

High Confidence Software for Cyber-Physical Systems

Sherif Abdelwahed
Mississippi State University
Starkville, MS 39762

Nagarajan Kandasamy
Drexel University
Philadelphia, PA 19104

Aniruddha Gokhale
Vanderbilt University
Nashville, TN 37235

sherif@ece.msstate.edu

kandasamy@ece.drexel.edu

a.gokhale@vanderbilt.edu

ABSTRACT

Many new and planned cyber-physical systems (CPSs) are realized as distributed real-time and embedded (DRE) systems. Examples of DRE CPSs we are interested in include data computing centers and automated warehouse management systems.

Categories and Subject Descriptors

D.2.11 [Software Engineering]: Software Architectures – domain-specific architectures.

General Terms

Algorithms, Management, Design, Reliability, Verification.

Keywords

Cyber-physical systems.

1. Emergent Traits of DRE CPSs

Next generation DRE CPSs illustrate the following emergent characteristics:

- *Complexity and scale.* As an increasing number of services are moved online, the required computing infrastructure keeps growing in complexity and scale. Effectively managing the performance of these large-scale computing systems subsequently becomes more complex, requiring increasing numbers of skilled personnel to configure, operate, and optimize them.
- *Dynamic and uncertain operating environment.* These systems typically execute in a dynamic and uncertain operating environment caused by multiple factors such as time-varying user workload, hardware and software resource failures, incomplete knowledge of the system operating state, and other vulnerabilities, such as security violations or denial

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

ASE Workshop on Automating Service Quality, November 2007, Atlanta, Georgia, USA © 2007 ACM ISBN: 978-1-59593-878-7 /07/11...\$5.00

of service attacks.

- *Use of on-demand computing models.* This is an emerging resource provisioning model to efficiently host applications, where computing resources e.g., servers, memory, and storage for data centers or warehouse resources e.g., forklifts and belts for warehouses, are dynamically made available to these applications as needed, and not statically allocated based simply on peak demand
- High-confidence is an essential quality of service (QoS) property of these emergent DRE CPSs that must be considered over the entire lifecycle of these systems, including design, development, deployment, and operation. Meeting these demands using today's stovepiped technologies is tedious, error-prone, and costly to develop, optimize, validate, deploy, and maintain.

2. R&D Needs for DRE CPSs

Realizing high-confidence software for DRE CPSs requires meeting the following criteria.

- *Trustworthiness* – DRE CPSs must be trustworthy, which includes the system's ability to meet performance objectives, and be resilient to failures and security attacks.
- *Autonomy* – DRE CPSs must be autonomous i.e., self healing, self configuring and self optimizing while maintaining good resource utilization.
- *Analyzability* – The algorithms and technologies used to develop DRE CPSs must be amenable to analyses and verification for different properties, such as timeliness guarantees, fault tolerance, stability and correctness.

3. Design and Architecture

The solution needs of high confidence DRE CPSs can be met by developing novel techniques described below and synergistically integrating them as shown in Figure 1.

- **Model-driven System Execution Modeling** – which requires developing modeling abstractions to obtain high-fidelity system models of DRE CPSs that describe its structure and expected (i.e., correct) behavior at various levels of abstraction [1] e.g., platform independent and platform-specific. These models will be executed

concurrently with actual system operation and the results compared. A divergence between the system behavior and that of the corresponding model may be treated as a symptom of an anomaly that must be diagnosed and fixed [2].

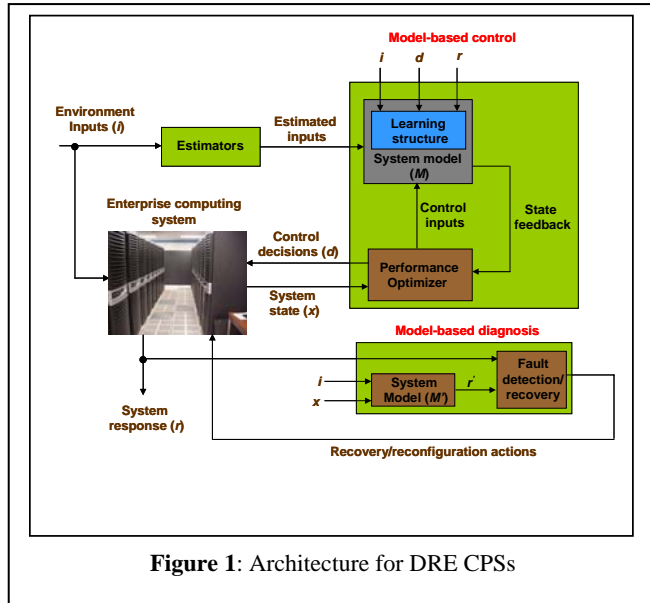


Figure 1: Architecture for DRE CPSs

- Model-based Diagnosis** – which requires developing distributed, model-based diagnosis techniques that use the observed divergence between the actual system and its models to isolate and characterize possible attacks or errors as well as the set of possibly corrupted resources [2]. These algorithms will have to detect, isolate, and estimate the state of corrupted hardware/software components using concepts from continuous and discrete-event diagnosis, and consistency-based causality analysis.
- Online Control** – which requires handling complex optimization problems under uncertainty within a *receding horizon control* framework. Based on our earlier work [3], the notions of uncertainty and risk will be incorporated within these frameworks to cope with unexpected changes to performance goals, dynamic workload, and hardware and software failures. Online parameter tuning and model learning techniques will need to be integrated within the control framework to both improve the quality of partially specified system models and adapt to changes in the system model itself over time (e.g., addition or removal of system components, replacement of components).
- Fault-adaptive Control** — which requires developing technologies that integrate health management and online control to achieve more robust operation of DRE systems. Health management incorporates fault diagnostics for

detecting, isolating, and identifying faults in components, and assessing their effects on the system performance, while the control technology aims to maintain high performance under uncertain operating conditions. Information produced by the diagnosis process is used to modify the controller such that the system meets its desired performance metrics or degrades gracefully under serious component failures and malfunctions. We call this integrated control and diagnosis approach fault-adaptive control to differentiate it from related approaches such as fault-tolerant or robust control. Fault-adaptive control technology aims to improve both the operating efficiency and reliability of the computing system by: (1) providing modeling methods to implement more effective online fault detection and isolation, especially for distributed systems; (2) exploiting the integration of health management and adaptive control; and (3) supplying a tool suite to implement fault adaptive control as an integral part of the systems engineering process

- Trustworthy Middleware Infrastructure** – which requires developing QoS-enabled, fault tolerant and self-healing middleware that (a) provides real-time system monitoring capabilities that collect the desired parameters of interest, and (b) can dynamically integrate the artifacts synthesized by the model-based diagnostics and control algorithms, and deploy these software artifacts optimally to assure continuous operation of DRE CPSs

4. Solution Approach

Realizing the vision of high confidence DRE CPSs need not start from scratch. A substantial set of early capabilities exist [4] that can be leveraged to meet the emergent demands of DRE CPSs. For example, as shown in Figure 2, existing capabilities developed at Vanderbilt, such as the PICML modeling language, can be used to model the assembly and deployment of DRE CPSs. Generative tools associated with PICML can synthesize the configuration and deployment artifacts necessary to automate the deployment of these systems via the DAnCE middleware. Distribution component middleware, such as CIAO, already

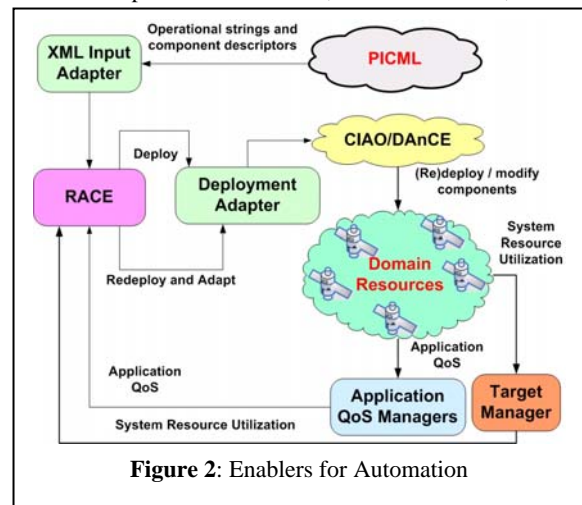


Figure 2: Enablers for Automation

provide the capabilities to support real-time component-based applications. The resource and allocation engine (RACE) is a framework that enables the plugging in of different control algorithms that promote runtime adaptation.

Additional capabilities, including analyses, model-based diagnostics, robust control algorithms, system behavioral modeling, and trustworthy and self-healing middleware must be developed to enhance these existing frameworks and meet the objectives of high confidence software for DRE CPSs.

5. REFERENCES

- [1] G. Karsai, J. Sztipanovits, A. Ledeczi, and T. Bapty, "Model-Integrated Development of Embedded Software," *Proceedings of the IEEE*, vol 91, no. 1, January 2003.
- [2] H. Shrobe, "Model-based diagnosis for information survivability," *Self-Adaptive Software*, Editors: R. Laddaga, P. Robertson, and H. Shrobe, Springer-Verlag, 2001.
- [3] S. Abdelwahed, N. Kandasamy, and S. Neema, "Online control for self-management in computing systems," *Proc. 10th IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS04)*, pp. 368-375, May 2004.
- [4] A. Gokhale, D. C. Schmidt, B. Natarajan, J. Gray, and N. Wang, Model Driven Middleware. In Q. Mahmoud, editor, *Middleware for Communications*, pages 163–187. Wiley and Sons, New York, 2004.