Moving To The Cloud:
Transitioning From Client-Server
To Service Architecture

Manoj A. Thomas, Virginia Commonwealth University, USA
Richard T. Redmond, Virginia Commonwealth University, USA
H. Roland Weistroffer, Virginia Commonwealth University. USA

ABSTRACT

This paper makes the case that the traditional three-tier client-server architecture requires a major overhaul to address the changing and rapidly increasing information processing and services needs of consumers. Revisions to the conventional architecture model are suggested and two examples of information systems applications are discussed to illustrate how the new information service architecture fits into the realm of future systems development.

INTRODUCTION

E-commerce and e-business implementations have traditionally relied on variations of the customary three-tier systems architecture model for developing and deploying client-server applications. With the advancement in software and hardware capabilities, the three-tier model is being challenged in two major ways. First, in today's highly connected world, mobility has become the baseline of computing standards. Users are demanding more and more freedom and flexibility in accessing personal and professional data without the inconvenience and security concerns surrounding cumbersome mobile computing platforms. Second, there is an unmistakable blurring of the line that distinguishes Internet based applications from stand-alone desktop applications. With the advent of *rich Internet applications* (*RIA*), software routines tend to look and perform the same, irrespective of whether the user is at his or her office desk or in a mobile mode at an Internet kiosk. The new generation of hybrid software applications provides the user with an experience that shows no difference whether used as a stand-alone computer based application, or used on a mobile smart phone. Industry analysts peg the number of Internet connected personal computing devices at 1.2 billion, and growing rapidly [Markoff 2008], accounting for almost one personal computer for every six people worldwide.

Besides RIA, two other phenomena are making stealthy inroads into the confines of every-day computing experiences of the ordinary person – *cloud computing* and *knowledge representation*. Cloud computing allows the public access to massive computing power at negligible costs. These computing resources are accessed via the Internet, hence the "cloud" metaphor, which reflects the diagrammatic use of a cloud to represent the Internet in many graphic depictions [Katzan 2008]. Knowledge representation, simply put, embraces the use of descriptive web annotating languages to mimic social reality in a machine processable manner. Surprisingly, neither of the two are inventions of the twenty-first century. The seeds for both were sewn very early in academic and research settings, long before the Internet. Something similar to cloud computing has been in existence in seemingly crude, yet relatively functional forms under various names such as distributed computing, clusters, and grid computing. Many commercial organizations and government research installations have consistently allocated vast resources for the development of software that can take advantage of the processing power of large numbers of server clusters (e.g. Oracle 10g for grid computing). Knowledge representation, on the other hand, falls under the bigger umbrella of *artificial intelligence* (*AI*). The deductive reasoning and problem solving capabilities inherent in AI have been used in numerous real world applications, such as robotics, natural language processing, cybernetics, and simulation of organic functions, stretching the human imagination. AI mostly relies on complex algorithms that may involve artificial neural networks, decision trees, or decision frames, among other techniques, to match facts (i.e. data)

against rule sets to infer conclusions that can then lead to actions. Knowledge representation is the area of AI that concerns itself mainly with the codification of semantics (over syntax) of a chosen domain of interest (e.g. pharmacogenomics, cancer research, or consumer behavior). *Expert systems* can then use these knowledge structures to infer conclusions based on the algorithms employed.

As a sub-discipline of knowledge representation, *ontologies* have emerged as an effective means for expressing domain concepts, concept properties, relationships among concepts, restrictions on properties, and for modeling real world facts. The computational ability to reason from the semantically shared ontological description of a domain of discourse, allows information foraging activities, such as contextual fact retrieval, data exploration, and data exploitation to be more effective. *Web ontology language* (*OWL*), an annotation language, has evolved as a popular choice for the semantic knowledge representation in ontologies. OWL is standardized by the *World Wide Web consortium* (*W3C*) and uses first order descriptive logic to create expressive statements that describe the domain.

The technological innovation referred to as cloud computing promises to provide software services and customizable hardware configurations for public access, similar to public utilities providing electricity, water, and high-speed Internet connections to the general community. The consumer will find himself faced with enormous levels of computing resources that up until very recently were only available to the most privileged of scientists working at heavily funded research institutes or elite military installations. The innovative entrepreneur will no longer be held up by limited computing resources, but instead be lavished with the convenience and abundance of number crunching capabilities from massively distributed parallel clusters of computers. Already Google and IBM, for example, have dedicated clusters of several hundred computers solely for the use of computer science students [Weiss 2007].

The objective of this paper is three fold. First, we aim to justify the need for revising the traditional, three-tier architecture concept, to address the drastic changes in information processing demands by the consumer and the increased information processing services supplied to the public. Second, we propose changes to the conventional client-server concept to deal with these new requirements. Third and finally, we illustrate how the proposed new architecture fits into the realm of future systems development via examples of information systems implementations.

The rest of the paper is structured as follows. Section two highlights innovations and technological advancements in the computing field that have resulted in new kinds of user expectations and behaviors, as motivation for proposing the next natural progression in systems architecture – the information service architecture. Section 3 then describes two example scenarios to validate the significance and aptness of the information service architecture. We conclude by highlighting once again the importance of the information service architecture, and point out some limitations that need to be addressed in future research.

INFORMATION SERVICE ARCHITECTURE

Initiatives like the *cloud computing interoperability form* (CCIF 2009), promoted by industry stakeholders, very likely will further the portability, integration and security specifications for the cloud based computing landscape. The concept of stand-alone computer applications has almost reached its end-of-life, with a few lingering exceptions, such as legacy applications maintained for the sake of providing support for vintage technology, critical security systems with dedicated desktop-server links, and systems in remote geographical locations where communication infrastructure creates a resources bottleneck. Recent developments in hardware and software have brought about a new approach to the way we interact and apply computing technology. In this new era of perpetually connected systems, powerful computers, cheap storage, and seemingly boundless network bandwidth, it is easy to challenge the continuing suitability of the classical three-tier client-server models, let alone their more complex multi-tier derivatives [Özsu et al. 1999]. This forces us to rethink the traditional client-server model from the ground up. Consequently, the need for a pragmatic change in architecture, wherein the concepts of cloud computing, knowledge representation, and asynchronous data access are integral components, is becoming increasingly evident.

While the three tiers of a conventional client-server environment generally consist of a *data layer*, an *application/business logic layer*, and a *presentation layer*, there are compelling reasons to re-organize these into a new architecture consisting of a *cloud layer*, a *knowledge layer* and a *presentation layer*, where the cloud layer encompasses both applications and data. The proposed information service model comprising of these three layers is shown in Figure 1.

The Cloud Layer

By providing platform independence, the cloud layer hosts the cloud computing services, comprised of the application support and the database services. Cloud computing may be defined [Armbrust et al. 2009; CCIF 2009] as the practice of running applications on massively parallel and distributed computer systems that are ubiquitously accessible to end-users without requiring definitive knowledge of the underlying hardware physicality or geographical location.

With the growing class of applications that are dynamically driven by customization, based on personal preferences, contextualization, and location awareness, the persistent access to data structures become an integral part of the application design. With the advent of cloud computing, the applications that used to make up the application/logic layer, executing transactional logic on data that used to be in the data layer, are no longer restricted to disparate territories, but rather are stretched across the entirety of the all encompassing and distributed cloud.
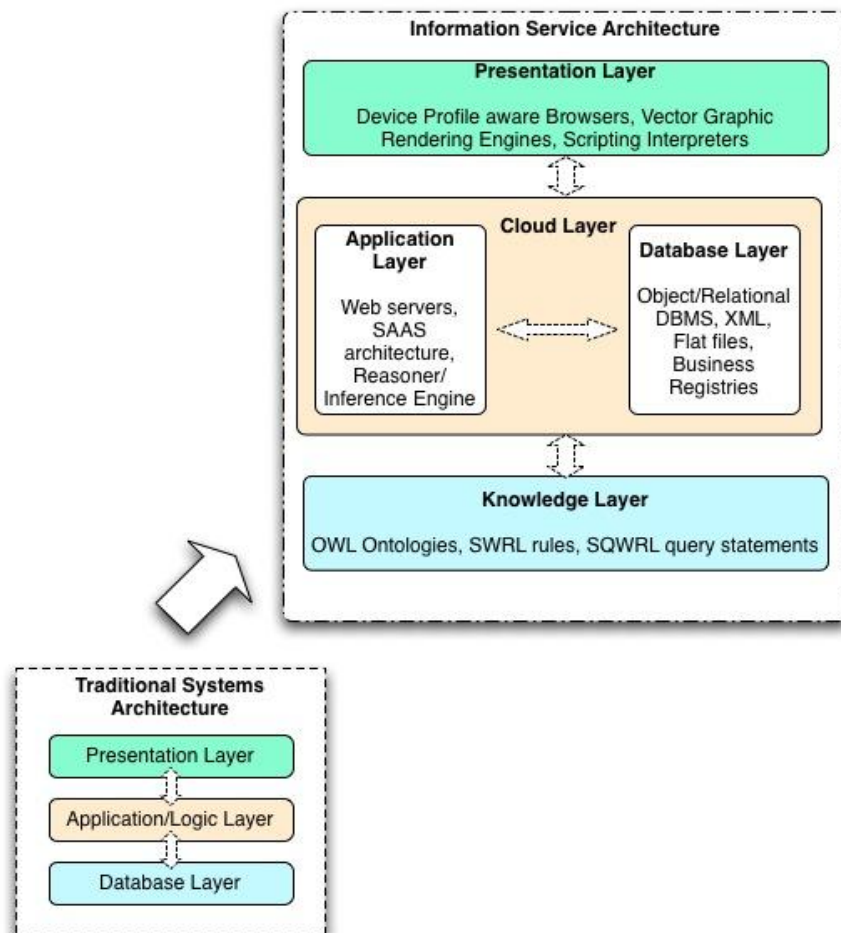

Figure 1. Information Service Architecture

The cloud layer provides a sandbox environment with arbitrary boundaries to ensure *autonomy*, *security* and *resource elasticity*. Autonomy is the freedom to assert computing capacity independently from other shared systems in the cloud (design autonomy), and the liberty to model the space with minimally stipulated restrictions as the computing needs change (ownership autonomy). Autonomy allows individuals to access their computing environment globally and control resource allocations remotely. The *Amazon EC2* service is a good example of such a service. Google's much anticipated *G-drive* is another example, albeit restricted to file storage operations. Security (in the cloud) refers to the ability to clearly distinguish and separate computing spaces within the bigger cloud of openly shared public resources. No cloud layer will be complete without a well-defined security manifest that upholds the notion of confidentiality, integrity and availability of data commensurate with any other organizational information systems security model [Bishop 2002]. Apple's *MobileMe* is a good example of a cloud service that synchronizes information between all personal devices in a secure reliable manner. Resource elasticity (in the cloud) is the economic flexibility to assemble and adapt computing needs to rapidly match the changing processing demands. Back-office operations are always awash with stories regarding the hellish task of building an enterprise server for immediate deployment, let alone overcoming the myriad hoops of corporate purchase authorizations. Resource elasticity allows the user to configure capacity on the fly, at affordable rates that are appraised, based on usage rather than on hardware costs. In essence, what we see is a shift towards a *centric-decentric* model for the computing resources. Centric because the massive data centers at various physical locations that power the cloud are managed by individual solution providers. Decentric because access is not just limited to one centralized location, but more similar to a mobile computing environment, available to every single cloud user at any give time.

Most large companies, such as Google, Amazon, IBM, and Microsoft, have made an entrance into the arena of cloud computing. While these companies benefit from the revenue generation that results from making available the otherwise idle cycles of their vast computing clusters at various data centers, there is also opportunity for overall economic gains by allowing any entrepreneurial mind to implement solutions requiring large resource allocations, previously a major financial inhibitor that limited the fruition of innovative ideas.

In short, the cloud layer does away with the boundaries between the application/business logic layer and the data layer of traditional models. The commercial feasibility of seamlessly combining these fundamental services into a single service has proven viable and cost effective as evidenced by commercial offerings such as Amazon's *elastic computing cloud (EC2)* services.

Knowledge Layer

The knowledge layer represents the organization of domain knowledge, independent of the cloud layer. This layer represents a repository of concept definitions – ontologies – that describe domain knowledge in a machine readable and human interpretable format. The term "ontology" in philosophy refers to the study of the nature of existence. Studer et al. define ontology as "a formal, explicit specification of a shared conceptualization," where *conceptualization* designates an abstract model of some phenomenon, identifying the relevant concepts of that phenomenon; *explicit* means that the relevant concepts, and the constraints on their use, are clearly and openly defined; *formal* implies that the ontology should be machine-readable; and *shared* reflects the notion that an ontology captures consensual knowledge, accepted by a group, rather than being private to some individual [Studer et al. 1998]. Ontologies formalize concepts (as *classes*), relations among classes, constraints (as *axioms*), and instances (as *individuals*), in a domain of interest. OWL is the more popular of descriptive languages used for expressing the construct formalisms.

Ontologies are now extensively used in applications like multimedia organization for distributed collaborative filtering [Wang 2006], personalization and recommendations systems, collaborative knowledge acquisition [Kuntz 2005], geographic information systems [Visser et al. 2002], and semantic information search and retrieval on the Internet. The benefits of using a common vocabulary to describe domain concepts have been proven beyond just academic research projects, and there is wide adoption of ontologies in commercial software. Unlike the sandboxed nature of the cloud layer, the descriptive ontologies in this layer are openly shared and freely accessible to all member instances from any cloud.

Presentation Layer

The latest technological innovations have also brought forth facelifts at the presentation layer. With on-demand computing, or *I-Services* [Katzan 2008], the consumer is relieved of the burden of software installation, hardware purchase, and maintenance. As the web browser is the primary tool for the end user to access the hosted services, there is strong momentum to develop an environment in which the browser can interpret and carry out complex, dense, and highly responsive user interactivity functions. Real-time data analysis, gauges and controls, contextual scripting engines for search query assistance, 3D graphing and charting, scalable vector graphic page content and menus, and asynchronous offline capabilities are all examples of features available through the browser that were previously limited to desktop based client applications.

This need for an add-on browser support environment is mainly due to traditional HTML falling short on functions that require complex information processing or intense image rendering. Implementing even simple, dynamic interactivity features on a web page requires the use of scripting languages, in addition to the hypertext languages used for presentation and formatting. For the level of user interaction that is now expected at the presentation layer, this calls for a robust, yet lean, support and rendering engine that can be invoked by the browser to service the lightweight, powerful web applications in a security-sandboxed and performance-optimized manner. The rendering engines are generally installed as add-ons or plug-ins to the browser. They ensure that the browser is not overtly tasked with the transaction overhead of information exchange (e.g. asynchronous data exchange and XML post-back, script language interpretation for *AJAX*, and Adobe's *ActionScript*) with the cloud layer. The presentation layer of the information service architecture model thus primarily addresses the organization of the browser environment, rendering engines, and other rich Internet application that can operate in an offline (asynchronous) or online (synchronous) mode, on a varying range of platforms, ranging from desktop workstations or tablet PC to low powered, small profile devices, like smart phones and netbooks.

APPLICATION SCENARIOS

The proposed information service architecture represents an effective model for most internet based computing tasks, like retrieving search results from queries, contextual information sharing, knowledge exploration, or even social collaboration on the web. Two application scenarios are described below that illustrate the adaption of the information service architecture in the design, fabrication, and organization of future web applications.

Federated Information System

Federated information systems (*FIS*) is the area of research that focuses on developing methods and tools that can be used to provide integrated access to a finite, predefined set of autonomous and heterogeneous data sources [Busse et al. 1999]. The participants operate independently, but negotiate autonomy among them in order to participate in the federated environment. The integration layer [Sheth et al. 1990; Busse et al. 1999] that enables the requisite levels of autonomy and interoperability among the underlying applications can be housed in the cloud layer. The cloud relies on virtualization techniques [Armbrust et al. 2009] to host instantiations of fully functional server environments that are intrinsically sandboxed for security and autonomy. Server virtualization in this environment can utilize the speed and processing capacity of the hundreds of thousands of processing units that are cohesively bound by the cloud.

The ontologies in OWL-S format at the knowledge layer store semantic information about the data sources. Semantic operations such as knowledge fusion, inferences, abstraction and supplementation [Busse et al. 1999] are carried out in the knowledge layer. Given the dynamic nature of federated information system, tight coupling between systems are limiting and inadequate for operations that require contextual semantic inferences. Maintaining a shared knowledge layer with open access ensures the requisite semantic homogeneity irrespective of the structural heterogeneity in the data model stored in the cloud. FIS typically employ software agents to facilitate service discovery and to automate the extensibility of the federated schema [Thomas et al. 2007]. The agents reside in the cloud layer, communicating with the other agents in the various server instances. They act upon ontological inferences on the global knowledge, to interact with the appropriate data sources and to offer collective services to

the participating organizations in the FIS. An FIS implementation using the information service architecture is shown in Figure 2.
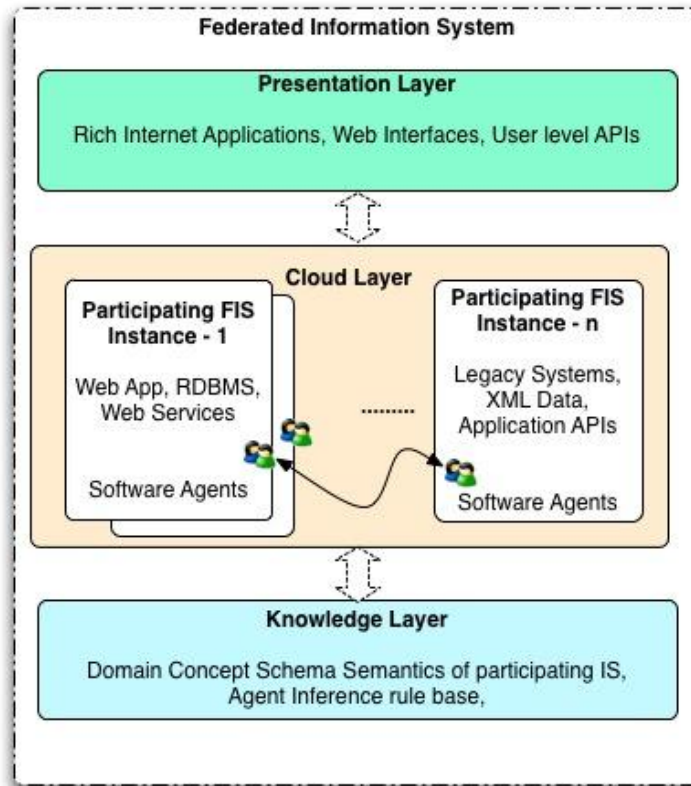

Figure 2. A Cloud Based Federated Information System

Comprehensive Emergency Response System

Crisis management requires the collaboration and coordination of different groups, such as fire marshals, paramedics, first response teams, Environmental Protection Agency personnel, etc., working together to disseminate information and provide diverse services to people in and around areas where crises have occurred. Crisis scenarios are complex, geographically bound, and invite broad repercussions. The information services architecture provides a sound footing suited for developing comprehensive response management systems that are geographically distributed through the cloud, equipped with real-time communication features and reporting capabilities with minimal user intervention. Existing emergency response management systems mostly support unidirectional messaging, with very little or no interactive capabilities, and require extensive human intervention. Research initiatives in the area of emergency response management systems focus on effectively integrating *geographic information systems* (*GIS*) maps, *VoiceXML*, and bidirectional communication, with the primary objective that a crisis must be contained by a specific point in time (an action threshold), after which the consequences of the problem are inescapable [Thomas et al. 2005].

To effectively reach out to the populace in case of a disaster, and to correlate firsthand inputs from the crisis location, the information service architecture relies on the persistent GIS data in the cloud layer. The application logic in the cloud assembles inputs from the field to synthesize decision predicates, such as generating reports, sequencing event logs, predicting evacuation routes, simulating scenarios, and using clinical data points, for the crisis response teams. Unsolicited and solicited voice data is transposed (*automated speech recognition*)*,* and

translated (*text-to-speech*) using VoiceXML parsers at the cloud layer to make them machine processable [Thomas et al. 2005]. The knowledge layer maintains the semantics for the crisis management and response action domain. A reasoner engine classifies incident data (field inputs) as *individuals* (instances) in the ontology. Contextual crisis management information is then inferred by means of semantic matching with the various, well defined and globally accessible resolution ontologies, stored in the knowledge layer.

As the software becomes a service in the cloud, the response teams gain the benefit of fast and geographically unconstrained access to critical information, necessary to coordinate the pre, current, and post crisis management activities. The software interface for the crisis management information system is accessible via any portable computing platform that has the requisite rendering engines installed. With GIS data sets being graphically dense and visually rich, the presentation layer in this example, at a minimum, requires parsers for XML based user interface language interpretation (e.g. Mozilla *XUL*, Microsoft *XBAP*), along with scripting engines with built-in support for vectored graphics (e.g. Adobe *Flex*, Microsoft *Silverlight*). These are readily available and free downloads from the commercial software vendors. A comprehensive emergency response system implemented using the information service architecture is show in Figure 3.
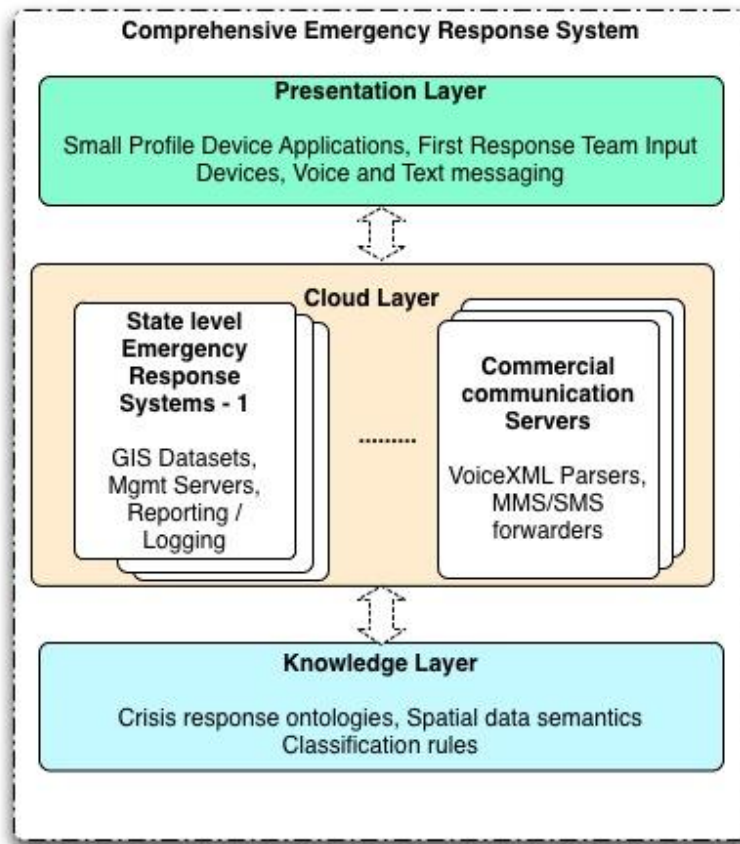

Figure 3. A Cloud Based Comprehensive Emergency Response System

CONCLUSION

The proposed information service architecture is an apt model for future application processing and information representation systems, as the shift from standalone PC based applications to Internet based web applications is progressing. This transition, together with the increasing popularity of low-cost computing devices

(i.e. netbooks with low power processors and solid-state devices, as opposed to notebooks with higher processing power and storage capacity) is begetting a new paradigm of computer usage, in which rich Internet applications artfully communicate with inexpensive, wirelessly connected, mobile computing platforms, with expert awareness of the users' contexts and geographic locations (Apple *mobile platforms* offers a myriad array of applications that refine knowledge filters based on location and geographical awareness). New software development initiatives are already in motion to develop *web operating systems*, with the intent of fully moving data and applications to the cloud. Microsoft recently released the *Azure* services platform that promises to make this vision a reality. *Amazon Machine Image (AMI), OpenLazlo*, and *Xcerion* are other similar solutions, with the goal of offering fully functional operating systems accessible through a browser interface. Google *Gears* and Adobe *Air* allow users to access web based information and services even when computers are not connected to the Internet. Microsoft *skyDrive* services allow users to store, share and backup data on the cloud. Recently IBM announced the release of its popular *Tivoli Management Services* for the cloud. As these software development platforms scale further to take advantage of the cloud services, internet based applications that once, due to computational limitations, only existed in the human imagination, will soon turn into reality. High definition video processing, localization tools for page translations, personalization algorithms relying on very elaborate rule sets, that until now existed only in experimental research settings, become general applications that are feasible, realistic, and relevant.

Large enterprises may still prefer to host and manage their own computing infrastructure for a variety of reasons, such as confidentiality, security and legal concerns. However, the maturity, timeliness, cost effectiveness, and flexibility of *service-on-demand* (*software as a service*, or *SAAS* [Armbrust et al. 2009]) models and *elastic server* systems that can be provisioned and deployed in a matter of minutes, is sure to challenge the few remaining benefits of in-house, self-maintained, back-office operations.

The two examples presented in this paper are proof of concepts used to illustrate the validity of the information service architecture in modeling complex information systems. Although cloud computing, ontologies and rich Internet applications exist in simpler forms today, the proposed model provides a basis for more general and complex applications.

Cloud computing is not fully void of obstacles either. Organizations tend to place stringent constraints on technology investments to moderate and control the operating costs of data centers. Expenditures on cooling, power consumption, and maintenance, necessary to ensure the continuity of operations of a vast mesh of interconnected cloud computers can be monumental. Armbrust et al. [Armbrust et al. 2009] list ten critical factors that limit growth opportunities in cloud computing. They range from strategic corporate concerns pertaining to information availability, data confidentiality, and performance viability, to software issues such as licensing models and reliability issues. Reining the operating costs while extending unlimited computing resources to the masses will remain a substantial challenge for the current SAAS providers and those planning to enter the potentially lucrative market arena of cloud computing.

AUTHOR INFORMATION

Manoj A Thomas is a faculty and Director of Technology in the Department of Information Systems at Virginia Commonwealth University. He holds a Ph.D in Information Systems, a Masters of Science in Information Systems as well as a Masters of Business Administration. His current research interests focuses on semantic web, knowledge engineering, and distributed computing. He has published in many practitioner and academic journals and presented at various Information Systems conferences. An electronic engineer by trade and an avid travel buff by choice, he rides and works on motorcycles when he wants to get away from the digital realm.

Richard Redmond is Chairman and Associate Professor, of the Department of Information Systems in the School of Business at Virginia Commonwealth University. His publications and research have been in the areas of software engineering, database systems, expert systems, agent-based systems, and applications of AI to business. More recently his focus has turned toward the semantic web and multi-agent based systems.

Heinz Roland Weistroffer is an Associate Professor of Information Systems in the School of Business at Virginia Commonwealth University in Richmond. His research interests include economics of information technology,

computer assisted decision support, security models, and systems development. He has published in IEEE Transactions on Software Engineering, the Journal of Multi-Criteria Decision Analysis, Annals of Operations Research, Socio-Economic Planning Sciences, Computational and Mathematical Organization Theory, Journal of Information Systems Security, Journal of Computer Information Systems, and Information Technology for Development, among many other journals, as well as in numerous conference proceedings such as AMCIS, HICSS, ECIS, ECITE, and DSI.

REFERENCES

1.   Armbrust, M., A. Fox, R. Griffith, A.D. Joseph, R.H. Katz, A. Konwinski, G. Lee, D.A. Patterson, A. Rabkin, I. Stoica, and M. Zaharia (2009) "Above the Clouds: A Berkley View of Cloud Computing," Technical Report UCB/EECS-2009-28, University of California, Berkley, available at http://www.eecs.berkeley.edu/Pubs/TechRpts/2009/EECS-2009-28.html.
2.   Bishop, M. (2002) *Computer Security: Art and Science*, Addison-Wesley Professional, 1st Edition.
3.   Busse, S., R.-D. Kutsche, U. Leser, and H. Weber (1999) "Federated Information Systems: Concepts, Terminology and Architectures," research report 99-9 of the Fachbereich Informatik, Technical University Berlin, available at http://www.cis.cs.tu-berlin.de/~sbusse/publications/1999/BKLW99.pdf.
4.   CCIF - Cloud Computing Interoperability Forum, (2009), accessible at http://www.cloudforum.org.
5.   Katzan, H. (2008) "Cloud Computing, I-Service, And IT Service Provisioning", *Journal of Service Science* 1(2), 57-64.
6.   Kuntz, C. (2005) "An Integrated Approach for Semantics-driven Information Retrieval," in Proceedings of the 11th International Conference on Human-Computer Interaction, Las Vegas, Nevada, USA, 22-27.
7.   Markoff, J. (2008) "Adobe Blurs Line between PC and Web," *The New York Times*, February 25, 2008.
8.   Özsu, T. and P. Valduriez (1999) *Principles of Distributed Database Systems*, Prentice Hall, 2nd Edition.
9.   Sheth, A.P., and J.A. Larson (1990) "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases," *ACM Computing Surveys*, 22(3), 183-236.
10.   Thomas, M.A., V.Y. Yoon, and R. Redmond (2007) "Extending Loosely Federated Information Systems Using Agent Technology," *International Journal of Intelligent Information Technologies* 3(3), 1-20.
11.   Thomas, M., R. Redmond, V. Yoon, and R. Singh (2005) "A Semantic Approach to Monitor Business Process Performance," *Communications of the ACM* 48, 55-59.
12.   Thomas, M.A., F.K. Andoh-Baidoo, and S. George (2005) "EVResponse – Moving Beyond Traditional Emergency Response Notification," AMCIS 2005 Proceedings.
13.   Visser, U., H. Stuckenschmidt, G. Schuster, and T. Vögele (2002) "Ontologies for Geographic Information Processing," *Computers and Geoscience* 28(1), 103–117.
14.   Wang, J., J. Pouwelse, R. Lagendijk and M.J.T. Reinders (2006) "Distributed Collaborative Filtering for Peer-to-Peer File Sharing Systems," in proceedings of the ACM Symposium on Applied Computing, 1026-1030.
15.   Weiss, A. (2007) "Computing In The Clouds", *netWorker* 11(4), 16-25.

NOTES