios [2]. The different compression modes are triggered by a thermostat which senses the input temperature and compares to its trigger point. Assuming the same constant flow, the trigger point can be easily translated into input heat (Figure **??**(b)).

Calculation of T_{input} can be done using the thermodynamic formula

$$T_{\mathrm{input}} = \frac{c_q \rho f}{P_{\mathrm{c}}} + T_{\mathrm{output}},$$

[2] The standard cooling technology is vapor-compression cooling, where the coolant vapor is compressed at one phase of the cooling cycle and then decompressed to produce the cooling effect

where c_q is the specific heat of air (at T_{input}), ρ is the mass density of air and f is the flow of air through the CRAC. Using the above equation, we can replace T_{input} on the x axis with the input heat (Figure **??**a).

4.3. *Identifying $\mathcal{P}$, $\mathcal{S}$ and $\mathcal{F}$*

In the discussion above, $\mathcal{P}$ has been defined as:

$$\mathcal{P} = \{P_{\text{c}}, T_{\text{input}}\}.$$

Although P_{c} is easy to be estimated, T_{input} requires the knowledge of the heat distribution in the room along with the air flow patterns. In general, T_{input} can be expressed as a weighted sum of the air temperatures from the heat sources in the room, namely the servers and the CRACs:

$$T_{\text{input}} = w_{c1}T_{\text{out},c1} + w_{c2}T_{\text{out},c2} + \ldots + w_{cn}T_{\text{out},cn} + w_{\text{nc}1} + T_{\text{out,nc}1} + \cdots + T_{\text{out,nc}m},$$

if there are n computing components and m non-computing components.

The parameter T_{input} in $\mathcal{P}$ not only it has a quantitative role in determining the CoP of the CRAC, but it also denotes the *exergy* of the incoming heat, i.e. the quality of the heat. The exergy determines how easy it is for the heat to be converted into useful work, which can in turn be used to cool down the data center or produce electricity. For example, heat supplied at around 98 °C is good to drive an *absorption chiller*, while heat at lower temperatures (circa 65 °C) can drive an *adsorption chiller* albeit at a lower CoP.

A factor against running the data center at high exergy temperatures is the *redline* temperatures specified by each equipment's manufacturer. For example, most computing servers have a redline air-inlet temperature of 35 °Cor less. The CRACs have to be set to an input temperature such that the air-inlet temperatures at the equipment do not exceed the respective redlines.

To estimate the air-inlet temperatures, we use an extension of the $\boldsymbol{w}$ vector. This extension is the *heat recirculation* matrix $\boldsymbol{D}$ whose each element d_{ij} denotes by how much the temperature of an air inlet at server j is affected by the heat produced at the server i. Using this matrix and the $\mathcal{G}_i^c$ at each server, we can compute the temperature vector at the air inlets of the data center equipment:

$$\boldsymbol{T}_{\text{inlet}} = \boldsymbol{D}\langle G_i^c \rangle.$$

One of the elements of the $\boldsymbol{T}_{\text{inlet}}$ vector is the CRAC's inlet temperature.

Considering the above, a resource management algorithm's objective (see Section 4.5 below) would be try and allocate the workload and configure the computing equipment (power mode) and CRAC equipment (power mode $\mathcal{S}$) in such a way as to maximize the exergy and keep all temperatures below the redline.

4.4. *Profiling for $\mathcal{G}_j^{nc}$ function*

The $\mathcal{G}_j^{nc}$ function can be profiled usually by experimentation. The non-computing component can be equipped with power meters and thermometers, and induced with varying thermal workload. Depending on the technology used, the instrumentation of the experiment could be complemented with flow meters to measure air flow or chilled water supply, or even sunlight intensity if the equipment uses solar power.

4.5. *Resource management classes*

Traditionally, resource management in data center has been that of scheduling and distributing the workload. There are relatively two classes of data center clusters: those that service batch-based and relatively long-running workload units (jobs), with a time sensitivity of minutes or hours, and those that service short workload units (transactions), with a time sensitivity of a couple of seconds or less. Due to the difference in nature of the workload, data center management algorithms fall into one of the algorithm classes in Section 3.4, i.e. under one of L✶✶, S✶✶, L✶F, S✶F, L✶N, and S✶N classes.

In the batch-oriented data centers, jobs may spend considerable time in the queue before they get serviced. In that manner, scheduling algorithms combine a temporal placement logic (temporal scheduling) and a spatial placement logic (spatial placement or server assignment). An example of spatial-only placement logic is that of Xint [46] which assigns jobs to servers in such a way as to minimize the maximum T_{input}. An example temporal algorithm is first-come first-served (FCFS) with back-filling of jobs that fit into available servers.

Although there may be periodicity of workload, there is virtually no algorithm that assumes some form of periodicity. Example L✶✶ algorithms include FCFS-LRH, where LRH stands for least recirculated heat, FCFS-XInt, EDF-LRH and EDF-XInt, where EDF is a earliest-deadline first ordering of the arrived workload units [11]. These algorithms assume steady-state behavior and can also be integrated with power management. An L✶F algorithm is SCINT (**sc**heduling to minimize cross-**int**erference) [11], which assumes an offline knowledge of the arrived jobs.

In the S✶✶ class, there is virtually no queuing of the workload units, they are directly passed to the compute nodes for servicing. An example of S✶✶ algorithm is *load balancing* (LB), a.k.a. *equal load balancing* (ELB), which stochastically or by round-robin distributes the transactions among the servers. Thermal-aware workload distribution (TAWD) is another S✶✶ approach, which distributes the workload in such a way as to reduce the work done by the cooling units.

Combination of scheduling with power management is a fairly recent trend in resource management in data centers. SCINT, can be considered an L✶F-S✶, because it can produce a *power schedule* of what systems to turn off and when to turn them off, under a steady-state physical model. FCFS-HTS (HTS stands for *highest thermostat setting*) and

EDF-HTS [24] algorithms are L✶✶-✶N, because they assume a transient model and do not have a power management scheme. On the other hand, TAWD combined with *thermal-aware server provisioning* (TASP/TAWD) is a S✶✶-S✶ algorithm, which makes a prediction of the workload intensity for a future workload and suspends a number of servers.

Other algorithms from the literature include TASA, which is a thermal-aware workload placement algorithm and falls under the L✶F-SN class. MinHR is a workload placement algorithm that considers the thermal impact of the operation of servers in the room [41]. However, it does not require offline knowledge of the workload and falls under the L✶N-SN class. Mercury is a power management software suite that adjusts the power of servers when they overheat [25]. It can support different workload arrival and execution patterns and falls under the ✶✶N-S✶ class. ECTC and MaxUtil are workload scheduling algorithms that try to consolidate tasks onto servers [40]. The basic difference of this is the assumption of non-exclusiveness between tasks and servers. These algorithms fall under the L✶N-✶✶ class. Proportional-Share in the Libra management software is a scheduling algorithm for assigning tasks to computers [42] and falls under L✶N-✶N class. GentleCool [43] is a scheduling algorithm that decides on the CPU share distribution among virtualized machines on a physical computer. This algorithm falls under the S✶N-SN class.

All these algorithms intend to reduce the power requirement of the data center. However, none are designed to use green energy sources. As such, from the definition of energy-sustainability in Section 2.1, none of these algorithms are completely energy-sustainable in nature. However, these algorithms need to be compared in terms of the grid power consumption (which is the manifestation of energy-sustainability when green sources are not available, as discussed in Section 2.1). In this regard, for long running workload, SCINT is the most energy-sustainable because of the offline knowledge of the workload during decision making and support of power management. L✶✶ class of algorithms has to compromise on energy-sustainability to support online decision making. For short running workload, TASP/TAWD is the most energy-sustainable since it supports power management when compared to TAWD.

5. Resource management to ensure energy-sustainability of BSNs

Figure 6 shows the instantiation of the generic CPS functional architecture in Figure 4 for the specific example of BSNs. The computing components in the BSN consist of sensor nodes or medical devices. The workload in BSNs are generally periodic and are known offline. For example, an infusion pump infuses drug into the human body following a fixed schedule. Also, a health monitoring application such as Ayushman [4] has a deterministic workload. In Ayushman, (as shown in the Figure 7) the sensors in the BSN sense physiological data for t_s seconds and store them in local memory. After t_s seconds they transfer the data to the base station in a single burst taking time t_{Tx}. Every communication is secured by encryption with a secret key which is established between each pair of BSN nodes. Key agreement between any two sensors is performed once in a day using the Physiological value based Key Agreement (PKA) [47] protocol each execution taking t_{PKA} time.

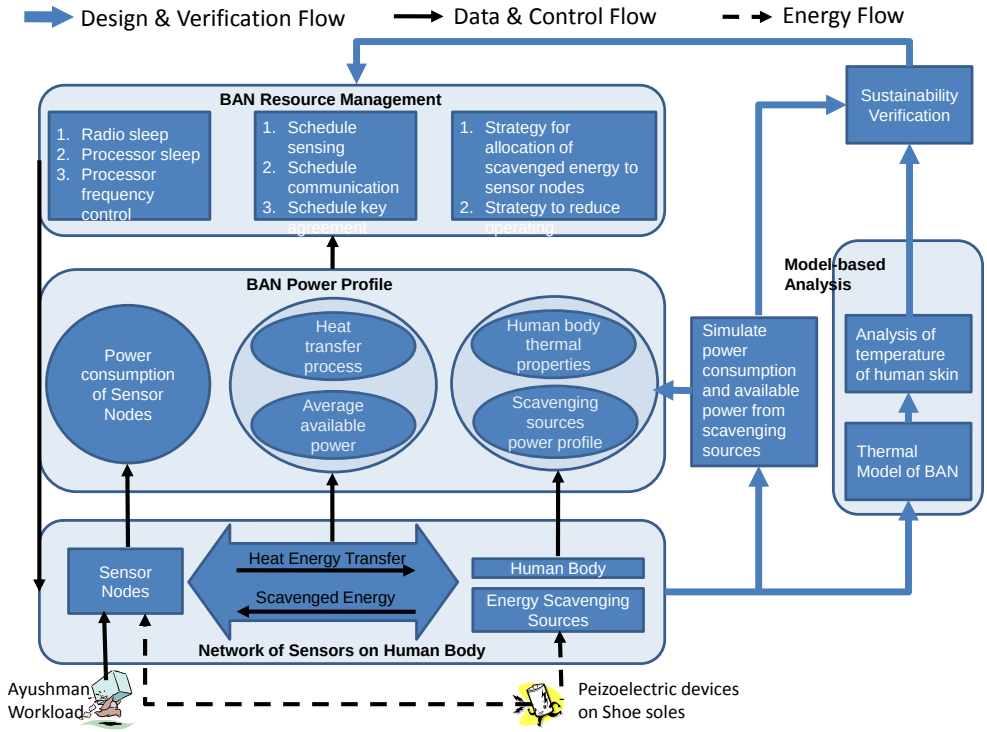


Fig. 6. BSNs can be modeled using the abstract holistic view of CPS as in Fig. 4.

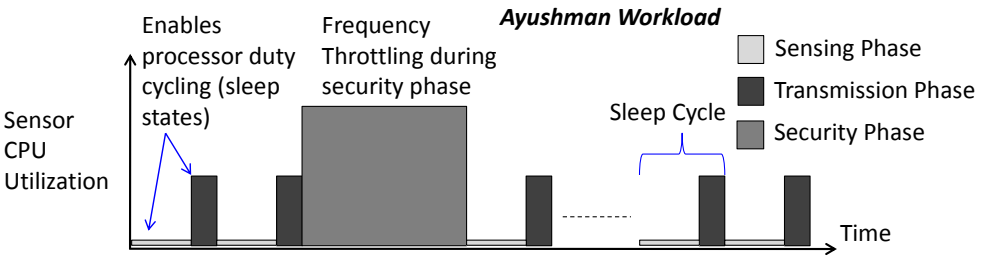


Fig. 7. BSN workload and application of power management strategies

Three types of non-computing units are considered in the BSN example: 1) human body, whose physiology is controlled by the nodes, 2) energy scavenging sources such as a peizo-electric device on shoe sole, which extract energy from the surrounding environment to provide operating power to the sensor nodes, and 3) medical actuators such as infusion pumps, which cause changes in human physiology according to commands from a computing system. The cyber-physical interactions between the computing and non-computing components are further three fold:

(i) Heat energy transfer from the sensor nodes to the human body, which causes rise in body temperature.

(ii) Electrical charge transfer from energy scavenging sources to the sensor nodes, which provide the operating power for the nodes.

(iii) Chemical energy interaction such as diffusion of drug in human blood caused by actuation decision from the

computing unit such as infusion pump controller.

Power profiling of a BSN involves three different aspects:

(i) **Sensor node power profiling:** The power consumption of the sensor nodes needs to be profiled for the different stages of the Ayushman workload . This profiling is performed for different power modes of the sensor node and for different power management strategies. Thus, this stage requires a feedback from the resource management stage as show in the Figure 6.

(ii) **Profiling of non-computing units:** Different types of profiling are required for the human body, the scavenging sources and the medical actuators:

  (a) The scavenging sources need to be profiled for the average amount of energy available per unit time.

  (b) The human body needs to be profiled for its thermal properties, which will govern its temperature rise.

  (c) The actuation sources need to be profiled for power dissipation due to the actuation process. For example, in case of infusion pump the drug infusion process requires insertion of needle in the human body. Friction of the needle with the tissue can lead to power dissipation.

(iii) **Characterization of the cyber-physical interactions:** Three types of cyber-physical interactions requires profiling of three different processes: 1) the heat transfer from the sensor nodes to the human body, 2) the charging of a sensor node with scavenged energy, and 3) drug diffusion processes.

The resource management stage consists of several power management strategies on the sensor nodes hardware, software and also on the non-computing energy scavenging sources.

(i) **Sensor hardware power management:** Power management strategies that are used for Ayushman are: 1) radio sleep scheduling, 2) processor level sleep mode scheduling, and 3) processor level frequency scheduling. Figure 7 shows when each of the strategies can be employed for the Ayushman workload. For example, radio and processor can be put to sleep during the sensing period but has to be active during the data transmission and PKA stages. Further, frequency control of the processor can be performed during the PKA stage to control its power consumption.

(ii) **Sensor software power management:** Since communication is more expensive than computation efficient scheduling of communication can achieve energy efficiency. Thus, in Ayushman we consider storing data locally and transmitting in bulk. This allows radio shutdown during the sensing phase and conservation of energy. Further, the PKA key agreement phase is also scheduled once in a day. The frequency of the key agreement determines the freshness of the key used in securing communication. Lower the frequency of the PKA execution lower is the energy consumption.

(iii) **Non-computing component management:** In case of the infusion pump controlling the frequency of infusion can reduce the power dissipated due to the actuation process.

Given the BSN, its power profile under the several management strategies, model based verification of its sustainability is performed. For this purpose, architectural model of the sensor node and the scavenging sources are developed to verify the sustainability. Further, to determine the thermal effects on the human body formal models were developed, which characterized the cyber-physical interactions.

5.1. *Characterizing $\mathcal{G}_i^c$ function*

The function $\mathcal{G}_i^c$ is obtained through experimental profiling of the sensor nodes. In case of a BSN the profiling experiments were performed for two different platforms: Intel Atom and TelosB motes based nodes. Intel Atom processor provides different modes of operation, which have different clock frequencies, while the TelosB motes only have a single operating mode. For the Atom processor, the power consumption for the most compute intensive operation in Ayushman, PKA, is experimentally obtained for different operating frequencies as shown in Table 2. In the table, percentage throttling means the percentage by which the operating frequency is reduced from the maximum. Further, the Atom processor support sleep modes where the power consumption is very low. Table 2 characterizes the function $\mathcal{G}_i^c$ for the two platforms. It can be clearly seen that $\mathcal{G}_i^c$ depends on the workload and the operating mode of the processor.

Table 2. Power Consumption of Atom for Ayushman workload (w_i)

Percentage Throttling	**Power Consumption (W)**
0	0.191
13	0.1864
25	0.17
37, 50, 62, 75	0.167
87	0.164

5.2. *Characterizing $\mathcal{G}_j^{nc}$ function*

In the BSN case two different examples are considered to explain the function $\mathcal{G}_j^{nc}$. In the first example, energy scavenging nodes are considered, which act as source of power while in the second case the infusion pumps are considered, which dissipate heat energy due to friction with human tissue. For each of these cases we identify the $\mathcal{P}$ and $\mathcal{S}$ sets in the following section.

5.2.1. *Identifying sets $\mathcal{P}$ and $\mathcal{S}$*

For the non-computing units that scavenge energy, the $\mathcal{P}$ set can represent the energy requirement of the computing components. For example in Ayushman the energy consumption of the BSN with d nodes can be computed from the Eq. 4.

$$E_{BSN} = d[\{(t_s)P_{sleep}^{atom} + t_{Tx}(P_{radio} + P_{active}^{atom})\}w + [t_{PKA}(P_{PKA}^{atom} + P_{radio})](\frac{d-1}{2})] \quad (4)$$

Note that $E_{BSN} \in \mathcal{P}$ depends on the computation workload as discussed in Section 3.2. The set $\mathcal{S}$ is a property of the non-computing component that does not depend on the workload. For the BSN it can represent the energy obtained from the scavenging sources. Table 3 [2] gives the power available from the scavenging sources and the expected

Table 3
Available Scavenging Power

Scavenging Source	Available Power (W)	Scavenge Time (Hrs)
Body Heat	0.1 - 0.15	24
Ambulation	1.5	2
Respiration	0.42	6
Sun Light	0.1	3

amount of time each scavenging source can operate. Each of these power and time values can be members of the set $\mathcal{S}$.

Equivalently, for the infusion pump, the infusion rate requested by the controller is a member of the set $\mathcal{P}$. The infusion rate is determined by a control algorithm [5], which attempts to maintain a constant drug level in the blood. Whenever drug infusion is requested by the control algorithm the infusion pump dissipated energy during the drug injection process. This energy dissipation is a member of the set $\mathcal{S}$.

5.2.2. *Identifying the impact function F*

In case of the energy scavenging example the impact function F can be the different between the elements in $\mathcal{P}$ and $\mathcal{S}$. If the difference is negative then the amount of scavenged energy is more than required. This indicates that the system is sustainable. If the difference is positive then it indicates that energy required is more than the available. Hence the system is unsustainable. Penne's bioheat equation [48] can be used as the impact function that relates the temperature rise in the human tissue because of the heat dissipated from the infusion pump.

5.3. *Resource management classes*

To ensure sustainability of the Atom based BSN three different approaches to communication scheduling and processor level sleep scheduling has been employed:

(i) with processor level sleep scheduling and communication (radio sleep) scheduling (P-M),

(ii) no processor level sleep scheduling but with communication scheduling (NP-M), and

(iii) without any processor level sleep scheduling or communication scheduling (NP-NM).

All the above mentioned strategies consider operation at the lowest frequency so as to achieve lowest power consumption. Figure 8 shows the number of BSN nodes sustained for 24 hours of Aysuhman operation using the different scavenging combination and for a particular design strategy. We consider combination of scavenging sources according to their applicability in real life situations. Combination of scavenging from body heat and respiration can be applied for monitoring of bedridden patients in hospital. Ambulation and sunlight can be used in military applications or in performance monitoring of outdoor sports like golf [49]. Ambulation and respiration can be used in case of athletes in training. Body heat and ambulation can be used for long term monitoring in a home environment. In the Figure 8 we

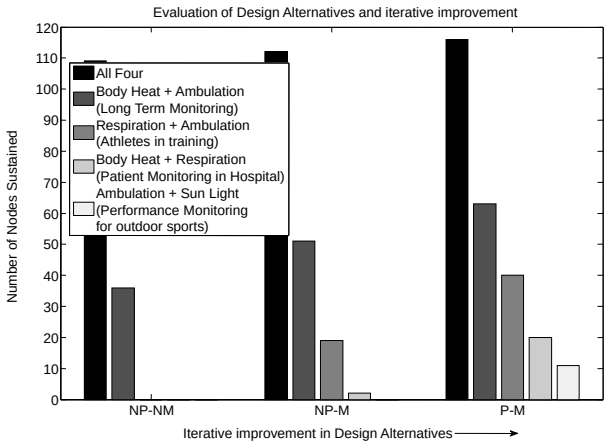


Fig. 8. Sustainability analysis results in terms of the number of computing units powered by the energy available from green sources (Section 2.1).

arrange the scavenging combination in order of highest to lowest available scavenged energy. The absence of some bars from the figure indicates that the corresponding scavenging combination cannot sustain even a single node for 24 hrs.

The Ayushman workload is highly periodic. In this regard the strategies P-M and the NP-M scheduling the communication and processor power modes in Ayushman are all offline. The schedules are predetermined and are optimized to achieve energy efficiency. These scheduling algorithms are aware of the workload characteristics. The Ayushman workload has both long running and short running jobs. The sensing job is short running and frequent job however the PKA execution between two sensors is long running but is done once in a day. Hence the P-M and NP-M scheduling algorithms for Ayushman are all ✶PF algorithms. However, the NP-NM scheduling algorithm works both online and offline and does not schedule the communication or change the power modes of the processor. It just blindly runs the processor in the lowest operating frequency. Hence it is also not dependent on the periodicity of the workload. Thus it is a Three✶ algorithm. However, from the Figure 8 we see that for certain combinations of the scavenging sources the NP-NM algorithm is not sustainable. Thus, it can be seen that a Three✶ algorithm may not be sustainable.

Further, the scheduling algorithms in Ayushman support power management. As in the Atom processor the lowest frequency of operation is chosen and the radio is shutdown whenever possible to save energy. All the algorithms at least control the operating frequency of the Atom processor to achieve energy efficiency and does not consider transient behavior of the physiology of human body. Thus, the P-M and NP-M algorithms are ✶PF-S✶ while the NP-NM algorithm is ✶✶✶-S✶. Making the algorithms aware of transient behavior of the human body is a challenging task and is an open problem.

In case of the infusion pump, the control algorithm is online as it calculates the required amount of drug infused as and when the operator commands are delivered. Further, it can operate in the offline mode also to maintain a

predetermined drug concentration. The control algorithm is applicable for both long term jobs, e.g. keeping a constant drug level for a long period of time, while it also control drug concentration during bolus requests, which are short term infusion requests by the human being. The infusion requests are generally periodic with infusion schedules. However, certain pumps also support intermittent bolus requests from patients. Thus, the workload can be both periodic or aperiodic. Thus the infusion control algorithm is a Three✶ algorithm. The control algorithm is aware of the transient behavior of the human body. It obtains feedback in terms of the current drug concentration in the human blood and then computes the future infusion rate so as to maintain the given drug concentration. However, the infusion control algorithm is not energy aware making it ✶✶✶-✶N.

6. Verification of CPS operations

This section provides an architecture level specification framework. The specification considers a CPS as a *global* collection of computing and physical components and the CPS is represented as a Global CPS (GCPS). A GCPS is a collection of distributed and networked cyber-physical subsystems, each of which consists of a single computing component (or node), referred to as a Local CPS (LCPS). The LCPS subsystem considers each computing node in the CPS as an isolated cyber-physical system enabling modeling and analysis of the interaction of individual computing node with the physical environment. An LCPS consists of two entities:

– *Computing unit*: Abstract model of the computing unit. This facilitates the modeling of the computing as well as the physical behavior of the computing nodes. In this regard, types of properties of the LCPS are defined: (1) *Computing property*, which characterizes the computing behavior (e.g., processor speed and available memory) and (2) *Physical property*, which characterizes the physical behavior of the computing component (e.g., power dissipation).

– *Physical environment*: This facilitates the modeling of the portion of the physical environment with which the computing unit interacts. Any type of interaction (intended or unintended) is modeled by the transfer of information (data or energy) between the computing unit and the physical environment and its corresponding effect is modeled by continuous equations. These equations are essentially the impact functions F (see Section 3.2.1). In this regard, two constructs are defined ROIm (unintended interactions) and ROIn (intended interactions) each of which contains: (1) *monitored parameter*, which are the non-computing parameters in $\mathcal{P}$ (e.g., temperature and available scavenged energy), which gets affected by the interactions, and (2) *region boundary*, the region of the control volume over which the monitored parameter varies (depends on the spatio-temporal equations governing the variation of the monitored parameter).

The interactions that are modeled using the ROIn and ROIm constructs are local to a single computing unit. However, the distributed nature of the constituent computing components in a GCPS can lead to cumulative effects of the interactions. These cumulative effects are modeled as global interactions and captured by overlap of ROIns and ROIms of

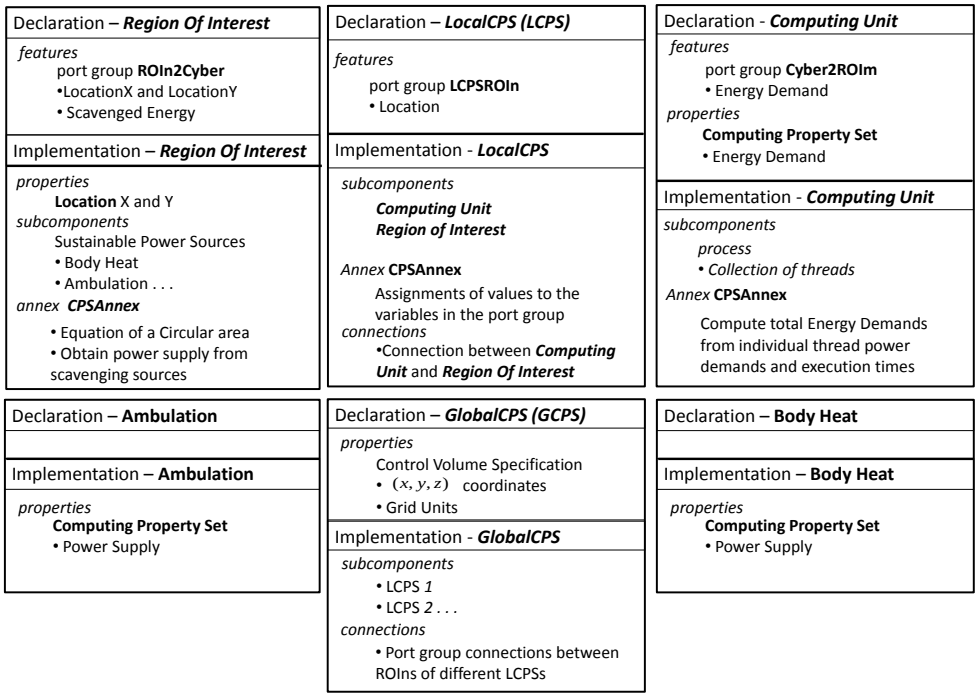


Fig. 9. Sustainability analysis AADL code sample

the constituent LCPS of a GCPS. The variation of the monitored parameter in the overlapped region is an aggregation of the variations in each individual ROIn or ROIm.

We have implemented the aforementioned construct as an extension (annex) to the industry standard Abstract Architecture Description Language (AADL) [3] . The sample code of the model for BSN is shown in the Figure 9. The AADL implementation of the sustainability analysis involves the specification of three entities: 1) the power consumption model of the computing unit (modeled as part of the Computing Unit construct in Figure 9) running Ayushman workload, 2) the power supply models of the scavenging sources (modeled as part of two sample scavenging sources, Ambulation and Body Heat in Figure 9), and 3) the modeling of the energy scavenged from ROIn (modeled as ROIn2Cyber construct in Figure 9). Sensing, data transmission, and PKA protocol steps are modeled as Threads and are characterized by power and energy demand properties. These properties are part of Computing Property Set.

A sustainable power source is modeled as a System (Power Source). Body Heat and Ambulation are different types of power sources shown in the model. These are modeled as implementations of Power Source. Voltage generated by these power sources are modeled by voltage property. Some of these stages consume different amounts of current when radio is turned on or off, these operating characteristics are represented as modes (RadioOn and RadioOff) in a thread. Similar model can be developed for the data center sustainability evaluation where the data center is GCPS,

[3] http://www.aadl.info/

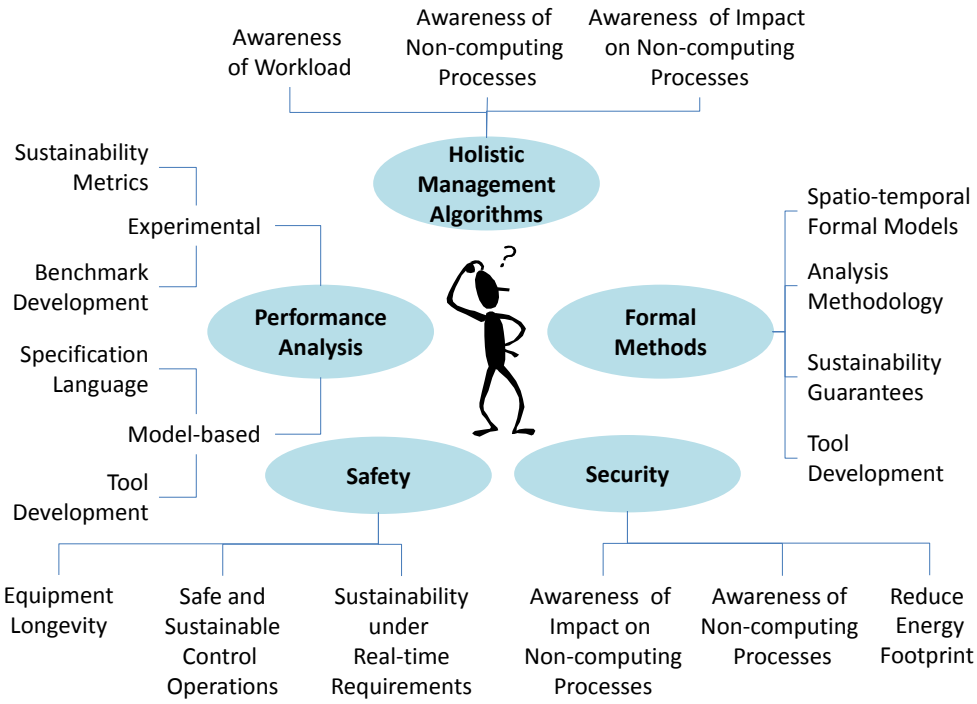


Fig. 10. Open issues and research directions for sustainable CPSs.

each chassis can be modeled as LCPS, inlets of all the chassis can form the ROIms and the output of the CRAC can be the ROIn.

7. Research directions and open problems

There are five major research directions towards design and verification of sustainable CPSs. Figure 7 depicts these research directions along with the open problems in each of these directions. The research directions are described below:

(i) **Holistic Management Algorithms:** Resource management algorithms for CPSs have to be *aware of the non-computing processes* and the *impact of the computing components* on these processes. Such awareness requires predictions based on the impact function F. In most cases, accurate characterization of the impact function requires analysis of complex transient behavior of non-computing processes. For example, transient modeling of the data center as an entire CPS includes transient modeling of cooling equipment and is still work in progress.

In case of BSNs, the transient behavior of the human body is generally non-linear and time variant. For example, in the infusion pump case the drug diffusion rate in the human blood over time follows a non-linear differential equation. Further, the amount of drug diffused is dependent on the drug concentration in the blood at a past time. This is because of the inherent delays in the transport of the drug through the human blood. Formal analysis of

properties of such physiology are not well established. Hence, there is no methodology to design an algorithm that considers the transient behavior of the human body.

Further, Five✷ algorithms need to be developed that can support different types of workload. Handling mixed workload is a challenge, and it requires an appropriate abstraction of the workload to capture different workload types. Another challenge for a Five✷ algorithm is on performing effective power management under online workload arrival. This would require a form of adequate prediction of the workload arrival patterns. In case of data centers, such studies exist for Internet traffic, but studies of high-performance computing (HPC) workload arrival patterns are virtually nonexistent mainly because their submission has been handled as an offline workload. In case of BSNs, since the workload is predominantly periodic, accurate predictions can be performed. As such, radio sleep scheduling and and power management can be employed as described in Section 5.3. Such solutions however is inapplicable for aperiodic workload without proper predictions.

Research in network traffic characterization [50] has shown that the traffic has non-stationary and self similar behavior and the predominant Markovian assumption on the workload arrival does not hold. In such cases, use of fractal analysis techniques common in statistical physics domain are suggested for use. However, such an approach needs to be studied further, and a methodology for developing and analyzing algorithms have to be developed. Thus, developing algorithms for sustainability considering the self similar and non stationary behavior of the input workload is an open problem.

(ii) **Formal Methods:** Formal modeling and analysis of CPSs are required to provide theoretical guarantees on their sustainability. On way to model both the discrete behavior of the computing processes components and the continuous dynamics of the non-computing processes in CPSs is by using traditional linear hybrid automata [51]. The variation of the system parameters can be expressed as linear differential equations. Most of the linear hybrid automata reachability analysis [52] implicitly assume that the first derivative of the continuous variables is constant. Such assumptions are not applicable in general for wearable devices since the differential equations representing the continuous dynamics may have higher orders [5]. Hybrid automata supporting higher order differential equations have been proposed in [53]. However, their reachability analysis is based on numerical simulation instead of an analytical evaluation. The hybrid automata proposed in [53] allow specification of continuous dynamics on two dimensions, but their analysis is also limited to only numerical simulations. As such, the following scientific gap needs to be filled:

- A hybrid automata with the combined capability of specifying *time varying*, *higher order*, *spatio-temporal* continuous dynamics.
- A reachability analysis methodology for such hybrid automata.

One way to fill the gap is to develop Spatio-Temporal Hybrid Automata (STHA). In STHA, discrete states can

not only describe system behavior in time but also in space. The events causing the state transitions can occur over time and while traversing through space. For analysis of such model, discretization can be performed for all but one dimension. In the remaining dimension, existing analysis techniques can be used. However, theoretical bounds need to be provided on any error incurred by the discretization. In this regard, proper tools need to be developed to perform the reachability analysis.

(iii) **Performance Analysis:** There are two principal directions to measure the performance of a CPS in terms of sustainability: (i) experimental, and (ii) model-based. Either simulations or real test-bed can be used for experimentation. A major issue in this regard is the proper metric for sustainability that can capture all the different perspectives described in Section 2. Secondly, proper benchmarks need to be developed that can be used to measure the performance of a CPS in terms of these metrics. The metrics need to be generic and abstract enough to capture all the different types of workloads mentioned previously in this section. In many CPSs, where real experiments can not be performed model-based analysis need to be performed. Various different modeling constructs required to capture the non-computing processes and the impact of the computing components on these processes have been discussed in Section 6. Recent research has further focused on combining formal models with performance models [54]. Novel model specification language (or extensions to existing languages such as AADL) need to be investigated for representation of these constructs. Lastly, model-based analysis tools need to be developed.

(iv) **Safety:** Operations of the CPSs need to be safe, i.e. it should not detrimentally impact the non-computing processes. Such detrimental impact can affect the equipment longevity (as in the case of data centers if safe operating temperatures are not maintained) and can have catastrophic consequences (as in the case of BSNs if proper monitoring or timely drug delivery is not performed). Equipment longevity can further impact the sustainability from the equipment recycling perspective (see Section 2). Safety has an inherent trade-off with energy-sustainability. For example, in data centers over-cooling can reduce energy-sustainability. Similarly, for data centers, it needs to be make sure that timely drug delivery is not compromised to reduce energy requirements. Handling the trade-offs between safety and sustainability is an open problem for CPSs.

(v) **Security:** CPSs pose novel problems and opportunities for information security because of the impact of the computing components on the non-computing processes and the possible awareness of the non-computing processes, respectively. On one hand, the information security needs to be akin to the safety and sustainability. For example, in case of BSN, when there is any medical emergency, the normal access privileges to physiological information needs to be dynamically updated so that any available medical personnel can access the required information. Further, security operation should not have high energy requirement. On the other hand, information from the non-computing processes can aid the security operations. For example, physiological signals can be

used to generate cryptographic keys in BSNs [47]. However, such key generation can involve complex signal processing operation which can impact the energy-sustainability [55]. As such, there is a trade-off between security and sustainability of CPSs. Any security policy has to handle such trade-offs.

8. Conclusions

In this paper, we discussed the various approaches for sustainable computing in CPSs. It is identified that the inherent dependencies among the computing and non-computing processes in a CPS have to be considered for designing sustainable systems and analyzing their sustainability. In this regard, a generic theoretical conceptualization of the power consumption of both computing and non-computing components is provided. This theoretical concept was then demonstrated using two representative CPSs: BSNs and data centers. Further, based on the dependencies of power consumption on workload characteristics, power management strategies, and management of non-computing units, sustainability management algorithms were classified. Five major research directions have been identified: (i) designing holistic management algorithms, (ii) formal methods for verification and theoretical guarantees, (iii) experimental and model-based performance analysis of CPSs, (iv) safety of CPSs, and (v) information security in CPSs. Various open problems in these directions have also been discussed.

References

[1] NewsLink Spring 08, Tackling Todays's Data Center Energy Efficiency Challenges.

[2] J. A. Paradiso et al, "Energy scavenging for mobile and wireless electronics," *Pervasive Computing, IEEE*, vol. 4, no. 1, pp. 18–27, Jan.-March 2005.

[3] Q. Tang et al, "Energy-efficient thermal-aware task scheduling for homogeneous high-performance computing data centers: A cyber-physical approach," *IEEE TPDS*, vol. 19, no. 11, pp. 1458–1472, 2008.

[4] Venkatasubramanian et al., "Ayushman: A Wireless Sensor Network Based Health Monitoring Infrastructure and Testbed," in *Distributed Computing in Sensor Systems*, July 2005, pp. 406–407.

[5] D. Wada and D. Ward, "The hybrid model: a new pharmacokinetic model for computer-controlled infusion pumps," *Biomedical Engineering, IEEE Transactions on*, vol. 41, no. 2, pp. 134 –142, feb. 1994.

[6] S. Roundy, E. S. Leland, J. Baker, E. Carleton, E. Reilly, E. Lai, B. Otis, J. M. Rabaey, V. Sundararajan, and P. K. Wright, "Improving power output for vibration-based energy scavengers," *IEEE Pervasive Computing*, vol. 4, no. 1, pp. 28–36, 2005.

[7] P. Rong and M. Pedram, "Power-aware scheduling and dynamic voltage setting for tasks running on a hard real-time system," Jan. 2006, p. 6 pp.

[8] L. Huaming et al, "An ultra-low-power medium access control protocol for body sensor network," 2005, pp. 2451 –2454.

[9] S. Ullah et al, "MAC hurdles in body sensor networks," in *ICACT'09: Proceedings of the 11th international conference on Advanced Communication Technology*. Piscataway, NJ, USA: IEEE Press, 2009, pp. 1151–1155.

[10] R. Kumar et al, "Computation hierarchy for in-network processing," in *WSNA '03: Proceedings of the 2nd ACM international conference on Wireless sensor networks and applications*, New York, NY, USA, 2003, pp. 68–77.

[11] T. Mukherjee et al, "Spatio-temporal thermal-aware job scheduling to minimize energy consumption in virtualized heterogeneous data centers," *Computer Networks*, June 2009.

[12] U. E. P. Agency, "Report to congress on server and data center energy efficiency public law 109-431," ENERGY STAR Program, 2007.

[13] A. Banerjee, S. Kandula, T. Mukherjee, and S. K. S. Gupta, "BAND-AiDe: A tool for cyber-physical oriented analysis and design of body area networks and devices," *ACM Transactions on Embedded Computing Systems (TECS), Special issue on Wireless Health Systems, (minor revision submitted for review)*, 2010.

[14] J. d. Geus, D. Posthuma, N. Kupper, M. Berg, G. Willemsen, A. Beem, P. Slagboom, and D. Boomsma, "A whole-genome scan for 24-hour respiration rate: a major locus at 10q26 influences respiration during sleep," Tech. Rep., 2005.

[15] H. Ibrahim, A. Ilinca, and J. Perron, "Energy storage systems–characteristics and comparisons," *Renewable and Sustainable Energy Reviews*, vol. 12, no. 5, pp. 1221 – 1250, 2008.

[16] H. Lund and G. Salgi, "The role of compressed air energy storage (caes) in future sustainable energy systems," *Energy Conversion and Management*, vol. 50, no. 5, pp. 1172 – 1179, 2009.
[17] "Review on thermal energy storage with phase change materials and applications," vol. 13, no. 2, pp. 318–345, 2009.
[18] M. R. Gary and D. S. Johnson, *A Guide to the Theory of NP-Completeness*. Freeman, 1979.
[19] D. Tsafrir, Y. Etsion, and D. G. Feitelson, "Backfilling using system-generated predictions rather than user runtime estimates," *IEEE Transactions on Parallel and Distributed Systems (TPDS)*, vol. 18, no. 6, pp. 789–803, Jun. 2007.
[20] J. L. W. Jennifer Burge, Partha Ranganathan, "Cost-aware scheduling for heterogeneous enterprise machines," in *Workshop on Green Computing (GreenCom), Proceedings of the IEEE Cluster conference*, Sep. 2007.
[21] J. Moore, J. Chase, and P. Ranganathan, "Weatherman: Automated, online, and predictive thermal mapping and management for data centers," in *3rd IEEE Int'l Conf. Autonomic Computing*, June 2006.
[22] J. Moore et al, "Making scheduling "cool": Temperature-aware resource assignment in data centers," in *2005 Usenix Annual Technical Conference*, April.
[23] Z. Abbasi, G. Varsamopoulos, and S. K. S. Gupta., "Thermal aware server provisioning and workload distribution for internet data centers," in *ACM International Symposium on High Performance Distributed Computing (HPDC10)*, Jun. 2010.
[24] A. Banerjee, T. Mukherjee, G. Varsamopoulos, and S. K. S. Gupta, "Cooling-aware and thermal-aware workload placement for green hpc data centers," in *International Conference on Green Computing Conference (IGCC2010)*, Aug. 2010.
[25] H. Taliver et al, "Mercury and Freon: temperature emulation and management for server systems," in *ASPLOS-XII: Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*. New York, NY, USA: ACM Press, 2006, pp. 106–116.
[26] L. Ramos and R. Bianchini, "C-oracle: Predictive thermal management for data centers," in *IEEE 14th International Symposium on High Performance Computer Architecture (HPCA2008)*., Feb. 2008, pp. 111–122.
[27] R. Sharma, T. Christian, M. Arlitt, C. Bash, and C. Patel, "Design of farm waste-driven supply side infrastructure for data centers," 2010.
[28] "Hp designs sustainable datacenter fueled by cow manure," http://www.smartplanet.com/business/blog/smart-takes/hp-designs-sustainable-datacenter-fueled-by-cow-manure/7189/.
[29] P. Ranganathan et al, "Ensemble-level power management for dense blade servers," in *(ISCA'06)*, Boston, MA, May 2006, pp. 66–77.
[30] R. K. Sharma, C. E. Bash, C. D. Patel, R. J. Friedrich, and J. S. Chase, "Balance of power: Dynamic thermal management for internet data centers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 42–49, 2005. [Online]. Available: http://doi.ieeecomputersociety.org/10.1109/MIC.2005.10
[31] L. K. Au, W. H. Wu, M. A. Batalin, D. H. McIntire, and W. J. Kaiser, "Microleap: Energy-aware wireless sensor platform for biomedical sensing applications," 2007, pp. 158–162.
[32] P. H. Chou and C. Park, "Energy-efficient platform designs for real-world wireless sensing applications," in *ICCAD '05: Proceedings of the 2005 IEEE/ACM International conference on Computer-aided design*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 913–920.
[33] "8 most "eco-friendly" nokia mobile phones," http://www.environmentteam.com/2010/04/07/8-most-eco-friendly-nokia-mobile-phones/.
[34] L. Yang, R. Vyas, A. Rida, J. Pan, and M. Tentzeris, "Wearable rfid-enabled sensor nodes for biomedical applications," may. 2008, pp. 2156 –2159.
[35] "Eco gadgets: Recyclable paper laptop for sustainable computing," http://www.ecofriend.org/entry/eco-gadgets-recyclable-paper-laptop-for-sustainable-computing/.
[36] H. Ghasemzadeh and R. Jafari, "Data aggregation in body sensor networks: A power optimization technique for collaborative signal processing," jun. 2010, pp. 1 –9.
[37] S. Nabar, J. Walling, and R. Poovendran, "Minimizing energy consumption in body sensor networks via convex optimization," in *BSN '10: Proceedings of the 2010 International Conference on Body Sensor Networks*. Washington, DC, USA: IEEE Computer Society, 2010, pp. 62–67.
[38] H. Ghasemzadeh, N. Jain, M. Sgroi, and R. Jafari, "Communication minimization for in-network processing in body sensor networks: A buffer assignment technique," apr. 2009, pp. 358 –363.
[39] L. Wang, G. von Laszewski, J. Dayal, X. He, A. Younge, and T. Furlani, "Towards thermal aware workload scheduling in a data center," dec. 2009, pp. 116 –122.
[40] Y. Lee and A. Zomaya, "Energy efficient utilization of resources in cloud computing systems," *The Journal of Supercomputing*, pp. 1–13, 2010, 10.1007/s11227-010-0421-3. [Online]. Available: http://dx.doi.org/10.1007/s11227-010-0421-3
[41] J. Moore, J. Chase, P. Ranganathan, and R. Sharma, "Making scheduling "cool": Temperature-aware resource assignment in data centers," in *2005 Usenix Annual Technical Conference*, April 2005.
[42] J. Sherwani, N. Ali, N. Lotia, Z. Hayat, and R. Buyya, "Libra: a computational economy-based job scheduling system for clusters," *Software: Practice and Experience*, vol. 34, no. 6, pp. 573–590, 2004.
[43] R. Ayoub, S. Sharifi, and T. Simunic Rosing, "Gentlecool: Cooling aware proactive workload scheduling in multi-machine systems," mar. 2010, pp. 295 –298.
[44] V. Subramanian, M. Gilberti, and A. Doboli, "Online adaptation policy design for grid sensor networks with reconfigurable embedded nodes." in *DATE*. IEEE, 2009, pp. 1273–1278.
[45] G. Varsamopoulos, Z. Abbasi, and S. K. S. Gupta, "Trends and effects of energy proportionality on server provisioning in data centers," in *International Conference on High performance Computing Conference (HiPC2010)*, Dec. 2010.
[46] T. Qinghui et al, "Thermal-aware task scheduling for data centers through minimizing peak inlet temperature," *IEEE Transactions on Parallel and Distributed Systems, Special Issue on Power-Aware Parallel and Distributed Systems (TPDS/PAPADS)*, vol. to appear, 2008.
[47] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, "Plethysmogram-based secure inter-sensor communication in body area networks," *Military Communications Conference, 2008. MILCOM 2008. IEEE*, pp. 1–7, Nov. 2008.
[48] H. H. Pennes, "Analysis of tissue and arterial blood temperature in the resting human forearm," in *Journal of Applied Physiology*, vol. 1.1, 1948, pp. 93–122.

[49] H. Ghasemzadeh et al, “Sport training using body sensor networks: a statistical approach to measure wrist rotation for golf swing,” in *Proc. of the Intl. Conf. on Body Area Networks*. ICST, 2009, pp. 1–8.
[50] R. Marculescu and P. Bogdan, “The chip is the network: Toward a science of network-on-chip design,” 2007.
[51] T. A. Henzinger, “The theory of hybrid automata.” IEEE Computer Society Press, 1996, pp. 278–292.
[52] G. Lafferriere, G. J. Pappas, and S. Yovine, “Reachability computation for linear hybrid systems,” in *In Proceedings of the 14th IFAC World Congress, volume E*. Elsevier Science Ltd, 1998, pp. 7–12.
[53] P. Ye, E. Entcheva, S. A. Smolka, M. R. True, and R. Grosu, “Hybrid automata as a unifying framework for modeling cardiac cells,” in *In Proc. of EMBS06, the 28th IEEE International Conference of the Engineering in Medicine and Biology Society*. IEEE Press, 2006, pp. 4151–4154.
[54] C. Baier, B. R. Haverkort, H. Hermanns, and J.-P. Katoen, “Performance evaluation and model checking join forces,” *Commun. ACM*, vol. 53, no. 9, pp. 76–85, 2010.
[55] K. K. Venkatasubramanian, A. Banerjee, and S. K. S. Gupta, “Green and sustainable cyber-physical security solutions for body area networks,” in *BSN ’09: Proceedings of the 2009 Sixth International Workshop on Wearable and Implantable Body Sensor Networks*. Washington, DC, USA: IEEE Computer Society, 2009, pp. 240–245.