Association for Information Systems
AIS Electronic Library (AISeL)

AMCIS 2009 Proceedings | Americas Conference on Information Systems (AMCIS)

2009

From the Client-Server Architecture to the Information Service Architecture

Manoj A. Thomas
Virginia Commonwealth University, mthomas@vcu.edu

Richard T. Redmond
Virginia Commonwealth University, richard.redmond@vcu.edu

From the Client-Server Architecture to the Information Service Architecture

Manoj A Thomas
Virginia Commonwealth University
mthomas@vcu.edu

Richard T Redmond
Virginia Commonwealth University
richard.redmond@vcu.edu

ABSTRACT

This paper aims to justify the need for refining the concept of the traditional three-tier client-server architecture to address the drastic changes we have encountered in the form of information processing needs demanded by the public and the information processing services supplied to the public. The paper suggests much needed revisions to the traditional approach and demonstrates how the Information Service Architecture fits into the realm of future systems development by using a fairly complex example of an information system implementation.

Keywords

Client-server architecture, information service architecture, cloud computing, ontologies, knowledge engineering.

INTRODUCTION AND MOTIVATION

E-commerce and e-business have traditionally relied on *multitier* 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 has been challenged in two main ways. First, in the highly connected world that we live in, mobility has become the baseline of computing standards. Users are leaning more and more towards the freedom and flexibility of accessing personal and professional data without the inconvenience and security concerns surrounding difficult-to-carry 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 look and perform the same, irrespective of whether the user is at his 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 is no different whether it is a stand-alone computer based application or it is used on a mobile smart phone. Industry analysts peg the number of Internet connected personal computing system at 1.2 billion and growing rapidly (Markoff, 2008), this level of prolific growth accounting to almost 1 personal computer for every 6 people worldwide.

Besides RIA, two additional areas of computing are making stealthy inroads into the confines of every day computing needs of the ordinary man - Cloud Computing and Knowledge Representation. Cloud Computing provisions the average person access to massively huge computing power at negligible costs. Knowledge Representation, in a simple sense, embraces the use of descriptive web annotating languages to mimic social reality in a machine processable manner. Surprisingly, neither of the two can really be quantified as technological innovations of the twenty-first century. The seeds of conception for both were sown very early in academic and research settings, long before the early days of the Internet. Cloud Computing has been in existence in seemingly crude, yet relatively functional forms under various names such as distributed computing, clusters, and grid computing. Numerous commercial organizations and defense research installations have consistently allocated vast resource 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 of AI have been used in numerous real world applications (robotics, natural language processing, cybernetics, simulating organic functions, etc.) that stretch beyond figments of basic human imagination. AI mostly relies on complex algorithms involving Neural Networks, Decision Trees, Frame Systems and Experts Systems to match facts (data) against rule sets to infer conclusions that can then lead to actions. Knowledge Representation is the area of AI that concerns mainly with the codification of semantics (over syntax) of a chosen domain of interest (e.g. pharmacogenomics, cancer research or consumer behavior). Reasoners and Expert Systems can then use the knowledge structures as a basis to infer conclusions based on the way facts match the rules specified in the knowledge base.

As a sub-discipline of Knowledge Representation, *ontologies* have emerged as an effective means for expressing domain concepts, concept properties, relationship 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 exploitation to be more effective. The annotation language OWL (Web Ontology 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 (DL) to create expressive statements that describe the domain.

The technological innovation that promises the potential to make massive computing power available to the common man is the concept of Cloud Computing. As a term that is gaining quick prominence, Cloud Computing envisions the notion of delivering software services and customizable hardware configurations to public access, similar to how public utilities (electricity, water, high-speed internet connections, etc.) are made available to the common man. We suddenly see ourselves face to face with outlandish levels of computing resources that up until very recently was only available to the most privileged of the computer scientists working at heavily funded research institutes or elite military installations. The innovative entrepreneur is no longer faced with the dilemma of limited computing resources, instead lavished with the convenience and abundance of number crunching capabilities from massively distributed parallel clusters of computers (Google and IBM 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. It aims to justify the need for refining the concept of the traditional client-server architecture to address the drastic changes we have encountered in the form of information processing needs demanded by the public and the information processing services supplied to the public. The paper then suggests much needed revisions to the traditional approach. Finally, it demonstrates how the Information Service Architecture fits into the realm of future systems development by using a fairly complex example of an information system implementation.

The rest of the paper is structured as follows. We first highlight new innovations and technological advancements in the computing field that have resulted in a different form of user expectation and computing behavior. They are used as the basis to propose the next natural progression towards a revised architecture – the Information Service Architecture. We then describe an example scenario to validate the significance of the Information Service Architecture. Finally, the conclusion section stresses the importance of the Information Service Architecture and points out some of the limitations of current technology that needs to be addressed in future research.

ARCHITECTURE

There is little doubt that initiatives like the Cloud Computing Interoperability Forum (CCIF, 2009) promoted by industry stakeholders will only further the portability, integration and security specifications for the cloud based computing landscape. The concept of standalone computer applications has almost reached the end of life, except in a very few lingering instances (such as legacy applications maintained for the sake of providing support for vintage software systems, critical security systems with dedicated desktop-server links, or remote geographical locations where communication infrastructure creates a resource bottleneck). Recent developments in hardware and software services have brought about a renewed approach to the way we interact and use applications. It takes little to no effort in challenging the goodness of the classical three-tiers of the client-server models (let alone their more complex multitier derivatives {Oszu and Valduriez, 1999}) in this new era of perpetually connected systems, powerful computers, cheap storage and boundless network bandwidth. We are left with no choice but to rethink even the simplest organization of the tradition client-server model of systems design. The call for a pragmatic change in the architecture, wherein the concepts of cloud computing, knowledge representation and asynchronous data access are treated as integral components in the age of information services, is louder than ever before.

While the three-tiers of a typical client-server environment in general terms can be described as consisting of the data layer, the application/business logic layer and the presentation layer, there are legitimate reasons to re-organize the three-tiers into - the knowledge layer, the cloud layer and the presentation layer. The Information Service Model comprising of the three layers is shown in Figure 1.

The Cloud Layer

By providing platform independence, the cloud layer hosts the cloud computing services, which comprises of the application support and database services. Cloud computing may be defined (Ambrust 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.

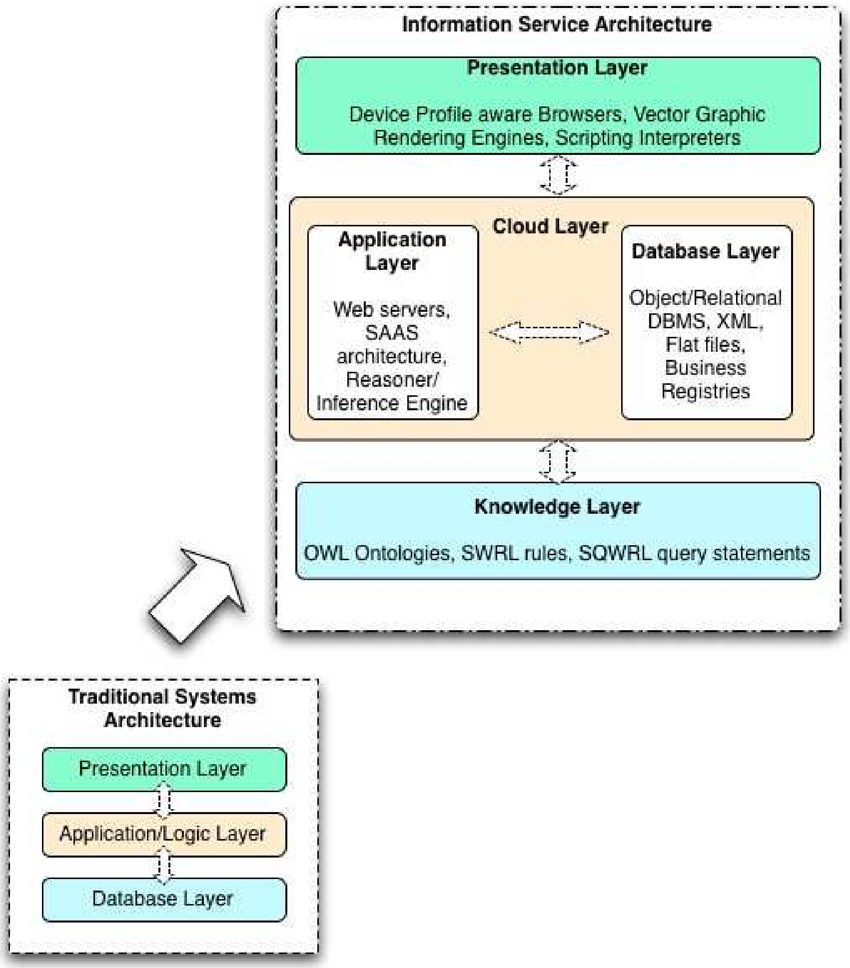

Figure 1. The Information Service Architecture

With the growing class of applications that are dynamically driven by customization based on personal preferences, contextualization and location awareness, persistent access to data structures become an integral part of the application design. With the advent of cloud computing, the applications in the second tier executing transactional logic with the database layer (third tier) are no longer residing or restricted to disparate territories, but stretched across the entirety of the all encompassing distributed cloud.

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 (design autonomy) independently from other shared systems in the cloud and the liberty to model the space with minimally stipulated restrictions (ownership autonomy) as the computing needs change. *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 resource. No cloud layer will be complete without a well-defined security manifest that upholds the notion of confidentiality, integrity and availability (Bishop, 2002) of data commensurate with any other organizational Information Systems Security model. Apple's (www.apple.com) 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 at usage rates instead of 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 of the popular Internet based companies (Google, Amazon, IBM and Microsoft) have made an entrance into the arena of cloud computing. It is not just merely these companies that benefit from the potential to generate revenue by exposing the otherwise idle cycles of their vast computing clusters at the various data centers. By extending the virtually unlimited computing resources that are geographically distributed, the opportunity presents itself for overall economic gains by allowing any entrepreneurial mind to implement solutions requiring large resource allocations, a major financial inhibitor that often limits the fruition of innovative ideas.

In short, the cloud layer dismisses 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 offering has proven viable and cost effective as evidenced by the commercial offerings such as the 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, which describe domain knowledge in a machine readable and human interpretable format. The term 'ontology' originates from philosophy, which means 'a systematic account of existence.' Studer, Benjamins and Fensel (1998) defines ontology as 'a formal, explicit specification of a shared conceptualization; *conceptualization* refers to an abstract model of some phenomenon in the world by having identified the relevant concepts of that phenomenon; *explicit* means that the type of concepts used, and the constraints on their use are explicitly defined; *formal* refers to the fact that the ontology should be machine-readable; *shared* reflects the notion that an ontology captures consensual knowledge, that is, it is not private of some individual, but accepted by a group.' Ontologies formalize concepts (as *classes*), relations among classes, constraints (as *axioms*) and instances (as *individuals*) in a domain of interest. Ontology Web Language (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, Stuckenschmidt, Schuster and Vogele, 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 academic research projects. It has resulted in the wide adoption of ontologies in surmounting number of commercial software.

As a layer that stores the generally used common semantic concepts, the knowledge layer serves shared descriptives for use by the upper layers. Domain knowledge may be stored in different formats at this layer. It may be a categorical tree that lists the most generic taxonomies as verticals (*e.g. Human, Software, Disease, etc.*) at the highest level, followed by specialized ontologies within the broader categories (e.g. *Human-dev-anat, Human-activities, etc.* under *Human*), or it could be a browsable collection of ontologies that are accessible via a distinct URI. Google base is a good example of the former, and the TONES Ontology Repository (http://owl.cs.manchester.ac.uk/repository/browser) is an example of the latter. Unlike the sandboxed nature of the cloud layer, the descriptive ontologies at this layer are openly shared and freely accessible to all member instances from any cloud. To perpetuate the concept of knowledge layer as an open ecology devoid of trademarks and patents, numerous checks and balances have to be in place to ensure quality and integrity of the enclosed domain descriptives. The concept quality can be ensured through peer assessment, relevance scores, popularity, and consumer preference (e.g., Google Base adopts this model). The integrity of concept structures will require knowledge engineers to adhere to a well-defined set of best practices and guidelines on ethics and professionalism.

As the demand for complex multi user applications (with unconventional capabilities like deductive reasoning, contextual guidance, logical inferencing from observed events and action triggers, etc.) increase, the knowledge layer will become a highly desirable underpinning necessary to provision the requisite semantic sharing and reasoning. During the software development process, the domain experts can browse the repositories for relevance and semantic richness and scrutinize ontologies for suitability and criteria match. Application developers can then load the identified URI accessible ontologies to their localized application logic (residing in the cloud layer) by solely relying on the ubiquitous hypertext transport. The asserted individuals (established facts) in the localized ontology are populated directly from the data layer (or through user inputs at the presentation layer) since any form of data may reside in the ontology. New information (inferred individuals generated by matching facts to rules) is articulated and classified in the ontology by reasoner engines residing in the cloud layer. Even though generalized *closure axioms* and *universal restrictions* will be implicitly stated in the common domain descriptive at the knowledge layer, goal specific rules for assertions and inferences may have to be added at the application logic. For instance, to accommodate the business objective of tracking special-promotions based sales of television units on a semantic web-based e-commerce store, a specialized SWRL rule may have to added at the application logic layer. During the software development phase, a SWRL rule

TV(?t) ∧ promotionSale(?t, "yes") ∧ itemInCart(AddToCart_Static, ?t) → PromotionGoal(?t)

may be added to the inherited version of the general consumer behavior ontology (*itemInCart* being an behavior concept in this ontology) and e-commerce ontology (*TV* being a product concept) sourced from the knowledge layer.

Presentation Layer

With the growing popularity of on-demand computing and I-Services, the consumer is relieved of the burden of software installation, hardware purchase and maintenance (Katzan, 2008). The latest technological triggers have also brought forth stricking facelifts at the presentation layer. More so than ever before, the web browser has turned into the primary tool for the end user to access the hosted services in the application/logic layer. This is in stark contrast to the popularity of fat-client desktop applications less than a decade ago. A hidden imperative to this growing dependence on browsers has been the innovative momentum to develop an add-on support environment that the browser can entrust to render and interpret 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, asynchronous offline capabilities are all examples of features available through the browser that were once only limited to desktop based client applications.

The need for the add-on browser support environment is mainly because traditional HTML falls short of functions that require complex information processing or intense image rendering. Without fail, implementing even the simplest form of a dynamic interactivity feature on a web page requires a certain reliance on scripting languages in addition to the presentation and formatting using hypertext languages. For the level of user interaction that is now expected at the presentation layer, this calls for robust yet lean support/rendering engine that can be invoked by the browser to service the lightweight, yet powerful web applications in a security-sandboxed 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 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 PCs to low powered small profile devices like smart phones and netbooks.

APPLICATION SCENARIOS

As a self-healing computing architecture comprising of thousands of computers spread thin but wide (Weiss 2007), the cloud layer in conjunction with the shared domain knowledge layer presents a tidy sleek model for even the most generic internet based computing tasks like retrieving search results from queries, contextual information sharing, knowledge exploration or even social collaboration on the web. One application scenario is described below that demonstrates the adaptation of the Information Service Architecture in the design, fabrication and organization of future web applications.

Comprehensive Emergency Response System

Crisis management requires the collaboration and coordination of different groups (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 invites broad repercussions. The Information Services Architecture dispenses 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 before the last point in time (an action threshold), after which the consequences of the problem are inescapable (Thomas, Andoh-Baidoo and George, 2005).

To effectively reach out to the common mass 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 (generating reports, sequencing event logs, predicting evacuation routes, simulating scenarios using clinical data points, etc.) for the crisis response teams. Unsolicited and solicited voice data is transposed (*Automated Speech Recognition* {ASR}) and translated (Text-To-Speech {TTS}) 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 in shared ontologies. The application hosted on the cloud layer uses ontologies available from the knowledge layer into which the static GIS datasets are loaded as *asserted individuals*. A reasoner engine hosted in a virtualized OS environment in the cloud classifies incidents (based on real time field inputs) and generates event response decision predicates as *inferred individuals* (new instances). Contextual crisis management information is reasoned by semantic matching with the various well-defined globally accessible resolution ontologies stored in the knowledge layer.

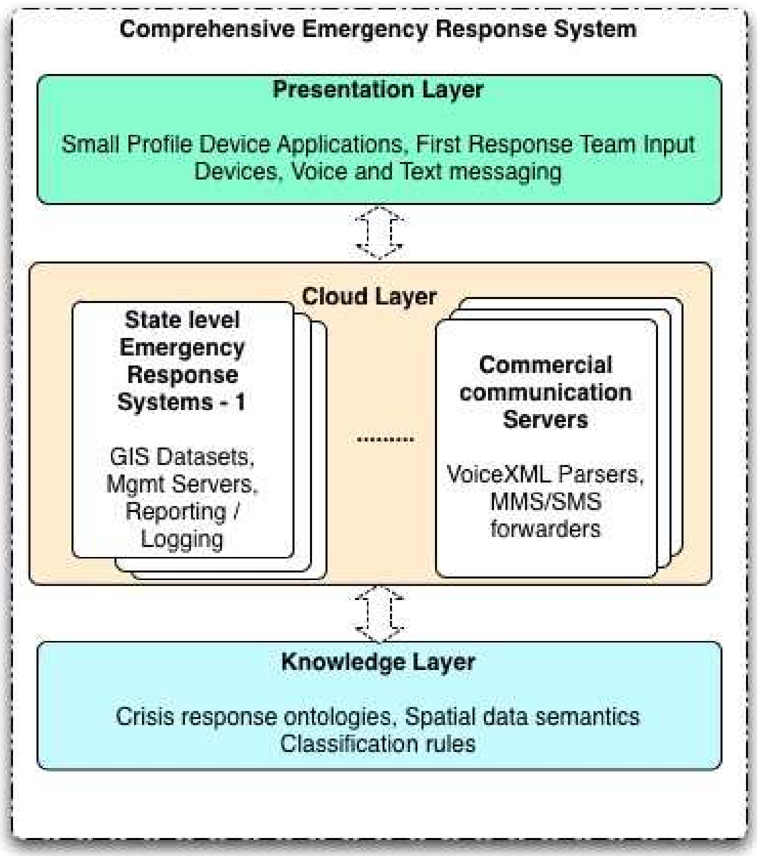

Figure 2. A Cloud Based Comprehensive Emergency Response System

As the software becomes a service in the cloud, the response teams gain the benefit of fast, 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 necessary rendering engines installed. GIS data sets, being graphically dense and visually rich, the presentation layer in this example will at a minimum require parsers for XML based UI language interpretation (e.g. Mozilla XUL, Microsft XBAP) along with scripting engines with built-in support for vectored graphics (e.g. Adobe Flex, Microsoft Silverlight). These are readily available free downloads from the commercial software vendors. A comprehensive emergency response system implemented using the Information Service Architecture is shown in Figure 2.

Conclusion

The Information Service Architecture represents the future of application processing and information representation systems. A consummate shift from standalone PC based applications to Internet based web applications is already in progress. Augment this further by the rapid popularity of low-cost computing devices (netbooks with low power processors and solid-state devices, as opposed to notebooks with higher processing power and storage capacity) and we are eye-to-eye with the new generation of computer savvy. Rich Internet Applications artfully communicate with these inexpensive wirelessly connected mobile computing platforms with a masterful awareness of the users' context and geographic location (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 application to the cloud. Microsoft recently released the *Azure* Services Platform that promises to make this vision into reality. OpenLazlo and Xcerion are other similar development initiatives with the goal of offering fully functional operating systems 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 their 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 only existed in the imagination of the human brain due to computational limitations will soon turn into reality. High definition video processing, localization tools for page translations, personalization algorithms relying on very elaborate rule sets, etc. that until now existed only in research settings become opportunistic 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, topping the list are confidentiality, security and legal concerns. The maturity, timeliness, cost effectiveness and benefits of flexibility associated with Service-On-Demand (Software As A Service, SAAS {Ambrust et al. 2009}) models and Elastic Server systems that can be provisioned and deployed in matter of minutes is sure to question the benefits of in-house self-maintained back-office operations. This further fortifies the emerging congruence of shared knowledge representations and elastic resource management under the unified umbrella of the Information Service Architecture.

The example of the Comprehensive Emergency Response System presented in this paper is a proof of concept used to demonstrate the validity of the Information Service architecture in modeling complex Information Systems. Although cloud computing, ontologies and Rich Internet Applications exists in simpler forms today, it will be a worthwhile effort to implement the concepts as simplistic forms of loosely coupled distributed semantic web applications.

Cloud computing is not fully clear 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. Amburst et al. (2009) lists 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 Service-On-Demand providers and those planning to enter this lucrative market arena of cloud computing.

REFERENCES

1. Ambrust, M., Fox, A., Griffith, R., Joseph, A., Katz, R., Konwinski, A., Lee, G., Patterson, D., Rabkin, A., Stoica, I. and Zaharia, M. (2009), "Above the Clouds: A Berkley View of Cloud Computing", Electrical Engineering and Computer Science Technical Report 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., Kutsche, R., Leser, U. & Weber, H., (1999) Federated Information Systems: Concepts, Terminology and Architectures available (18 Feb 2009) 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).
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, 25th February 2008.
8. Ozsu, T. and Valduriez, P., (1999) "Principles of Distributed Database Systems", Prentice Hall, 2nd Ed.

9. Sheth, A, P. and Larson, J, A. (1990) "Federated Database Systems for Managing Distributed, Heterogeneous, and Autonomous Databases", *ACM Computing Surveys*, 22(3), 183-236.
10. Studer, R., V.R. Benjamins and Fensel, D. (1998) "Knowledge Engineering: Principles and Methods," Data and Knowledge Engineering, 25, 161-197.
11. Thomas, M., Redmond, R., Yoon, V. & Singh, R., (2005), "A Semantic Approach to monitor Business Process Performance", Communications of the ACM, 48, 55-59.
12. Thomas, M., Andoh-Baidoo, F., and George, S., (2005) "GeoResponse – Moving Beyond Traditional Emergency Response Notification", The 11th Americas Conference on Information Systems.
13. Visser, U., H. Stuckenschmidt, G. Schuster, and T. Vogele (2002), "Ontologies for geographic information processing," Computers and Geoscience, 28, 103–117.
14. Wang, J., J. Pouwelse, R.L. Lagendijk and M.J.T.Reinders (2006) "Distributed Collaborative Filtering for Peer-to-Peer File Sharing Systems," in Proceedings of the 2006 ACM Symposium on Applied Computing, 1026-1030.
15. Weiss, A. (2007) "Computing In The Clouds", ACM. 16-25.