Proceedings of the International MultiConference of Engineers and Computer Scientists 2014 Vol II,
IMECS 2014, March 12 - 14, 2014, Hong Kong

Unified Dynamic Resource Supply Model to Support Cyber Physical System

Yuji Dong, Kaiyu Wan and Yong Yue

Abstract—**Cyber Physical System (CPS) is a complex research field with plenty of challenges. The Dynamic Resource Supply Model(DRSM) we proposed before can describe the resources and support manipulation of resources in CPS to handle all resource allocations. However after extensively investigating past research results about resource specifications and allocations, we realize the DRSM is not sufficient for resource usage due to the insufficiency of previous Resource Operating Decision Machine (RODM). In particular, two modules: *Context Manager* and *Decision Engine* we proposed have no detailed mechanism on how to make decisions and manage context. Therefore in this paper, we first give formal definition of resource, and propose two key points in resource model from our definition. Next, DRSM is compared with various resource models and allocation algorithms in different areas. Then, we divide resource management approaches into two fundamental parts for discussion, that is, resource specification and resource usage specification. Finally the improved DRSM has been presented.**

Index Terms—**Resource Allocation, Resource Model, Cluster Computing, Cyber Physical Systems**

I. Introduction

Cyber Physical Systems are integrating physics-based and digital world models, which will contribute immeasurably to a broad range of industries including energy, infrastructure, health care, manufacturing and military etc [1], [2], [3]. Although many researchers have been working on the various aspects of CPS, there are many challenges remaining due to the lack of science and technology foundations covering design methodology, security, and system integration etc. Because of the scalability and complexity of systems, the design methodologies including specification, modelling, analysis and verification of such systems are highly required. In our previous work, we built a basic Dynamic Resource Supply Model to support resource generation and allocation mechanism in CPS. In the methodology we proposed, instead of modelling the whole Cyber Physical System, we consider to simplify the problem with handling all resources generation, management and allocation first. The Dynamic Resource Supply Model can act as a separate layer belonging to entire CPS. Therefore specifying, modelling, analysing and verifying complex Cyber Physical System can be simplified.

In traditional computer science, the topics about resources allocation are focused on CPU, memory or communication bandwidth allocation to improve the system performance. In CPS, resource is not restricted to traditional computation and memory resources. In such a complex system consisting of humans, software systems, hardware components and other physical processes, resource, in a generic sense, could be basic entities like instruction cycles in CPU, or abstracted entities from many basic entities like servers in cluster, or even some other particular resources in special domains. The resource concept is becoming obscure and complicated. A complex CPS may rely on many different resources and some of them have complex dependencies. The resources usage also become fuzzy and hidden in the complex structures.

This paper is to investigate past researches on resources in different fields; give a definition of resources; and propose the methodology dealing resources in complex Cyber Physical System to improve our Dynamic Resource Supply Model. Section II is the literature review of various resource management and allocation strategies in different fields like Operating System, Virtual Technique, Cloud Computing and Resource Description Framework. In addition, the resource model - ”Dynamic Resource Supply Model (DRSM)” [32] proposed by us is briefly introduced. In Section III, we give a formal definition of resource and analyse two important parts in resource model, i.e., Resource Specification and Resource Usage Specification. In addition, we discuss and analyze the requirements for dynamic resource supply model. As a result, the improved DRSM with more details and mechanisms is presented in Section VI. We conclude this paper with brief summary and future work in Section V.

II. Background

In the computer science field, the research about resources can be generally grouped into two categories: computational resources and information entities. The computational resources include CPU, memory, I/O, etc., and information entities are mostly used on network communication area. Below we discuss these two categories in detail.

A. Computational Resources

Resource investigations on computational resources are focused in operating systems, virtual techniques, grid computing, cluster computing and cloud computing research areas.

- Resource Management is one of the core fundamental aspects in *Operating System*, which could date back to 50 years ago [25]. Recently, with the development of distributed systems and real-time systems, people investigate the resources in operating systems from different point of view. Microsoft Research developed an object-based real-time kernel and programming environment, i.e., Rialto Operating System [7], to support modular distributed real-time resource management. Exokernel, which is an operating system architecture for

Manuscript received January 19, 2014.
Yuji Dong, Kaiyu Wan and Yong Yue are with the Department of Computer Science and Software Engineering, Xi’an Jiaotong-Liverpool University, China. Email: yuji.dong07@xjtlu.edu.cn kaiyu.wan@xjtlu.edu.cn yong.yue@xjtlu.edu.cn
This research is supported by Research Grants from National Natural Science Foundation of China (Project Number 61103029), and XJTLU PGR Scholarship PGRS-12-03-05

application-level resource management developed by M.I.T., changed the traditional operating system abstraction to best achieve performance and functionality goals of particular applications [20]. In the Exokernal architecture, all hardware resources are exported through a low-level interface to untrusted library operating system, which uses this interface to execute system objects and policies. In the ECOSystem, which manages energy as first class operating system resource, Currentcy Model is proposed as unifying abstraction to integrate power management into the hardware abstraction and resource allocation.[21]

- *Virtual technique* like JVM and VMware is a very important technology recently. The researches on virtual machine systems are mainly focused on the resource policies. In [22], several different managing memory mechanisms and policies in VMware ESX Server are introduced. The RM project [26] starts a pioneering work about the need for resource management in Java.
- In *Grid Computing*, an information service architecture is proposed to discover, characterize, and monitor resources, services and computations in [17]. From a different point of view, in [12] the distributed resource management is supported with an extended Peer-2-Peer computing framework which uses service-oriented architecture to do grid computing. In [11], the research focuses on resource selection for Grid Applications to get the suitable resources from heterogeneous environment.
- *Cluster Computing* is a popular topic since google published their Google File System and after Hadoop was developed. Mesos is a very famous resource sharing and management platform to host multiple diverse cluster computing frameworks, such as Hadoop and Message Passing Interface (MPI) [23]. Different from the Mesos which builds an extra resource management layer to support different cluster computing frameworks, resource is isolated to support on-demand deployment of clusters in [18].
- In *Cloud Computing*, the essential task is to provide appropriate resources to any user connected into cloud. The resource allocation and scheduling are investigated in heterogeneous environment in [13] and [16]. In [14] dynamic resource allocation is discussed with distributed multiple criteria decision analysis.

B. Information Entities

On the other side, the researches about information resources are mostly focused on expression and ontology. The researches start from basic fundamentals like Resource Description Framework (RDF), Representational State Transfer (REST), to many different areas like Resource Space Grid (RSG), and Web resource database.

- *Resource Description Framework*[4] is a model describing Web resource by using formalized statements with Uniform Resource Identifier. It is proposed to support further research on web as a metadata system.
- *Representational State Transfer*[5] was firstly proposed by Roy Fielding who is one of the principle author of the HTTP specification to describe computer network architecture style. One of the most important concept in this work is that the resources are manipulated by transferring the representational states of the applications.
- *Resource Space Grid*[24] is a platform built based on Resource Space Model which treats any single resource as an entity located at multi-dimensional space where each dimension is one resource attribute.
- *Web resource database*[6] tries to discover suitable resources on internet more efficiently and accurately.

C. Dynamic Resource Supply Model

In complex environment like CPS systems [8], the resource is becoming more and more complicated. In [29], [30], a resource-centric architecture for service-oriented cyber physical system was proposed to treat resources as a first class object. The resources consumed in CPS are neither just CPU, memory and I/O, nor data entities. In the complex systems, the functions are provided from collaborating among humans, software systems, hardware components and other physical processes. Therefore in order to design a resource model used at CPS, we should present a unified resource model which can be applied in various application domains. In Dynamic Resource Supply Model we proposed at [32], the resource model contains three core parts: Resource Supply Description Template, which is to describe static attributes; Resource Instance Notation, which is to describe dynamic changing attributes; and Resource Operating Decision Machine, which is to manage all resources.

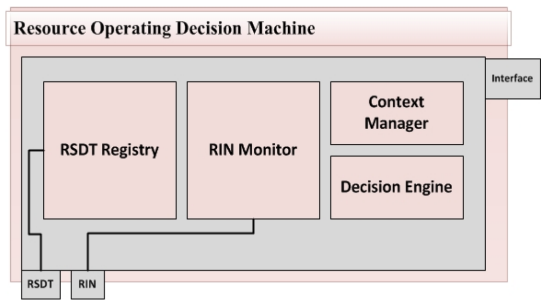

Fig. 1. Structure of Resource Operating Decision Machine

The Figure 1 is a basic structure of Resource Operating Decision Machine we proposed in [32] to manage all resources in Cyber Physical System. Any Resource Provider who wants to join into this Cyber Physical System has to register into *RSDT Registry* in RODM firstly. Then the *RIN Monitor* will monitor the resource providing situation. The *Decision Engine* can decide the resource usage policies and scheduling. Besides, this RODM can provide an interface to connect other system to manage related resources. However, we didn't provide how this decision engine work in the previous paper. In the section IV, the detailed modules in this RODM will be introduced to explain how to manage these resources, make decision, and connect to Cyber Physical Systems.

The Figure 2 is an architecture describing how to deploy this resource model into Cyber Physical Systems with Service Oriented Architecture. In this diagram, n RPs are providing the resources to a CPS through the access layer. Each RP holds several different RSDTs to provide different

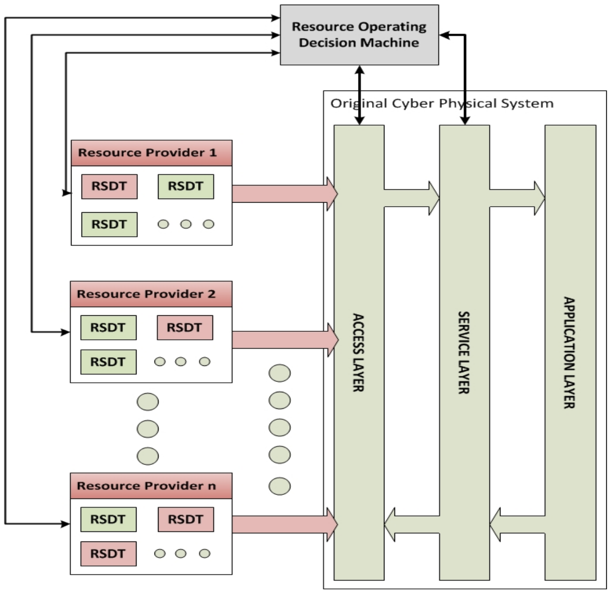

Fig. 2. Apply the RDSM into a Cyber Physical System

resources. The RSDT in red means this RSDT is providing resources at run time. Any RP has to register into the RODM and the RODM has to monitor the resource providing situation, so the RPs and access layer in CPS both have connections to RODM. Eventually, the services have the rights to check the resources and manage related contexts between resources and services; therefore RODM also has a connection to service layer in CPS.

III. Analysis of Dynamic Resource Supply Model

The Dynamic Resource Supply Model proposed by us in [32] is a basic model which lacks of details and mechanisms. In order to improve our DRSM, in this section we will first give a formal definition of resource and propose two key points in resource model from our definition. Next, DRSM will be compared with various resource models and allocation algorithms in different areas. Then, we will divide resource management approaches into two fundamental parts, that is, resource specification and resource usage specification. Finally the rationale of this division will be explained from both the theoretical definition and practical implementation.

A. Resource Definition

Below we will give a formal definition of resources and use this definition to analyse the important role of resources.

We assume there exist a system S, let

$$S(I) = O, \quad O \neq \emptyset \tag{1}$$

where $I = \{i_1, i_2, i_3...\}$ is a set of inputs, and $O = \{o_1, o_2, o_3...\}$ is a set of outputs of system S.
There is a function f, let

$$f(I') = O', \quad O' \neq \emptyset \tag{2}$$

where $I' = \{i'_1, i'_2, i'_3...\}$ is a set of inputs, and $O' = \{o'_1, o'_2, o'_3...\}$ is a set of outputs of function f, when $I' \subseteq I$, if $O' \subseteq O$, then $f \in S$.

$$(I' \subseteq I) \wedge (O' \subseteq O) \rightarrow (f \in S) \tag{3}$$

We assume the entire system S has n functions $(f1, f2, ...f_n)$ and the set of them is F which is the functional requirement of system S

$$F = f_1 \cup f_2 \cup f_3 \cup ... \cup f_n, \quad F \subset S \tag{4}$$

There is a set $R_f = \{r_1, r_2, r_3...\}$, let

$$\exists r \notin R_f \rightarrow f(I') \neq O' \tag{5}$$

then r is one of the resource to provide function f and R_f is the set of resources to provide function f. If the function $f \in S$, we name R_f the resources of system S.

From the equation 5, we can get

$$\begin{aligned} &\exists r \notin R_f \rightarrow f(I') \neq O' \\ \iff &f(I') = O' \rightarrow \forall r \in R_f \end{aligned} \tag{6}$$

The equation 6 means for any function in the functional requirements of system S, if the function can be implemented correctly, the required resources should be all provided. This result indicates that the resource is a *necessary condition* for the specified function f in system S.

As F is the requirements of system S, we have $F \subset S$, and $f \subseteq F$, from the 6, we know

$$S(I) = O \rightarrow \forall r \in R \tag{7}$$

This means the resource is a *necessary condition* of system S but not a *sufficient condition*. We can not promise the system will have no problem even if it has all required resources. However, the system must somehow go wrong when the resources are not enough.

In a simple system, the resources may not be considered individually because people prefer to model the system with *necessary and sufficient conditions*. In contrast, when the system is too complex, it's impossible to model and verify the system by using traditional modelling methods because of state explosion. In this situation, we need to figure out another way to model the system.

For a specified system S, we assume there is a set of conditions $C = \{c_1, c_2, c_3...\}$:

$$\forall c \in C: \quad (S(I) = O) \rightarrow c \tag{8}$$

if we have

$$C \rightarrow (S(I) = O) \tag{9}$$

the conditions set C is the *necessary and sufficient condition* for system S.

In this way, the design and verification of system can be decomposed into different sub-problems to simplify the system. When there are enough different proved *necessary conditions*, we can say we have very strong confidence that the system can run correctly. If we treat the resource as an important *necessary condition* in system S, the problem in this situation becomes how to find a proper set of resources $R' = R_1 \cup R_2 \cup R_3 \cup ...$, which can let any subset R_k satisfy:

$$\exists r \notin R_k \rightarrow f_k(I') \neq O' \tag{10}$$

where f_k is one of the functional requirements in system S.

To find all proper resources, *Resource Specification* is necessary, which we will introduce in the next Section.

After we find all required resources for this system, we still need to use these resources in right way. We define the process to use the resource as:

$$RM(R_{sub}, S) \Rightarrow State_{normal}(S) \quad (11)$$

If something is wrong, it will go:

$$RM(R_{sub}, S) \Rightarrow State_{exception}(S) \quad (12)$$

To simplify this question, we just use $State_{normal}(S)$ to express that the system S works normally and use $State_{exception}(S)$ to indicate that something goes wrong with the system. The design and verification of the system is to make sure the system state is always in normal. For the usage of resources in system, *Resource Usage Specification* is used to illustrate the model of resource usage.

B. Resource Specification

From the above analysis, the resource specification is used to confirm whether the system has all required resources to satisfy its requirements. In the definition, we treat all resources as a big set of elements containing several different subsets for different requirements of system. However, the resources in real world are much more complex and we can not treat all resources as single unconnected elements. Therefore in the Resource Specification, we need to:

1) Specify standards for the required resources in the system.
2) Check and decide whether the resource is correct.
3) Manage many different resources in an efficient and correct way.

After investigating the previous research results, we can see the resource specification is quite good for all unrelated static resources. The required resources usually have different schemas to represent different attributes of resources. For example, in the Mesos framework [23], a slave i will always report to the master that how much resources it owns. In equipments descriptions of operating systems, there are lots of different attributes to describe the equipment details. In our Resource Supply Description Template (RSDT), we use similar approach to describe required resources with defined template. However, the specification and management of complex dynamic related resources are much more complicated. For example, managing data resource like RDF, RSM or Semantic Web, needs to accurately express different resources in order to provide further ontologies and semantics while the traditional computational resources like CPU, memory or I/O do not have complex relations. In a more complex system like Cyber Physical System, both the fine-grained computational resources and other different resources which may have complex dependencies should be considered. All resources need to be described correctly to confirm all required resources can be provided to the system. In our RSDT, we have *Characters*, *Attributes* and *Other Resource in the context of Use*, which are different from other static characteristics to solve this problem. The *Characters*, which are connected with Resource Usage Specification, are to explain how to use this specified resource. The *Attributes*, which are connected with Resource Instance Notation, contain all attributes that will get real values when the system is running. Then the system can dynamic check and decide whether the resource is correct. The *Other Resource in the context of Use* has to manage all dependencies between different resources. However two further problems remain unsolved :

1) How can *Characters* be used to express the use of this resource?
2) How can resource dependencies be managed?

In order to support resource usage, resource usage specification is necessary to indicate how to use a specific resource(see below Section III.C). For resource dependencies, our current approach of using $key - value$ is not powerful enough. We are considering to design a specified language to express the relations between resources and even further ontologies.

C. Resource Usage Specification

Resource Usage Specification defines how the system should allocate and use resources, using the different algorithms and policies. From the definition we defined at section III.A, the mechanisms about using these resources should be specified to confirm that the system does not go to exceptional states. In our Resource Model, because resource types are unlimited, the Resource Usage Specification is a critical and difficult problem. To solve this problem, we need to consider the following three aspects:

1) Whether the use of resource now is correct?
2) What kind of resource allocation mechanism is better?
3) How can the system decide to apply a particular mechanism or policy?

To solve the first question, a checker is needed to act as a resource policy filter, which can confirm everything works correctly. For example, the *Assignment Manager*, *Resource Selector* and *Match Maker* designed at [27] support correct resource allocation. For the second question, it depends on the environment and context. In many resource management approaches, all required resources in the system are consumed for several different tasks and the best resource usage is to make resource consumption efficient and accurate [7], [9], [13], [14], [16]. In the complex distributed system, people seek to make the most fairness in resource allocation [15]. Sometimes in the energy limited environment, managing energy is a first class resource in the system [21]. In the complex system, the better resource allocation mechanism usually depends on the environment and context. Therefore we need to decide which requirement is more important than others and choose the right resource allocation mechanism. For example, if the system requires high-level security, then we need to choose the best resource allocation mechanism which can pursue and guarantee security of the system. To support this kind of changing resource usage methods, we need a more flexible way to describe how to use related resources. In [10] and [28], resource usage policy specification languages are provided to specify different types of resource usage policies. In our Resource Operating Decision Machine, we also need to design a specified resource usage specification language for specifying different types of resource usage policies in complex environment.

IV. Improved Dynamic Resource Supply Model

Based on the above analysis and reflection, we propose a supplement to the original basic Dynamic Resource Supply

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

TABLE I
RESOURCE SUPPLY DESCRIPTION TEMPLATE

Template Holder: <*the resource provider*>
Template Unify ID: <*the identity of this template*>
Resource Name: <*the provided resource name*>
Characters:
{*to help decide how to use this resource*}
Attributes:
{*can be inherited to the resource instance*}
Cost: *cost per unit*
Availability:
{*a set of constraints like time, location, etc.*}
Legal Rules for Supply: *URI to a website*
Other Resource in the Context of Use:
{*a set of context resource dependencies*}

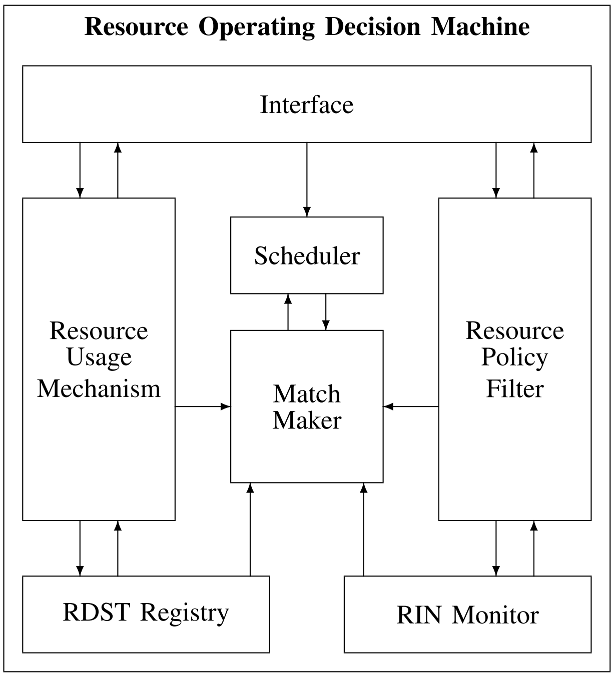

Fig. 3. The Structure of Resource Operating Decision Machine

Model in this section. Resource Model consists of three parts: Resource Supply Description Template, Resource Instance Notation and Resource Operating Decision Machine. The RSDT is defined in Table I.

This kind of resource template can be held by the resource provider and when the resource provider is added into the Cyber Physical System, the RSDT will register at the Resource Operating Decision Machine. The basic structure of RODM remains the same while the Decision Engine will contain more modules inside to make correct decisions. The Figure 3 indicates the detailed architecture of RODM.

Because of the nature of Cyber Physical System, we should design an architecture to support a distributed, large-scale, and heterogeneous system. In this RODM, in addition to the RSDT Registry and RIN Monitor, the Decision Engine is divided into four parts to make high-level decisions based on real-time context. The four parts are:

1) *Resource Usage Mechanism* is the module which can analyse the information from RSDT and decide how to use the resources. Because this RODM machine has to make decisions at the high-level, the Resource Usage Mechanism needs special *Resource Usage Specification Language* to load the mechanism from the registered RDST. There are some researches about resource usage specification language in distributed system or virtual machines[10], [28]. However, our resource usage environment is different from distributed computing. Because of the complexity of resource types and characteristics, we need to design a new language to specify the usage mechanism of resources.
2) *Resource Policy Filter* is the module connected with RIN Monitor and supports the Exception Handler for RIN Monitor. This module is a critical important module in RODM to verify whether the supported resources are correct for the tasks from the Cyber Physical System. It is used to control the resource quality and then guarantee the security of resource management model. If a resource instance can not pass this resource policy filter, the RODM machine will start the Exception Handler to handle the errors.
3) *Scheduler* is the connector between the Cyber Physical System and this RODM machine. All resources managed by the RODM machine need to provide some specified task from the Cyber Physical System through the interface. Because both the cyber physical system and resources used by the system could be distributed, large-scale, and heterogeneous, the tasks need to be arranged to allocate suitable resources correctly. This Scheduler module can arrange a large amount of different tasks coming from the interface.
4) *Match Maker* is the core decision engine to choose the correct resources to support related tasks. Different task needs different resources and this module will analyse, calculate and decide the eventual resource allocations. It has the connections to all other modules to support complicated decisions.

V. CONCLUSION

In this paper we compared the resource management approaches in different fields and apply some of their design virtues to improve our Dynamic Resource Supply Model. With the formal definition of resource in this paper, we proposed two significant aspects of resource model: Resource Specification and Resource Usage Specification. Our Dynamic Resource Supply Model is discussed from our definition and these two aspects with some principles are proposed in the model for further extension. Inspired by past research results, the original simple Resource Operating Decision Machine is extended with several modules including Resource Usage Mechanism, Resource Policy Filter, Scheduler and Match Maker. Through analysis and comparison, we realize the necessity of designing a specified resource usage specification language for specifying the usage mechanism of resource and this will be our on-going future work.

REFERENCES

[1] National institute of standards and technology, *Foundations for Innovation in Cyber-Physical Systems*, (Rosemont, Illinois), energetics incorporated Columbia, Maryland, January 2013.
[2] D. E. C. Janos Sztipanovits, John A. Stankovic, ed., *CCC Workshop on New Forms of Industry - Academy Partnership in CPS Research*, (George Mason University), 2009.
[3] S. S. Sunder, "Foundations for innovation in cyber-physical systems." NIST CPS Workshop in Chicago, March 2012.

ISBN: 978-988-19253-3-6
ISSN: 2078-0958 (Print); ISSN: 2078-0966 (Online)

[4] G. Klyne, J. J. Carroll, and B. McBride, "Resource description framework (rdf): Concepts and abstract syntax," *W3C recommendation*, vol. 10, 2004.

[5] R. T. Fielding, *Architectural styles and the design of network-based software architectures*. PhD thesis, University of California, 2000.

[6] B. Yuwono and D. L. Lee, "Wise: a world wide web resource database system," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 8, no. 4, pp. 548–554, 1996.

[7] M. B. Jones, P. J. Leach, R. P. Draves, and J. S. Barrera III, "Modular real-time resource management in the rialto operating system," in *Hot Topics in Operating Systems, 1995.(HotOS-V), Proceedings., Fifth Workshop on*, pp. 12–17, IEEE, 1995.

[8] E. A. Lee, "Cyber-physical systems-are computing foundations adequate," in *Position Paper for NSF Workshop On Cyber-Physical Systems: Research Motivation, Techniques and Roadmap*, vol. 2, Citeseer, 2006.

[9] R. Raman, *Matchmaking frameworks for distributed resource management*. PhD thesis, UNIVERSITY OF WISCONSIN, 2001.

[10] P. Uppuluri and A. Gupta, "Resource usage policy specification for managing application resources," in *Information Technology: New Generations, 2008. ITNG 2008. Fifth International Conference on*, pp. 881–887, IEEE, 2008.

[11] C. Liu, L. Yang, I. Foster, and D. Angulo, "Design and evaluation of a resource selection framework for grid applications," in *High Performance Distributed Computing, 2002. HPDC-11 2002. Proceedings. 11th IEEE International Symposium on*, pp. 63–72, 2002.

[12] P. Uppuluri, N. Jabisetti, U. Joshi, and Y. Lee, "P2p grid: service oriented framework for distributed resource management," in *Services Computing, 2005 IEEE International Conference on*, vol. 1, pp. 347–350, IEEE, 2005.

[13] G. Lee, B.-G. Chun, and R. H. Katz, "Heterogeneity-aware resource allocation and scheduling in the cloud," in *Proceedings of the 3rd USENIX Workshop on Hot Topics in Cloud Computing, HotCloud*, vol. 11, 2011.

[14] Y. O. Yazir, C. Matthews, R. Farahbod, S. Neville, A. Guitouni, S. Ganti, and Y. Coady, "Dynamic resource allocation in computing clouds using distributed multiple criteria decision analysis," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*, pp. 91–98, Ieee, 2010.

[15] A. Ghodsi, M. Zaharia, B. Hindman, A. Konwinski, S. Shenker, and I. Stoica, "Dominant resource fairness: fair allocation of multiple resource types," in *USENIX NSDI*, 2011.

[16] G. Lee, *Resource Allocation and Scheduling in Heterogeneous Cloud Environments*. PhD thesis, University of California, Berkeley, 2012.

[17] K. Czajkowski, S. Fitzgerald, I. Foster, and C. Kesselman, "Grid information services for distributed resource sharing," in *High Performance Distributed Computing, 2001. Proceedings. 10th IEEE International Symposium on*, pp. 181–194, IEEE, 2001.

[18] B. Ghit, N. Yigitbasi, and D. Epema, "Resource management for dynamic mapreduce clusters in multicluster systems," in *High Performance Computing, Networking, Storage and Analysis (SCC), 2012 SC Companion:*, pp. 1252–1259, IEEE, 2012.

[19] K. Czajkowski, I. Foster, N. Karonis, C. Kesselman, S. Martin, W. Smith, and S. Tuecke, "A resource management architecture for metacomputing systems," in *Job Scheduling Strategies for Parallel Processing*, pp. 62–82, Springer, 1998.

[20] D. R. Engler, M. F. Kaashoek, *et al.*, "Exokernel: An operating system architecture for application-level resource management," in *ACM SIGOPS Operating Systems Review*, vol. 29, pp. 251–266, ACM, 1995.

[21] H. Zeng, C. S. Ellis, A. R. Lebeck, and A. Vahdat, "Ecosystem: Managing energy as a first class operating system resource," in *ACM SIGPLAN Notices*, vol. 37, pp. 123–132, ACM, 2002.

[22] C. A. Waldspurger, "Memory resource management in vmware esx server," *ACM SIGOPS Operating Systems Review*, vol. 36, no. SI, pp. 181–194, 2002.

[23] B. Hindman, A. Konwinski, M. Zaharia, A. Ghodsi, A. D. Joseph, R. Katz, S. Shenker, and I. Stoica, "Mesos: A platform for fine-grained resource sharing in the data center," in *Proceedings of the 8th USENIX conference on Networked systems design and implementation*, pp. 22–22, USENIX Association, 2011.

[24] H. Zhuge, "Resource space grid: model, method and platform," *Concurrency and Computation: Practice and Experience*, vol. 16, no. 14, pp. 1385–1413, 2004.

[25] G. Oppenheimer and N. Weizer, "Resource management for a medium scale time sharing operating system," in *Proceedings of the first ACM symposium on Operating System Principles*, pp. 14–1, ACM, 1967.

[26] G. Czajkowski, S. Hahn, G. Skinner, P. Soper, and C. Bryce, "A resource management interface for the java platform," *Software: Practice and Experience*, vol. 35, no. 2, pp. 123–157, 2005.

[27] M. Raunak and L. Osterweil, "Resource management for complex, dynamic environments," 2013.

[28] N. Damianou, N. Dulay, E. Lupu, and M. Sloman, "The ponder policy specification language," in *Policies for Distributed Systems and Networks*, pp. 18–38, Springer, 2001.

[29] K. Wan and V. Alagar, "A resource-centric architecture for service-oriented cyber physical system," in *Grid and Pervasive Computing*, pp. 686–693, Springer, 2013.

[30] K. Wan and V. Alagar, "Modeling resource-centric services in cyber physical systems," in *Proceedings of the International MultiConference of Engineers and Computer Scientists*, vol. 2, 2013.

[31] D. Georgakopoulos, and M. P. Papazoglou, "Service-oriented computing," *The MIT Press*, 2008.

[32] Y. Dong, K. Wan and Y. Yue, "A Dynamic Resource Supply Model towards Cyber Physical System (CPS)," in *2014 International Symposium on Computer, Consumer and Control (IS3C)*, 2014.