# **CLOUD COMPUTING MESH MODEL FOR DECISION SUPPORT SYSTEMS**

Harry Katzan, Jr. Savannah State University

## ABSTRACT

This paper introduces the use of service science to enhance decision-making capability by providing service transparency to information from diverse knowledge sources in a complex operating environment. The concept of a *live collector* that runs in the cloud is developed. The collector gives information visibility, context mediation, and presentation facilities on a dynamic basis by employing the principles of cloud computing. The subject of cloud computing is introduced in the paper, and a case study from decision support is presented.

## INTRODUCTION

Much of modern business is dependent upon information that is locked in on-premises information systems. Clearly, the price for maintaining and accessing this information is high and incorporates infrastructure, hardware, software, development, and professional services including the in-house costs of utilization and deployment. Information systems of this sort have evolved such that the aforementioned information can be encapsulated, exposed as components, and accessible as services over the Internet. In many cases, the components are usable in systems development for the development of other on-premises software applications. [Kat08b, pp. 119ff] The concept is well established and known as *Service-Oriented Architecture* (SOA). [Erl08] The drawback of most, but certainly not all, SOA applications is that the up-front costs are high, time consuming, and yield only a marginal contribution to business agility. Many business operations require that information from several on-premises systems be visible for consistent and effective decision-making. The provisioning of information from several systems to facilitate decision making is the subject of this paper.

## **CLOUD COMPUTING CONCEPTS**

*Cloud computing* is a collection of technologies for accessing computer facilities via the Internet. From a user perspective, ubiquitous access is provided to software and information through the use of a web browser. The adjective "cloud" reflects the diagrammatic use of a cloud as a metaphor for the Internet and suggests that the same services are available in several forms, such as conventional web access, mobile platforms, and mesh computing.

From a provider perspective, cloud computing refers to software deployed as a hosted service and accessed over the Internet. There are two aspects to service provisioning: infrastructure services and application services. Both use a *cloud platform*, which is essentially an operating system that runs in the cloud. [Cha08a, Cha08b] *Infrastructure services* include authorization/authentication/security facilities, integration between infrastructure and application services, and online storage facilities. *Application services* refer to ordinary business services that expose "functional" services as SOA components. Cloud platforms are a lot like enterprise-level platforms, except that they are designed to scale up to Internet-level operations supporting millions of clients.

Cloud computing clients are grouped into three categories: traditional business users, independent software vendors (ISVs), and consumers. *Traditional business users* run their applications in the cloud to avoid up-front deployment costs and to minimize development and deployment time. *ISVs* develop multi-

tenant applications that can address a wider marketplace for services and provide economy of scale for the client. *Cloud consumer services* are typically free to the client, as well as being accessible from any location via the Internet, and yield advertising-supported revenue for the provider. Consumer services have a near-zero marginal cost of distribution to clients.

## **APPLICATION ARCHITECTURE**

For the purposes of this paper, two forms of cloud application architecture are established: the multitenant model and the visibility model. The *multi-tenant model* runs on a cloud platform and uses a common data model for all tenants. This model provides economy of scale for clients based on long tail monetization. [Kat08b] This model, popular with ISVs, probably will use the virtualization feature of cloud operating systems to achieve multi-tenancy, and requires that all clients run the same basic business application. [Kat86, Cha08b]

The *visibility model* reflects proprietary software that runs on a cloud platform. It uses a single tenant model and demands a separate data model. Each tenant may have several users, each associated with the same client. Internet access is required for on-premises and other cloud applications. This model may involve the use of specially written computer programs to support particular business applications, such as decision support.

## SYSTEM DESIGN

Achieving service visibility from different information systems requires a software application that is hosted on a cloud platform and adopts the visibility model. We are going to call this application the *live collector* that creates an operational scenario on a dynamic basis. The collector is script-driven and creates an instance for each client. It is important that the application runs as a client, where the client can sustain several users. An application instance is synthesized for each user. Figure 1 gives a structural and process design diagram of the live collector with functional sub-components, as required. The main components are the script-driven instance generator, the accesser to on-premises data objects, the aggregator that derives a composite picture of the application domain, the presenter that derives the requisite presentation facility, and a storage manager for handling the data-centric aspects of cloud computing. The systems environment is conceptualized in Figure 2. The design is actually a *mesh model* intended to connect a user's PC, mobile device, and other cloud services via the Internet. The mesh software functions as a control hub providing unified services on a demand basis.

## **Instance Generator**

The *live collector* is composed of software components that can be assembled dynamically to instantiate a specific application instance. The *instance generator* is a script-driven functionality that generates an operational instance. This operational instance, when executed, will access components from on-premises information systems with the accesser component and then synthesize a composite data structure, relying on the context mediator when needed. The composite data structure will feed the presenter component that generates and updates the display. The aggregator will employ a pulsar module that can dynamically refresh the display from on-premises components. The storage manager, also script-driven, will store operational information in a cloud database. Since the instance runs in the cloud, it can be accessed from any location and with fixed and mobile facilities.

## Accesser

The primary function of the *accesser* component is to locate needed information on on-premises information systems. To achieve data location, the accesser will employ Web Services technology, which includes a registry of system components. [Cer02] The locater looks up needed components in the

registry and returns their Internet-accessible reference locations. The accesser can return that information to the aggregator for subsequent composition and display,

## Aggregator

The *aggregator* is a software component that resides in the live collector with the sole objective of combining information as prescribed in the script. The aggregator is conceptualized as a rule-based software component that would necessarily employ the services of a "context mediator" to resolve semantic differences in otherwise incompatible entities, as described in the following quotation. "Context mediation is a field of research that is concerned with the interchange of information across different environments, which provides a vehicle to bridge semantic gaps among disparate entities." [Buc04] To insure the timeliness of information, the aggregator should exercise a pulsar component that would refresh information on a periodic basis.



Figure1. Design of the Live Collector.



Figure 2. Conceptualization of the Live Collector.

## Presenter

The *presenter* is an open-ended display component – sometimes regarded as a "dashboard" – designed to match informational requirements with display technology. Presenter output is expected to be archived for subsequent reference and to protect against repudiation.

#### **Cloud Storage Manager**

The *cloud storage manager* is a front-end to a cloud database component designed to handle cases of multi-tenant data structures with multiple users. A key issue with cloud storage management is "Who owns the data?" It is generally felt that ownership considerations can be resolved with effective service-level agreements.

### Workflow

As implied in Figure 1, the workflow for service visibility is the following: instance generation, access, synthesis, presentation, and storage. Since instance execution is script driven, the flow can be adjusted on a dynamic basis, perhaps with the use of intelligent agents. [Rus03]

## **BUSINESS PERSPECTIVE**

It would seem that there is something in cloud computing for everyone. For the IT manager, it means reduced upfront costs for hardware and software. For the business manager, it means shorter development time and increased business agility. For the business analyst, it means the ability to effectively work from a remote location. For the executive, it means obtaining information on a client, business partner, or competitor from diverse systems and then combining that information in a form useful for decision making, such that the information is timely, accurate, and complete. Often, an enterprise will have several lines of business with clients that address one or more of those lines. In these cases, it is more than useful for an executive to have a composite position for decisions that affect that client.

In the next section, we give a case study involving data from different knowledge sources that are combined to form a composite picture of a situation. It demonstrates that the cloud is a practical technology for quantitative as well as qualitative data.

#### **CASE STUDY FROM CONSENSUS THEORY**

The effectiveness of unstructured decisions made under uncertainty directly involves two important concepts: the representation of the problem domain and the completeness of the solution space. A *category* is a means of representing the problem domain so that relevant structural information may be assessed and decisions can be made. A *frame of discernment* ([Sha76], [Kat92], [Kat06], [Kat08c]) is a set of mutually exclusive and collectively exhaustive possibilities for the solution space.

## Category

A category is a means of structuring a problem domain with objective of engaging in a predictive modality in which one or more future events may be identified and analyzed. Let  $C_i$  be one of the categories used to stratify the problem domain such that the collection

$$C = \{C_1, C_2, ..., C_n\}$$

represents a complete conceptualization of the dynamics under investigation and *n* is the number of categories.

Associated with each category is a set of probabilities representing an assessment of a future outcome based on its underlying categorical imperative. Thus, a category is a mechanism for isolating a single view of the problem under consideration. The ontological definition of a *category*, as a conceptual entity with no attributes in common with other categories, is adopted in this paper.

## Frame of Discernment

A frame of discernment is a means of representing the possibilities under consideration, as in the following examples:

## E = {McCain, Obama} M = {Up, Unchanged, Down}

Clearly, the elements in a frame of discernment are, in fact, propositions that can be interpreted as events or states. Thus, if component  $s_i$  of system S over domain V were associated with the symbol "**McCain**," then that state is equivalent to the proposition, "The true value of V for component  $s_i$  is **McCain**," or in ordinary language, " $s_i$  prefers **McCain**."

Accordingly, the set S of propositions  $S_i$ ,

 $S = \{S_1, S_2, ..., S_n\}$ 

commonly represent the collection of states of a system under analysis. Clearly, at an agreed upon point in time, one proposition is true and the others are false.

#### Uncertainty

Prior to the agreed point in time ( $\tau$ ), we obviously do not know the state of the system under analysis or its components with any degree of certainty. The expectation that a part of the system will be in a particular state at time  $\tau$  is denoted by a real number  $p(S_i)$  associated with each of the propositions in the frame  $S = \{S_i\}, i=1,2,...,n$ , such that

 $0 \le p(S_i) \le 1$ 

and

$$\sum_{i=1}^{n} p(S_i) = 1$$

This is simply the addition rule for mutually exclusive events.

#### **Consensus Theory**

*Consensus theory* is a methodology for combining evidence based on Dempster-Shafer theory ([Sha76], [Kat92], [Kat06], [Kat08c]) and the mathematical combination of evidence ([Dem67]). Dempster-Shafer theory has commanded a considerable amount of attention in the scientific and business communities, because it allows a knowledge source to assign a numerical measure to a proposition from a problem space, and provides a method for the measures accorded to independent knowledge sources to be combined. Dempster-Shafer theory is attractive because conflicting, as well as confirmatory, evidence from multiple sources may be assimilated.

The basis of Dempster-Shafer theory is the frame of discernment  $(\Theta)$ , introduced previously. Accordingly, a knowledge source may assign a numerical measure to a distinct element of  $\Theta$ , which is equivalent to assigning a measure of belief to the corresponding proposition. In most cases, the numerical measure will be a basic probability assignment. A measure of belief may also be assigned to a subset of  $\Theta$  or to  $\Theta$  itself.

#### **Support Functions**

Consider a frame of discernment  $\Theta$  and its power set denoted by  $2^{\Theta}$ . For example, given the frame:

 $\Theta = \{a, b, c\}$ 

The power set is delineated as follows:

 $\begin{array}{rl} 2^{\Theta}=& \{\{a,b,c\},\\&\{a,b\},\{a,c\},\{b,c\},\\&\{a\},\{b\},\{c\}\} \end{array}$ 

In Dempster-Shafer theory, a knowledge source apportions a unit of belief to an element of  $2^{\Theta}$ . This belief can be regarded as a mass committed to a proposition and represents a judgment as to the strength of the evidence supporting that proposition. When viewed in this manner, evidence focuses on the set corresponding to a proposition; this set is called a *focal set*.

The support for a focal set is a function *m* that maps an element of  $2^{\Theta}$ , denoted by *A*, onto the interval [0,1]. Given a frame of discernment  $\Theta$  and function *m*:  $2^{\Theta} \rightarrow [0,1]$ , a support function is defined as follows:

 $m(\phi) = 0$ , where  $\phi$  is the null set  $0 \le m(A) \le 1$ , and  $\sum_{A \subset 2^{\Theta}} m(A) = 1$ 

The support function m is called a *basic probability assignment*, which is assigned by the knowledge engineer or domain specialist.

A support function is called a *simple support function* if it reflects, at most, one focal set not equal to  $\Theta$ . A simple support function assigns a measure of belief to the focal set *A*, as follows:

m(A)>0  $m(\Theta)=1-m(A)$ m(B)=0, for all  $B \subset 2^{\Theta}$  and  $B \neq A$ 

The simple support function for a focal set *A* assigns a portion of the total belief exactly to *A* and not to its subsets or supersets. The remainder of the belief is assigned to  $\Theta$ , and this is where Shafer's work departs from conventional Bayesian analysis. Because certainty function must add up to 1,  $m(\Theta)=1-m(A)$ .

It is possible that a body of knowledge or evidence supports more than one proposition, as in the following case. If

$$\Theta = \{a, b, c, d\}$$
$$A = \{a, b\}$$

and

$$B = \{a, c, d\}$$

then the evidence supports two focal sets, which in the example, are A and B. If m(A)=0.5 and m(B)=0.3, then  $m(\Theta)=0.2$ . A support function with more than one focal set is called a *separable support function*. Separable support functions are normally generated when simple support functions are combined.

The notion of combining simple support functions is a practical approach to the assessment of evidence. An analyst obtains information from a knowledge source, and it leads to an immediate conclusion – not with certainty, but with a certain level of belief. This is a normal straightforward means of handling human affairs and is precisely what people do. Then when additional information comes in, the prior evidence and the new information are combined to obtain a composite picture of the situation.

## **Combination of Evidence**

A method of combining evidence is known as Dempster's rule of combination ([Dem67]). Evidence would normally be combined when it is obtained from two different observations, each over the same frame of discernment. The combination rule computes a new support function reflecting the consensus of the combined evidence.

If  $m_1$  and  $m_2$  denote two support functions, then their combination is denoted by  $m_1 \oplus m_2$  and is called their *orthogonal sum*. The combination  $m_1 \oplus m_2$  is computed from  $m_1$  and  $m_2$  by considering all products of the form  $m_1(X) \cdot m_2(Y)$ , where X and Y range over the elements of  $\Theta$ ;  $m_1(X) \cdot m_2(Y)$  is the set intersection of X and Y combined with the product of the corresponding probabilities.

For example, consider the frame of discernment

 $\Theta = \{\text{healthy, tests, sick}\}$ 

and views A and B, based on two different observation over the same frame:

 $X = \{\{\text{healthy}\}, 0.6\}, \{\{\text{tests}\}, 0.3\}, \{\{\text{sick}\}, 0.1\}\} \\ Y = \{\{\text{healthy}\}, 0.4\}, \{\{\text{tests}\}, 0.4\}, \{\{\text{sick}\}, 0.2\}\}$ 

The entries are combined as follows using Dempster's rule of combination:

 $m_1 \oplus m_2(\{\text{healthy}\}) = 0.24$   $m_1 \oplus m_2(\{\text{tests}\}) = 0.12$   $m_1 \oplus m_2(\{\text{sick}\}) = 0.02$  $m_1 \oplus m_2(\{\emptyset\}) = 0.62$ 

Thus, for  $A_i \cap B_i = A$  and  $m_1 \oplus m_2 = m$ , the combination rule is defined mathematically as:

$$\boldsymbol{m}(\boldsymbol{A}) = \sum_{A_i \cap B_j = \boldsymbol{A}} \frac{m_1(A_i) \cdot m_2(B_j)}{A_i \cap B_j = \boldsymbol{\emptyset}} \frac{m_1(A_i) \cdot m_2(B_j)}{A_i \cap B_j = \boldsymbol{\emptyset}}$$

The denominator reflects a normalization process to insure that the pooled values sum to 1. So, in this instance, the normalization process yields the combination

$$X \oplus Y = \{\{\text{healthy}\}, 0.63\}, \{\{\text{tests}\}, 0.32\}, \{\{\text{sick}\}, 0.05\}\}$$

after normalization by dividing the combined assessment by (1-0.62) or 0.38. Because the problem is wellstructured, the representation can be simplified as

$$X \oplus Y = \{0.63, 0.32, 0.05\}$$

For views A={ $A_1, A_2, ..., A_n$ } and B={ $B_1, B_2, ..., B_n$ }, the combination rule can be simplified as

$$A \oplus B = \{A_1 \times B_1 / k, A_2 \times B_2 / k, \dots, A_n \times B_n / k\}$$
where  $k = \sum_{i=1}^n A_i \times B_i$ 
[1]

We will refer to equation [1] as the *simplification rule*.

An example of the preceding concepts is demonstrated trough the elicitation of expert opinion.

## **Elicitation of Expert Opinion**

Typically, experts do not agree, especially when system failure is concerned. Typical examples might be the crash of an expensive fighter aircraft or the collapse of a building. Consider a situation wherein the frame of discernment is  $\{A,B,C\}$  reflecting that the failure could be caused by Component A, Component B, or Component C. Expert #1 believes the failure is due to Component A with probability 0.75, Component B with probability 0.15, or

Component C with probability 0.10. Expert #2 believes the failure is due to Component A with probability 0.30, Component B with probability 0.20, or Component C with probability 0.50. The support function are:

## Expert #1 = {{{A},0.75}, {{B},0.15}, {{C},0.10}} = {0.75, 0.15, 0.10} Expert #2 = {{{A},0.30}, {{B},0.20}, {{C},0.50}} = {0.30, 0.20, 0.50}

Table 1 summarizes the application of the simplification rule to this problem. The opinion of the experts is summarized and reflects the differing opinions.

Support Function	Probability Assignment	Entropy
Expert #1 (=X)	$\{0.75, 0.15, 0.10\}\$	1.05
Expert #2 (=Y)	$\{0.30, 0.20, 0.50\}\$	1.49
X×Y	$\{0.738, 0.098, 0.164\}$	1.08

Table 1. Elicitation of Expert Opinion.

The strong opinion of Expert #1 in favor of Component A, reflected in the low entropy, has a major influence on the consensus.<sup>1</sup>

## SUMMARY

The basis for service visibility across differing domains is service science, cloud computing, Web Services, and Service-Oriented Architecture. Cloud computing is a collection of technologies for accessing computer facilities via the Internet. The two aspects of cloud service provisioning are infrastructure services and application services. Cloud computing clients are grouped into three categories: traditional business users, independent software vendors (ISVs), and consumers. The two forms of cloud application architecture are the multi-tenant model and the visibility model. A live collector is a software application used to achieve service visibility across lines of business. The main components of a live collector are the instance generator, the accesser, the aggregator, the presenter, and the storage manager. The collector through mesh technology is useful for combining information from disparate knowledge sources.

#### REFERENCES

- [Buc04] Buchner, A., *Context Mediation among Knowledge Discovery Components*, www.dissertation.com, (from the abstract), (2004).
- [Cer02] Cerami, E., Web Services Essentials, Sebastopol, CA: O'Reilly Media, Inc., (2002).
- [Cha08a] Chappell, D., "A Short Introduction to Cloud Platforms," Microsoft Corporation, (August 2008).
- [Cha08b] Chappell, D., "Introducing the Azure Services Platform," Microsoft Corporation, (October 2008).
- [Dem67] Dempster, A.P., Upper and Lower Probabilities Induced by a Multivalued Mapping, *The Annals of Statistics* 28:325-339, (1967).
- [Erl08] Erl, T., SOA: Principles of Service Design, Upper Saddle River, NJ: Prentice-Hall Inc., (2008).
- [Kat86] Katzan, H., Operating Systems: A Pragmatic Approach (2e). Hoboken, NJ: John Wiley and Sons, 1986.

<sup>1</sup> Entropy for n elements is computed as:

$$H(x) = \sum_{i=1}^{n} x_i h(x_i) = \sum_{i=1}^{n} x_i \log(1/x_i) = -\sum_{i=1}^{n} x_i \log(x_i)$$

- [Kat92] Katzan, H., *Managing Uncertainty: A Pragmatic Approach*, New York: Van Nostrand Reinhold Co., (1992).
- [Kat06] Katzan, H., Consensus, *Proceedings of the Decision Science Institute National Conference*, San Antonio TX. (November 2006).
- [Kat08a] Katzan, H., Cloud Computing, I-Service, and IT Service Provisioning, *Journal of Service Science*, Volume 1, Number 2, (2008).
- [Kat08b] Katzan, H., Foundations of Service Science: A Pragmatic Approach, New York: iUniverse, Inc., (2008).
- [Kat08c] Katzan, H., Categorical Analytics Based on Consensus Theory, *Journal of Business and Economics Research*, Volume 6, Number 8, (2008).
- [Kno08] Knorr, E and G. Gruman, "What cloud computing really means," *InfoWorld*, April 07, 2008, <u>www.infoworld.com</u>, (2008).
- [Mar08] Martin, R and J. Hoover, "Guide to Cloud Computing," *Information Week*, June 21, (2008), <u>www.informationweek.com</u>, (2008).
- [Mil08] Miller, M., Cloud Computing: Web-Based Applications That Change the Way You Work and Collaborate Online, Indianapolis: Que Publishing, (2008).
- [Rus03] Russell, S. and P. Norvig, *Artificial Intelligence: A Modern Approach* (2ed), Upper Saddle River, NJ: Prentice-Hall Inc, (2003).
- [Sha76] Shafer, G., A Mathematical Theory of Evidence, Princeton, NJ: Princeton University Press, (1976).
- [The67] Theil, H., *Economics and Information Theory*, New York: American Elsevier Publishing Company, Inc., (1967).