ISSN:0975-9646

Comparative Study of Abstraction in Cyber Physical System

Pooja Sobhrajan
Computer Engineering Department
DYPIET Pimpri, Pune

Swati Y Nikam
Information Technology Department
DYPIET Pimpri, Pune

Abstract -- **Recent trends in computing include increases in both distribution and wireless connectivity, leading to highly dynamic, complex environments on top of applications that must be built. The way of designing and ensuring the correctness of applications in these environments is becoming more complex. The goal of the research in wireless systems is to provide abstractions of complex low-level concepts to application programmers, easing the design and implementation of applications. Similar to other abstractions, it creates logical collections of sensing devices. Earlier abstractions were focused on wireless sensor networks (WSNs) and did not address aspects of Cyber Physical System (CPS). A new class of applications for wireless sensor networks requires similar complexity encapsulation. However, sensor networks have some unique characteristics, including dynamic availability of data sources and application quality of service requirements that are not common to other types of applications. These unique features, combined with the inherent distribution of sensors, limited energy and bandwidth resources, dictate the need for network functionality and the individual sensors to be controlled to best serve the application requirements. Programming abstraction is to make good application design and to reduce algorithmic complexity, inter-component coupling and total lines of code. The abstraction is used to implement several complex applications.**

Keywords: sensor network, cyber physical system, actuator, communications, computation, programming, software.

I. INTRODUCTION

Using the Internet we can interact with people and get useful information from wide world in a very short time. Thus, the Internet transformed how humans interact with one another, CPS will transform how we interact with physical world around us. However, there is a gap between the cyber world in which information is transmitted and modified and the physical world in which we live. The last two decades have brought a digital revolution that has transformed the industry. This change is not a choice, but it is determined by fundamental economic and technological long term trends, that have created an environment which allows and requires a wide and varied range of new capabilities.

Computer-enabled systems are becoming ubiquitous, complex and often play a critical role. Architectures are moving from isolated embedded control systems to open interactive systems that sense and affect their environment, with essential integration of the cyber and the physical, hence the term “cyber-physical” system. As an example, consider modern cars. They are not only concerned with controlling the operation of engine, brakes, locks, but also with helping the driver be aware of surroundings (other vehicles, people, obstacles), entertainment, and with monitoring system function and tracking maintenance. In addition, there is an explosion of small applications running on mobile devices. A future in which these applications combine and collaborate to provide powerful new functionality, not only in the realm of entertainment and social networking, but also by harnessing the underlying communication and people power for new kinds of cyber crowd sourcing tasks.

CPS integrate the dynamics of the physical processes with those of the software and communication, providing abstractions and modelling, design and analysis techniques for the integrated whole. The dynamics among computers, networking and physical systems interact in ways that require fundamentally new design technologies. The technology depends on the multi-disciplines such as embedded systems, computers, communications, etc. and the software is embedded in devices whose mission is not computation alone, e.g. cars, medical devices, scientific instruments, and intelligent transportation systems [3].

The recently appeared CPS term will have to enable the development of a modern vision for the social services that transcend time and space to dimensions never seen before. A complex CPSs definition was given by Shankar Sastry from University of California, Berkeley in 2008: "A CPS integrates computing, communication and storage capabilities with monitoring and/or control of entities in the physical world, and must do so dependably, safety, securely, efficiently and real-time". CPSs are not: the traditional embedded systems or the real-time systems, the today's sensor networks and only desktop applications, but they have certain characteristics that define them, and presented below:

(i) Cyber capabilities in every physical component; (ii) Networked at multiple and extreme scale;
(iii) Dynamically reconfiguring/reorganizing;
(iv) High degrees of automation, the control loops must close;
(v) Operation must be dependable and certified in some cases;
(vi) Cyber and physical components are integrated for learning.

Following are the features of cyber physical system

i. *Cross-domain sensor sources and data flows:*
Multiple types of sensors will be adopted at the same time in intelligent CPS applications. Moreover, these cross-domain sensing data will be exchanged over heterogeneous networks.

ii. *Embedded and mobile sensing:*
Sensors are no longer static and may have high-degree mobility through carriers such as smart phones and

vehicles. This implies that sensing coverage would vary over time, exhibiting more uncertainty.

iii. *User contribution and cooperation:*
Participatory sensing would be common in CPS. A give-and-take like data contribution model is needed to encourage sharing. As sharing happens more frequently, the privacy issue becomes a concern too.

iv. *Elastic loads requiring cloud-supported storage and computing capability:*
With the maturity of cloud computing, the pay-as-you-go concept is likely to be adopted in CPS to serve storage, computing, and communication needs. This allows CPS developers to focus on their own work. Similarly, users can choose the part of CPS applications that they really want, rather than over-load the system.

v. *Accumulated intelligence and knowledge via learning and data mining technologies:*
Data in CPS may have high dynamics and uncertainty. Retrieving useful knowledge will rely on learning and data mining technologies. It is essential to understand the temporal and spatial correlations of these sensing data. Further, feedback from users and actuators may help us to build up unknown knowledge.

vi. *Rich interactions among many objects and things through the Internet:*
A lot of sensor–sensor, sensor–actuator, actuator–actuator, actuator–user, user–user, user–object, object–object, object–thing, thing–thing, and thing–user interactions may occur in CPS applications. Such rich interactions demand flexible communication channels to make easy to our applications.

II. Current Research Challenges In CPS

Currently, the research is divided into isolated sub-discipline, such as: communications and networking, systems theory, mathematics, software engineering, computer science, sensors. Thus, digital systems are designed using a variety of modeling tools and formalisms. Each representation brings out some features and doesn't take into account the other in order to make controllable analysis. Usually, formalism represents either the cybernetic processes or physical but not as necessary to achieve CPSs. The following paragraphs present the main directions of research needed in CPSs domain that is still in its early-stage:

A. *Abstraction and Architectures* - Innovative approaches to abstractions (formalisms) and architectures to enable control, communication and computing integration for the rapid design and implementation of CPSs have to be developed. It allow the integration and interoperability of heterogeneous systems that composed the CPSs in a modular, efficient and robust manner.

B. *Distributed Computations and Networked Control* - refers to new frameworks, algorithms, methods, and tools related to time-and event-driven computing, software, variable time delays, failures, reconfiguration, and distributed decision support systems to satisfy the high reliability and security requirements for heterogeneous cooperating components that interact through a physical environment.

C. *Verification and Validation* - Hardware and software components as well as the systems have to overcome their actual stage and to achieved a high degree of dependability, re-configurability, and when is required to be certified. New models, algorithms, methods, and tools to verify and validate software components and also entire system from early design stage represent the research directions addressed to the scientific community.

Also, the scientific challenges in the CPSs field were highlighted are the following:

(1) The realignment of the abstraction layers in design flows -the abstractions must include physical concepts such as time and energy. These changes related to the layers of abstraction will allow the synthesis of computations with physical properties and physical system dynamics that are robust against implementation uncertainties.

(2) The development of the semantic foundations for composing heterogeneous models and modeling languages that describe different physics and their associate logics.

(3) The development of a new understanding of compositionality in heterogeneous systems that allows the creation of large, networked systems that satisfy essential physical properties and deliver the required functionality in a reliable way.

(4) The development of a technology for achieving the predictability in partially compositional properties.

(5) The development of a model based, precise and predictable technology foundation for system integration.

(6) The development of a new infrastructure for agile design automation of CPSs.

(7) The development of new open architectures for CPSs that will allow the building of the national-scale and global-scale capabilities.

(8) The development of architectures and tools for reliable CPSs from unreliable components and resilient.

III. Need For Abstraction In CPS

The bandwidth and energy limitations of sensor nodes require that in-network processing be performed to reduce the amount of data that must be transferred out of the network. Application designers are faced with the problem of decomposing an initially straightforward data-collection task into a parallel program with local communication among sensor nodes. Currently, sensor application designers spend a great deal of effort building up low-level machinery for routing, data collection, and energy management. To move away from this state of affairs by providing higher level programming interfaces that abstract these details, yet provide enough flexibility to implement efficient algorithms. It provides a unified interface for message passing across a large family of parallel machines. It hides the details of the communication hardware and provides efficient implementations of common collective operations, such as broadcast and reduction. It has been extremely successful in the parallel processing community as it is high-level enough to shield programmers from most of the details of the underlying machine, yet low level enough to permit extensive application-specific optimizations. It provides communication interfaces that serve a similar role for sensor networks.

IV. ABSTRACTION AND ITS TYPES

Abstraction is a powerful technique for the design and implementation of complex systems. A model developed at a higher level of abstraction allows one to tackle complexity by initially hiding the details and elaborating them later. A higher level of abstraction has a positive effect on the simulation speed and ease of development of the model, but could affect the accuracy of the model developed.

Types of abstraction are as follows:-

A. Neighborhood abstraction -- A neighborhood programming abstraction for sensor networks, wherein a node can identify a subset of nodes around it by a variety of criteria and share state with those nodes. This abstraction allows developers to design distributed algorithms in terms of the neighbourhood abstraction itself, instead of decomposing them into component parts. It reduces algorithmic complexity, inter-component coupling, and total lines of code.

B. Grouping Abstraction –Grouping mechanisms tend to focus on forming groups that are more general and typically contain membership criteria beyond just neighborhood constraints. It allows operations to be performed on the chosen subset through code migration.

C. Macro programming abstraction – Macro programming environment for a network of sensors automates the process of decomposing global programs into complex local behaviours. It employs in which a user writes a single program is then distributed across the sensor network.

V. COMPARATIVE STUDY OF VARIOUS ABSTRACTIONS

Currently there is very little published literature that compares abstraction making our work unique. Some of them are as follows:-

Abstraction Type	Advantages	Disadvantages	Applications
Neighbourhood Abstraction **(Hood: A Neighborhood Abstraction for Sensor Networks)**	It defines a relationship between several fundamental concepts of neighborhood.	a) Cannot group nodes that belong to diff. networks. b)Actuators are not supported.	Object Tracking Application
Neighbourhood Abstraction (**Logical Neighborhood Programming Abstraction for Wireless Sensor Networks)**	a) Achieves a longer system lifetime. b) Better Resource utilization	a)Does not support actuators b) Cannot write new application without reprogramming them.	Home Automation Environment monitoring
Neighbourhood Abstraction **(Programming Sensor Networks Using Abstract Regions)**	Increase the lifetime of the system	a) Cannot group sensors from different networks, actuators.	Object Tracking, Contour Finding
Grouping Abstraction **Bundle: A Group-based Programming Abstraction for CPS**	Provide fine grained access right control and conflict resolution mechanism.	a) Does not capable of in-network aggregation and processing. b)Strong dependency	a)Environmental Monitoring b)Tracking Application c)Control Automation
Grouping Abstraction **Spatial Views: Space-Aware Programming for Networks of Embedded Systems**	Time and other resource constraints allow the programmer to manage the inherent volatility of underlying network.	a) Does not support actuators. b)Heterogeneity or multiple networks are not supported.	Location Tracking
Macro programming Abstraction **Macro Lab: A Vector-based Macro programming Framework for Cyber-Physical Systems**	The user is free to write Deployment-independent programs that is simple, robust and easy to understand. It can reduce overall resource consumption.	Does not support actuators.	Object Tracking Application

VI. Conclusion

As sensor networks become more common, better tools are needed to aid the development of applications for this challenging domain. Programmers should be shielded from the details of low-level radio communication, addressing, and sharing of data within the sensor network. At the same time, the communication abstraction should yield control over resource usage and make it possible for applications to balance the tradeoff between energy, bandwidth consumption and the accuracy of collective operations. The essential idea is to capture communication patterns, locality and resource tradeoffs in a high-level language that compiles down to the detailed behavior of individual nodes. Shielding programmers from the details of message routing, in-network aggregation and achieving a given fidelity under a fixed resource budget should greatly simplify application development for this new domain.

References

[1] Jong-Seok Choi, Tim McCarthy, ManeeshYadav, Minyoung Kim, Carolyn Talcott,Eric-Gressier-Soudan, "Application Patterns for Cyber Physical Systems", in IEEE, 2013, pp. 52-59.

[2] VasanthRajamani, SanemKabadayı, and Christine Julien, "An Interrelational Grouping Abstraction for Heterogeneous Sensors" ACM Transactions on Sensor Networks, Vol. V, No. N, May 2008, pp.1-29.

[3] Jianhua Shi Jiafu Wan* Hehua Yan, HuiSuo, "A Survey of Cyber Physical System", In Proceedings of the International Conference on Wireless Communications and Signal Processing, China, 9th November 2011.

[4] E. Cheong, J. Liebman, J. Liu, and F. Zhao, "Tinygals: A programming model for event-driven embedded systems," in Proceedings ACMSymposium Application and computing, 2003, pp. 698–704.

[5] C. Borcea, C. Intanagonwiwat, P. Kang, U. Kremer, and L. Iftode, "Spatial programming using smart messages: Design and implementation," in Proceedings 24th International Conference Distributed Computing System, 2004, pp.690–699.

[6] Y. Ni, U. Kremer, and L. Iftode, "Spatial views: Space-aware programming for networks of embedded systems," in Proc. 16th International Workshop Languages Compilers Parallel Computing, 2003, pp. 258–272.

[7] K. Römerand, C. Frank, P. J. Marrón, and C. Becker, "Generic role assignment for wireless sensor networks," in Proceedings 11th Workshop ACM SIGOPS Eur. Workshop, 2004, pp. 7–12.