

Mapping the Cybernetic Principles of Viable System Model to Enterprise Service Bus

Nizami Jafarov and Edward Lewis

School of Engineering and Information Technology

University of New South Wales at Australian Defence Force Academy

n.jafarov@adfa.edu.au, e.lewis@adfa.edu.au

Abstract—This paper describes the results of a theoretical mapping of the cybernetic principles of the Viable System Model (VSM) to an Enterprise Service Bus (ESB) model, with the aim to identify the management principles for the integration of services at all levels in the enterprise. This enrichment directly contributes to the viability of service-oriented systems and the justification of Business/IT alignment within enterprise. The model was identified to be suitable for further adaption in the industrial setting planned within Australian governmental departments.

Keywords—cybernetics; enterprise service bus; service oriented architecture; viable system model

I. INTRODUCTION - THE NEED FOR ACTION

Nowadays, contemporary enterprises aim to transform their monolithic Information Technology (IT) infrastructures into more agile and flexible software systems in order to improve the aspects of management of enterprise services in the creation and capture of service values [1]. Essentially, enterprise services fall into four categories (Fig. 1), commonly outlined according to Enterprise Architectures (EA) used in organisations, each operating within a defined layer [2] that includes:

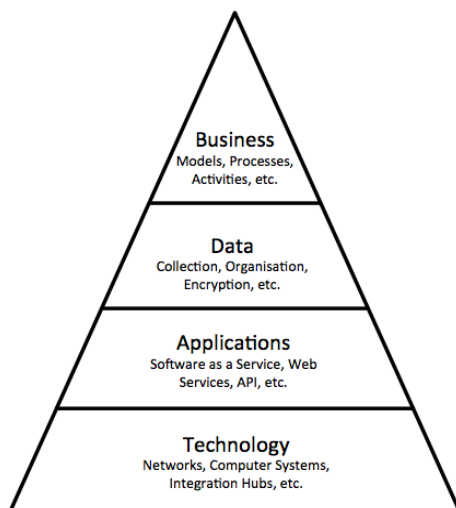


Fig. 1. Four types of services in enterprise.

1. Business – is the most abstract layer that includes representations of services that are present within organisation in one form or another.
2. Data – is the layer that includes various data manipulation services, both open-source and proprietary.
3. Applications – is the implementation layer that exposes IT functionality in the form of services using Web Services, Application Programming Interfaces (API)s and similar technologies.
4. Technology – is the layer that includes services, which are the means for interconnecting IT assets to form an infrastructure, so they can be utilised cooperatively.

In enterprise context, these services are in continuous interaction with one another at various levels of service abstraction. This interaction, vertically up the services pyramid (alignment) or horizontally between services of the same level, heavily influences the management of services within organisations, as it usually goes beyond technical relationships between system elements, to socio-technical settings that include human-in-the-loop.

To manage IT services, organisations adapt various methodologies, which include diverse range of Information Technology Service Management (ITSM) frameworks. Generally, ITSM focus goes beyond technology and internal aspects of organisation, to the management of quality of services provided to end-customers and the relationship with them [3].

To transform monolithic IT systems, organisations adapt the Service Oriented Architecture (SOA), which is a contemporary paradigm that has a potential to propagate granularity to complex business processes within enterprise setting by adapting the principles of service orientation [4]. These principles outline the roadmap for the decomposition of the processes into loosely coupled services [5].

SOA is a systems design approach that emerged as a response to ever-accelerating changes in business processes,

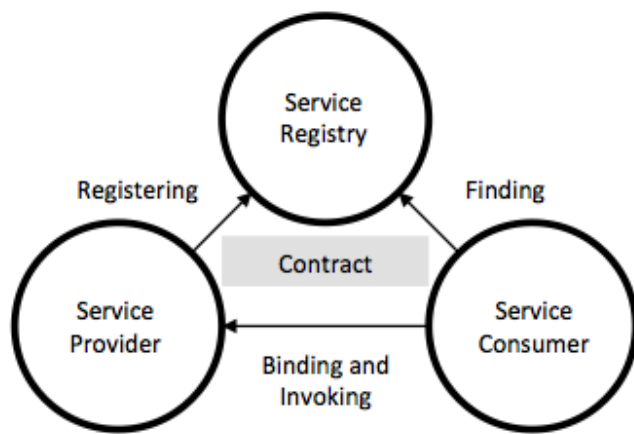


Fig. 2. Basic SOA Scenario.

products and services [6]. Identified by some organisations as the most important architectural paradigm in IT [7], SOA outlines the design principles, patterns and techniques [8] that address requirements for such system characteristics as the increase of service quality, open standards, service abstraction layers, discoverability, federation, extensibility, composability, loose coupling, location transparency, organisational agility, scalability, modularity, reusability, flexibility and interoperability, amongst others [9]. These characteristics enable the system to adapt rapidly to changes in either the business environment or the technology that makes up the infrastructure supporting the business.

In a common SOA environment two actors are playing the main roles: service providers and service consumers. The interaction often involves an intermediate registry or directory that is used by both parties: service providers register to the directory, so that consumers can discover them. Service providers are also endowed with service contracts that outline their functionality and connectivity options for the potential service consumers, as depicted in the figure below (Fig. 2).

However, along with SOA that outlines the principles of service orientation, another important component that enables some or most of these principles has also emerged. This component is Enterprise Service Bus (ESB).

ESB, often coined as a backbone of SOA [10], is the critical component of SOA infrastructure [11] that provides essential communication and integration facilities required to implement service-oriented systems [12]. ESB arose as one of the important SOA compound design patterns [13] and can be described as a highly distributed, message-oriented middleware integration engine that is built upon open standards to provide routing, invocation and mediation mechanisms to maximize quality of service interaction and transaction management between pervasive services in a secure and reliable manner [14]. Its capabilities in message handling

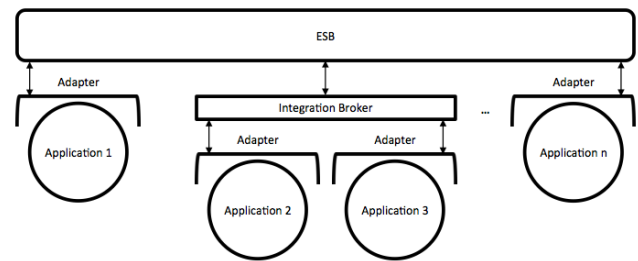


Fig. 3. Basic ESB Scenario.

also enable the design of systems that can be readily integrated with each other, so they can also adapt to rapid changes in supply or demand [15]. The use of an ESB is a response to the risks of ineffectual and inefficient enterprises that arise from such changes.

In a common ESB-enabled environment, applications are integrated through a variety of adaptors and integrations brokers that expose applications functionality in the form of service end-points. These endpoints interact with each other in an asynchronous manner, by utilising the reliable message delivery and other services of ESB that acts as an intermediate layer between the participants, as depicted in the figure below (Fig. 3).

ESB implementation is considered to be one of the core milestones in the realization of SOA in organisations [16]. Nowadays, the market is overblown by numerous varieties of ESB products with quite distinct sets of features that complicate and delay the overall adaption of ESBs in enterprise environment [17]. The central challenge for the adaption of ESB is the absence of consensus on the common model of ESB: although, professionals in both industry and academia propose sets of features ESB shall be endowed with [15], [18]–[21], there are no principles that outline the standardized approach that would combine these features into a justified holistic whole, nor is there an approach that would discriminate between the overlapping features or those that do not contribute to the integrity of the system from the ones that do. Moreover, interpretation of the existing principles varies from vendor to vendor, which complicates the management of the necessary integration of enterprise services.

To overcome such issues, the Viable Enterprise Service Bus Model (VESBM) is proposed. VESBM is created from the symbiosis of ESB and the Viable System Model (VSM) to provide a roadmap for the generation of a vendor-neutral blueprint for the management of distinct middleware infrastructures that drive the integration of the services that form the IT infrastructure. This paper describes the results from the theoretical mapping of the cybernetic principles of VSM to ESB as one of the initial steps towards creation of the model. The steps are described in greater detail in the following

sections: Section 2 gives the need for this method; Section 3 defines the model of VESBM; Section 4 provides the results of the mapping of VSM to ESB; Section 5 explains how the VESBM is planned to be tested in an Australian governmental department; and Section 6 concludes the paper with the current and future research directions.

II. MOTIVATION - THE NEED FOR METHOD

In the era of Cloud, holism in the ESB is important for the viability of service-oriented systems as they undergo significant design changes through various evolution stages. This viability is especially important in times of technological disruptions, which might lead to possible disintegrations between abstract business models and their actual service representations, so affecting the overall Business/IT alignment within the enterprise. The impact of disruptive technologies on business, attention to which was initially brought by Bower and Christensen [22], is one of the most debated topics of the last decade. It is also the most sensitive topic for business [23], especially in the scope of Cloud Computing that not only provides cheap and virtually limitless processing power and storage [24], but also disrupts the alignment of technology with business [25].

Cloud Computing is closely correlated with Service-oriented Computing [26] and SOA is one of its essential technical foundations [27]. According to the National Institute of Standards and Technology, the Cloud Computing has three service models, which are the Software as a Service (SaaS), Platform as a Service (PaaS) and Infrastructure as a Service (IaaS) [28]. If applied to the common services of enterprise, described and illustrated earlier (Fig. 1), each service layer, except Business, will fall into a particular Cloud Computing service model, such as:

1. The IaaS model – will include the Technology layer, as it is comprised of virtual machines, load balancers, network infrastructures, etc.
2. The PaaS model – will include the Data layer, as it is comprised of databases, web servers, development tools, etc.
3. The SaaS model – will include the Applications layer, as it is comprised of email platforms, Customer Relationship Management (CRM) platforms, Enterprise Resource Planning (ERP) platforms, etc.

The Business layer, along with other possible services of enterprise, will fit into the so called Everything as a Service (EaaS/XaaS) model [29] that is associated with each of the IaaS, PaaS and SaaS models separately, but also includes business models, processes, activities and so on. Perhaps the move towards Business Process Outsourcing is as example of the extension of the XaaS model.

Cloud Computing led to the development of new computing paradigms, amongst which are Hybrid Computing of Grid and Cloud [30], High Performance Computing (HPC) and Virtualisation in the Cloud [31] and most prominently the popularised, during the last years, use of Application Programming Interfaces (API) to connect disparate mobile platforms through the Cloud by utilisation of external resources, which empower applications on smartphones and overcome hardware limitations by partially off-loading their execution into the Cloud [32], to mention a few [33].

The newest case of using APIs was mainly driven by technological advancements in smartphones, tablets as well as social networks and social commerce [34]. It became a basis for fundamentally new business models and a new type of economy, known as the API Economy [35]. The impact of API on global IT market is already visible [34]: 77% of the current top 50 freeware and top 50 shareware applications are connected to backend services, whereas 23% are standalone; APIs are also launched as the core delivery strategy in most organisations and out of 1000 APIs in global directory 11% are oriented on mobile platforms, 38% are oriented on non-mobile platforms and the remaining 51% are shared between the both; by 2016 the number of public APIs will raise from the current 8,826 to 30,000 and according to the survey conducted by Layer7 86.5% of organisations will have an API program in place within 5 years [36].

Currently, the existing research in the domain of service management is extensively focused on the management service compositions and various service models, rather than the management of the middleware infrastructures, such as the ESB that integrate those services. The importance of infrastructure management is increasing, especially in the era of Cloud, as was highlighted by Golden [37]: “... *failure to adopt the infrastructure management practices pioneered by the large public providers will consign an operations group to slow, reactive and expensive processes, which are unacceptable in a world that has embraced automation as a quotidian practice.*”

Infrastructure management is a wide research area in a variety of IT domains and its challenges are under the spotlight in a range of seminal works: Sotomayor et al. [38] addressed the problem of Virtual Infrastructure Management in Private and Hybrid Clouds through the OpenNebula – an open source, virtual infrastructure manager that can deploy virtualized services on a local pool of resources as well as external IaaS Clouds; Yan et al. [39] introduced the Monsoon – a Cloud Management System that aimed to address the issue of the management of complexity of IaaS service model for enterprise users in Hybrid Cloud environments; Liu et al. [40] proposed the Multi Cloud Management Platform that aimed to locate Cloud Users and Cloud Sites and provide Unified Cloud

Services in the Multi-IaaS Cloud environment; Krintz and Wolski [41] described the issues in the Unified API Governance and suggested the solution in the form of a distributed software platform – the AppScale, which can manage, unify, deliver, and compose APIs in a commercial setting. Yet, currently there is no model proposed to address the complexity of management of distinct middleware infrastructures that drive the service-oriented integration.

III. VIABLE ENTERPRISE SERVICE BUS MODEL

The VESBM is proposed to overcome this challenge. Its aim is to provide a roadmap for the generation of a vendor-neutral blueprint for the management of distinct middleware infrastructures. By providing this blueprint, VESBM becomes a remedy against risks that may arise in enterprises that use middleware products from distinct vendors to integrate IT services. The VESBM is built upon the cybernetic principles of VSM and the service-oriented principles of ESB.

VSM is the cybernetic model for designing the control and communication aspects of a viable system. The VSM was created by Stafford Beer in 1972 and is fully described in his trilogy: *Brain of the Firm* [42], *Heart of Enterprise* [44] and *Diagnosing the System for Organisations* [45]. At the core of the VSM is the concept of viability that is – a system, which is “able to maintain a separate existence” [44]. VSM encompasses a wide range of concepts such as Viability, Self-reference, Invariance, Recursion, Management of Complexity, Variety, Homeostasis, Requisite Variety, Regulation, Regulatory Centre, Resource Bargain, Accountability, Channel Capacity, Transduction, Oscillation, Criteria of Variety Relevance, Comparator, Feedback, Convergence, Autonomy, Meta-system and Algedonic Signals [45]. All these concepts are built around five key management functions and one extended function that are Operations, Co-ordination, Control and Audit (the extension of Control), Intelligence, and Policy, symbolized in VSM as System ONE, TWO, THREE and THREE*, FOUR, and FIVE, respectively. Together, these functions are connected through a series of information channels and communication flows to form the VSM, as depicted in the figure below (Fig. 4).

During the last years, the VSM got an increasing attention from professionals in both industry and academia. It was adapted in a wide range of quite distinct domains, which continuously justified its interdisciplinary application evaluation. The VSM was used: in the creation of a model to structure a System Dynamics Mapping and Modelling Project for the Australian Taxation Office [46]; in the development of a Modelling Framework for Analysis of Viability in Service Systems [47]; in the business context to develop an Enterprise Process Architecture [48]; in the creation of the Enterprise

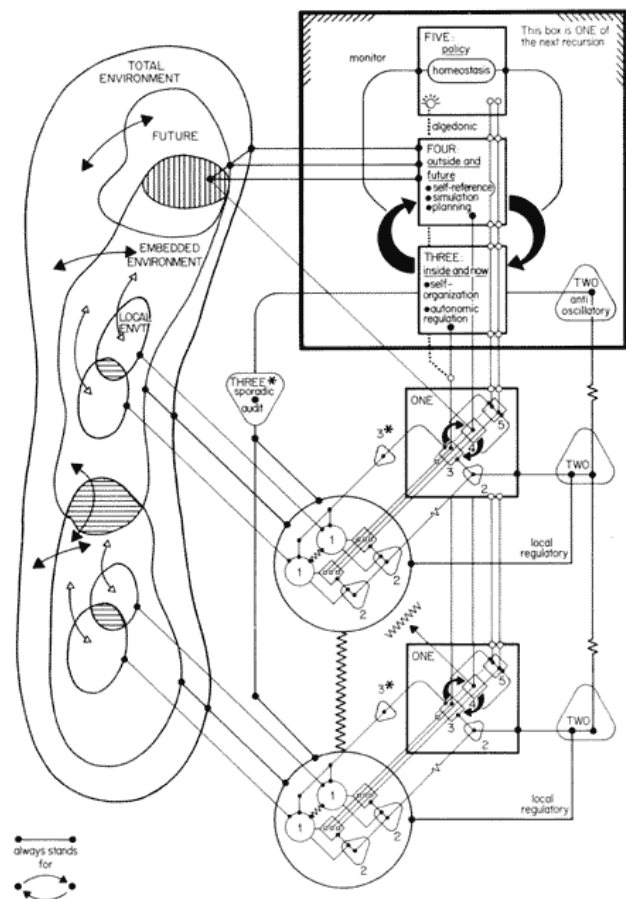


Fig. 4. The VSM of Stafford Beer [45].

Viability Model to extend the Enterprise Architecture Frameworks for Modelling and Analysing Viability under Turbulence [49]; as a potential approach for the Enterprise Architecture Management [50]; as a high-level reference architecture for complex systems in the development of the Viable System Architecture [51]; as an approach in mapping the Enterprise Architecture Principles in The Open Group Architecture Framework to the cybernetic concepts [52]; as well as in the creation of the Viable Governance Model – the theoretical model for the corporate governance of IT [53].

VESBM (Fig. 5) is the symbiosis of ESB and VSM, and thus shares the features common to both of them. It does not aim to change the design of existing ESB implementations available on the market, but to combine them under the vendor-neutral management umbrella. VESBM does not describe all of the ESB characteristics, but prescribes the core set it shall be endowed with to form a viable holistic whole. Thus, VESBM not only provides a blueprint for management, but can also help in revealing non-viable aspects in the design of the Service-oriented-system-in-focus.

By adapting the VSM, VESBM can provide the holistic model for the management of integration of IT services that are

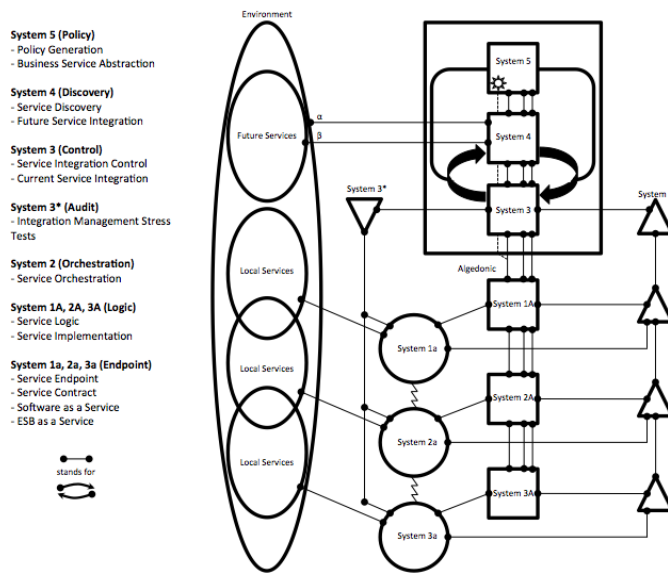


Fig. 5. Viable Enterprise Service Bus Model.

continuously justified against their viability to feed the relevant ITSM frameworks used in organisation (Fig. 6).

VESBM is structured according to the VSM to avoid possible disintegrations at various levels of Business/IT service abstractions.

IV. RESULTS

This section provides the results of the mapping of the cybernetic principles of VSM [45] to ESB [15], starting with the table below (Table I). This mapping reveals 21 aspects that need to be realized in the management of ESB and enriches them with the cybernetic principles of VSM. Where applicable correlation with ITSM is also provided.

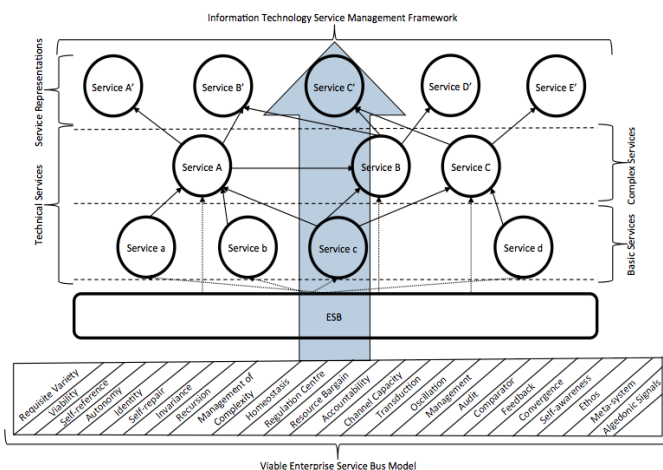


Fig. 6. VESBM feeding the ITSM.

TABLE I. RESULTS OF VSM TO ESB MAPPING (S – SIMILAR, P – PARTIALLY SIMILAR, X – NON-EXISTENT).

Aspect	VSM	ESB	Match
Conceptual Mapping			
1	• Variety	• Availability of a variety of tools ready for integration (e.g., application adapters, integration brokers, etc.)	• P
2	• Communication Channels	• Support of real-time throughput of business data and execution of process flows through secured and reliable channels	• P
3	• Preserving Value	• <i>Not Realised</i>	• X
4	• Viability	• <i>Not Realised</i>	• X
5	• Self-reference	• <i>Not Realised</i>	• X
6	• Invariance	• <i>Not Realised</i>	• X
7	• Autonomy	• Creation of federated environment with autonomous ESB instances/units	• S
Functional Mapping			
8	• System ONE • Recursion	• Geographical pervasiveness of ESBs; Service endpoints invocation; Support of remote configuration, management and control through external services; • One-by-one versus all-at-once model of incremental adoption of ESBs; Distributed integration and selective deployment of services and ESBs	• P • P
9	• Management of Complexity • Homeostasis • Requisite Variety	• Ability to orchestrate processes in dynamically changing environment • Extending ESB functionality through the use of layered services • <i>Not Realised</i>	• P • P • X
10	• Regulation Centre	• <i>Not Realised</i>	• X
11	• Resource Bargain • Accountability	• <i>Not Realised</i> • <i>Not Realised</i>	• X
12	• Channel Capacity • Transduction	• Balancing of network load • Use of common data type (e.g., XML); Mediation; Distributed data transformation; Standardised integration	• P • S
13	• System TWO/Oscillation	• <i>Not Realised</i>	• X
14	• System THREE/Management • System THREE*/Audit	• Management of ESB through specialised tools (e.g., Java Management Extension (JMX)) • Audit of the performance of ESB and integrated services	• P • P
15	• Comparator	• <i>Not Realised</i>	• X
16	• Feedback	• Replacement of services in times of failure through the use of re-routing capabilities	• P
17	• Convergence	• <i>Not Realised</i>	• X
18	• System FOUR/Self-awareness	• <i>Not Realised</i>	• X
19	• System FIVE/Ethos	• <i>Not Realised</i>	• X
20	• Meta-system	• <i>Not Realised</i>	• X
21	• Algedonic Signals	• Event-driven ESB; Complex event processing that involves time-bias	• P

A. Variety, Communication Channels, Preserving Value, Viability, Self-reference, Invariance, Autonomy (Aspects 1-7)

There is a range of conceptual similarities and differences between the VSM and the ESB. Both are designed to provide communication and control means for parts of organisations in which they are deployed. Both are endowed with communication channels that are used for information flow.

However, VSM is focused on managerial aspects of a System-in-focus and the analysis of its states using the variety as a measure of complexity that counts the number of possible

states, whereas ESB's *raison d'être* is to integrate disparate, heterogeneous applications using a wide variety of integration tooling to expose the applications in the form of services. Yet, one significant difference between VSM and ESB is the concept of viability that is fundamental to preserving value created in the System-in-focus, which in the given context is an integrated IT system.

In the VSM, viability implies separate existence. However, separate existence is interpreted in the context of autonomy that the system has in a given environment. Autonomy does not imply an existence in vacuum, but in the environment that may consist of other systems, each having a separate identity. The system is also self-referential, which means that: its logic is closed-in on itself and each part makes sense precisely in terms of the other parts; it is self-aware; it maintains its identity; has facilities for self-repair; and is recursive. The system is invariant, which means that the system is consistent at any level of its recursion.

The VESBM derives these VSM principles to create the foundation for the procedures that would guide the management of ESB. In the VESBM terms, ESB is treated as a viable entity within the ITSM scope for the given IT infrastructure: ESB shall maintain separate existence from other ESBs; ESB shall be autonomous; Autonomous ESB shall still interact with other ESBs; ESB shall be self-referential and its logic shall be closed-in on itself and each of its parts shall make sense precisely in terms of the other parts; ESB shall be self-aware; ESB shall have and maintain its identity; ESB shall have facilities for self-repair; ESB shall be recursive; and the core ESB structure, defined according to VESBM, shall remain consistent regardless of the scope it is used in and the abstraction the VESBM is applied to.

B. System ONE, Recursion (Aspect 8)

VESBM adapts the VSM Recursion to guide the implementation of incremental ESB integration projects. This can benefit IT infrastructure with managerial procedures to form pervasive ESB grids that are capable of spawning across enterprises, connecting distributed organisational departments, business units and corporate divisions in both centralised and decentralised manners.

In the VSM, System ONE is responsible for producing the viable system (Fig. 4, 5). System ONE of the System-in-focus forms a set of embedded viable systems. These viable systems are continuously interacting with the environment that surrounds them. Thus, none of the instances of the viable system exists in isolation. System ONE in many ways resembles the ESB, but its conceptual scope is broader. In the scope of VSM, subsidiaries, firms and corporations are all viable entities that contain organisations and are contained within them. In other words, all of them are the recursions of

the viable system. However, in the VSM recursion does not imply to loose insertion of one system into another, but to an absolutely precise definition of viability [45]. Similarly, depending on the size of organisation, a number of ESBs might be deployed within IT infrastructure. As with VSM, no ESB instance is assumed to exist in isolation, same as the departments, divisions and organisations that operate them. Usually, ESBs are assigned to particular divisions, but assignment of one ESB per department could also be a case. Together, these ESBs can form a grid, which is essentially a service-oriented environment that interconnects all the ESB participants into an ecosystem.

Conceptually, an ESB grid is no different to an ESB, yet their managerial scope is different. VESBM realizes this difference and can outline managerial procedures to guide the incremental adaption of ESBs in the environment, as the abstractions of ESB within IT infrastructure can be handled if formed according to the recursive structure of VSM.

C. Management of Complexity, Homeostasis, Requisite Variety (Aspect 9)

VESBM adapts the VSM Homeostasis and Requisite Variety to assist the ESB management in attenuating and amplifying the procedures for the ESB capability extension, so that enough requisite variety is available to deal with the changes in IT environment and internal states of ESB. This can benefit the ITSM procedures with additional extensions that consider the variety of technologies that can be employed in the ESB as a response to risks that can arise in IT environment.

In the VSM, variety is a measure of complexity that counts the number of possible states the System-in-focus can be in. For the System-in-focus the variety of its environment always greatly exceeds the variety of operations that serves or exploits it, which in turn exceeds that of management that regulates or controls it (Fig. 4, 5). To be able to handle variety the system shall reach a homeostatic state. Homeostasis is a special state within a given viable system where all the sub-systems are responsible for stabilizing its internal environment. To create acceptable conditions for homeostatic regulation, the Requisite Variety needs to be restored. The Law of Requisite Variety states that – only variety can absorb variety [38]. The restoration is achieved through the deployment of combination of attenuators and amplifiers that form continuous loops of variety involvement. VESBM can endow ESB with these attenuators and amplifiers in the form of relevant regulatory procedures to restore the variety of technologies employed within the ESB to acceptable homeostatic conditions, so that it can adequately respond to changes. Once these conditions are reached the managerial aspects of ESB can be exploited to further extend and interface with ITSM framework used in IT infrastructure.

D. Regulation Centre (Aspect 10)

VESBM adapts the VSM Regulation Centre to provide requisite variety of the management of ESB. This can benefit the alignment of regulatory methods of the lower- and higher-level service representations (Fig. 6) described within ITSM framework used in IT infrastructure.

In the VSM, any intervention of Senior Management to System ONE operations does not lead to Requisite Variety. Organisations usually employ a range of constraints for variety proliferation in the form of technologies or methods of operations. The need for regulation is apparent: *“The management of the System-in-focus, called the Senior Management, is IN PRINCIPLE unable to entertain the variety generated by any one (never mind all) of its subsidiary viable systems that constitute System One”* [45]. This statement implies decentralization and autonomy. Given that the focus is at services that constitute ESB, which are utilised by the subsidiaries of the organisation, a special Regulatory Centre needs to be deployed to amplify managerial variety and attenuate operational variety of the ESB. Thus, the VESBM adaption of Regulatory Centre can help in the alignment of regulatory methods of the lower- and higher-level service representations described in terms of ITSM, leading to homeostasis between the management and operations of the ESB.

E. Resource Bargain, Accountability (Aspect 11)

VESBM adapts the VSM Resource Bargain and Accountability to guide the allocation of shared resources. This can benefit IT infrastructure with measures to manage the performance of resources used by ESB in providing services with potentially overlapping functionality.

In the VSM, resource allocation, also known as Resource Bargain, is a dynamic process. It is described as the deal *“by which some degree of autonomy is agreed between the Senior Management and its junior counterparts”* [45]. It is responsible for the attenuation of possible alternatives as a part of a homeostatic loop between the management of System ONE and the Senior Management of the System-in-focus. In VESBM terms, it means that ESB needs to be endowed with a degree of autonomy and the Resource Bargain will eventually impose Accountability for the resources allocated. Accountability shall be interpreted in terms of variety engineering. In other words, as it is apparently impossible to monitor all the activities that are undertaken by ESB, Accountability shall be regarded as the high variety attenuator of possible outcomes. In conjunction with the Regulatory Centre, the adaption of Resource Bargain and Accountability by VESBM can help in defining management procedures, within the scope of ITSM framework in use, to mitigate the issues with the responsibility for the use of resources in the ESB.

F. Channel Capacity, Transduction (Aspect 12)

VESBM adapts the VSM Channel Capacity and Transduction to add redundancy to messages transmitted through ESB. This can benefit the ITSM framework used in IT infrastructure by ensuring that the designed procedures are defined according to the fully traversed messages.

In the VSM, transduction is responsible for encoding and decoding messages that have been transmitted through information channels. Once messages cross the system boundary, they get transduced so that the senders and receivers could interpret them. Similarly to VSM, ESB provides data transformation services at the core of integration strategy. ESB uses XML as the common data type for messages transmitted, which can be enriched as required. In the VSM, information channels are designed with little redundancy to transmit more information than generated at a given time [45]. Such redundancy is often necessary for the transmission of messages that need to be enriched so that the receivers could interpret them fully. VESBM can define the managerial procedures for the message exchanges in the ESB according to this requirement to further contribute to the integrity of ITSM framework used in IT infrastructure.

G. System TWO/Oscillation (Aspect 13)

VESBM adapts the VSM Anti-oscillation to avoid possible service performance deterioration during the interaction of several ESB instances. This can benefit the service match-up mechanisms used in ESB and described in ITSM.

In the VSM, System TWO is responsible for Anti-oscillatory activities in the System ONE of the System-in-focus (Fig. 4, 5). Oscillation is a specific sickness of homeostasis [45]. It leads towards continuous adjustments of every System ONE instance to every other one, within the given system. Because, the process is continuous, such adjustments may never end, eventually leading to Oscillation. Oscillation needs to be dumped, as it can lead to the draining of resources, disintegration and total collapse of the entire system.

System TWO is positioned as the device that aims to manage possible disintegrations and in the context of ITSM it can be compared to the team of managers responsible for the operational level agreements. These are the agreements between the IT service provider and other parts of the same organisation, which assist in the provision of services. These other parts may include departments as well as subsidiaries. In the scope of the same organisation, all its parts and especially its subsidiaries may have their own ESBs that are designed to provide a particular set of services specific to each subsidiary. Given this perspective, the negotiations that take place during the formation of agreements can be positioned as the mechanism to limit the freedom of variety generation in each subsidiary and its eventual proliferation in the System-in-focus.

In VESBM, these negotiations are the Anti-oscillatory activities that can provide performance measurement and aim to integrate subsidiaries, in this case through ESBs, under agreements to assist in providing services to end customers.

H. System THREE/Management and System THREE/Audit (Aspect 14)*

VESBM adapts the VSM Management and Audit to ensure that the existing services are operating at expected levels of performance. This can benefit the alignment of technical services available in IT infrastructure with generic service representations described in ITSM framework in use (Fig. 6).

In the VSM, System THREE is associated with the “here-and-now” daily Management of the organization [45] (Fig. 4, 5). System THREE has an extension known as Audit and marked as System THREE*. It is designed to undertake Audit activities in a sporadic, high variety and intra-operational manner. The responsibility of System THREE is to monitor how System TWO operates, whereas System THREE* is to undertake Audit activities directly upon the operations of the System ONE.

The activities of System THREE* are defined in terms of System THREE to restore its Requisite Variety. System THREE* is designed as the sixth channel within the system and is defined according to The First Axiom of Management, which underpins the following equation:

Variety of Audit channel = Total horizontal variety generated by System One minus Varieties of all other five channels (1)

In VESBM, this equation (1) means that the Audit channel can close the gap between the regular service management activities of ITSM and the irregular ones that take the form of irregular audits of the performance of services provided through ESB.

I. Comparator (Aspect 15)

VESBM adapts the VSM Comparator to ensure that services that constitute abstract service representations are meeting their purpose (Fig. 6). This can be beneficial in times when there is a need for the justification of the end results generated by services against the initial results expected to be generated by the relevant service representations described in a broader ITSM scope.

In the VSM, comparison of the declared purpose of the system against the purpose that is imputed from the results system delivers is done through a special mechanism known as a Comparator. The Comparator compares the purpose and the end results continuously and thus prevents ad-hoc workarounds when the purpose deviates from what has been declared. It is common for the purpose to be declared at higher-levels of

recursion, but then morph at lower-levels, as it gets transduced. Thus, it is natural for systems to take actions based on their interpretations of the declared purpose. This might change their initial states to the resultant ones and eventually lead to possible disequilibrium in the system. In VESBM, the Comparator can be used in designing managerial procedures that will prevent similar issues in the ESB management in times when services that supposed to implement particular functionality, deviate from it by either encompassing broader functionality or cutting it down to a narrow one.

J. Feedback (Aspect 16)

VESBM adapts the VSM Feedback as a complementary to the Comparator to guide the modification of purposes.

In the VSM, the Feedback mechanism, in conjunction with the Comparator, is used to drive the modification of the purpose of the system according to error signals received. In viable systems, a possible disequilibrium between the System-in-focus and the next tier systems can be avoided through corrective actions that converge the purposes the systems declare and thus bring them to compromise. In VESBM, Feedback can be used to define the procedures for error signaling to further extend and interface with ITSM framework used in IT infrastructure.

K. Convergence (Aspect 17)

VESBM adapts the VSM purpose Convergence to equalize the purposes of abstract service representations with the purposes of technical services that constitute them (Fig. 6). This activity can be beneficial in avoiding possible disintegrations of technical services with their actual service representations in times of modifications that are conducted at a broader ITSM scope.

In the VSM, convergence of purposes is a continuous process that is natural to the system. Because viable entities of the system are supposed to operate with a defined degree of freedom, possible disequilibrium of purposes of higher-level recursions with that of lower-level ones is accepted. However, as viable systems of the System-in-focus are supposed to have a separate existence, the difference in purposes is allowed as long as it does not leave the spectrum of overall cohesiveness of the System-in-focus. In VESBM, Convergence can be used to guide the alignment of services integrated through ESB, with the services described in a broader ITSM context.

L. System FOUR/Self-awareness (Aspect 18)

VESBM adapts the VSM Self-awareness to drive the integration strategy of future services that are external to the IT infrastructure and maintain the quality of service performance. This can benefit the process of choosing potential service candidates that might be needed as a part of ITSM strategy, but not be available in ESB.

In the VSM, System FOUR is responsible for providing Self-awareness for the System-in-focus (Fig. 4, 5). It is concerned with the management of outside-and-then of the system, and is in continuous interaction with external environment. In the ESB, the external environment includes services external to its ecosystem that can be found through the Universal Description Discovery and Integration (UDDI) or similar service directories. At a technical level ESB is already aware of inside-and-now as well as outside, but not necessarily -then.

What it misses is the managerial approach for the outside-and-then that would conceptually equalize it to the ITSM framework used in IT infrastructure. In the context of ITSM, System FOUR can be compared to the team of managers that is responsible for the Service Strategy, as espoused in ITIL [3]. It involves in-depth analysis of the environment prior to the Service Design phase. However, System FOUR is also endowed with a number of channels that aim to satisfy the common area of interests. Each area has dedicated Alpha and Beta loops. Both loops are comprised of Amplification and Attenuation channels.

In VESBM, Alpha loop takes the responsibility for the continuous monitoring of services that are potential candidates for integration through ESB. Its Alpha Amplifier is responsible for the continuous projection of the internal ESB service ecosystem to the external ones to identify the space of service relevance. Its Alpha Attenuator is responsible for the continuous proactive observation of the service relevance to avoid the waiting attitude for any potential service candidate appearance in the external environment. The Alpha loop does not consider services separately in each particular service directory (e.g., UDDI), but monitors the intersection of all the services in all the directories monitored (Fig. 4, 5). All of this is applicable to the Beta loop, with only one remark that the external environment of which it is concerned is of unknown future.

Purely monitoring activities can be implemented in the ESB through the deployment of services that would monitor service directories that are of interest. However, the Alpha and Beta channels in the VESBM also involve quality analysis that can enrich the monitoring with additional procedures, which in turn can influence the management of services within ITSM framework used in IT infrastructure.

M. System FIVE/Ethos (Aspect 19)

VESBM adapts the VSM Ethos to steer the services provided through ESB in the unknown, future and dynamically changing environment. This steering can benefit the process of creation of new, decommission of old and modification of existing services of IT infrastructure that are described in the scope of ITSM.

In the VSM, System FIVE is responsible for creating corporate ethos of the System-in-focus (Fig. 4, 5). Ethos is responsible for creating flexible atmosphere, instead of rigid set of objectives, to act as the “*variety sponge of gigantic capacity*” [45]. It is the logical closure of the VSM, the point of self-reference, where no more systems are placed above at a given level of recursion and where the identity is completely asserted.

In the context of ITSM, System FIVE can be compared to the team of managers that influences the course of IT infrastructure. In the VSM, System FIVE defines the rules that determine the criteria of relevance for the patterns that need to be recognised by System FOUR and filtered from less relevant ones in the unknown space of continuously changing future. In VESBM, these rules can be seen as service specifications that dictate what type of technology needs to be considered within ESB. Apparently, in some situations a particular technology might not be available, though the scope it will be operating in is known. Thus, the adaption of Ethos by VESBM can create the awareness of what kind of business requirements need to be met through the ESB service compositions as well as what kind of patterns of services and technologies need to be considered to implement these requirements.

N. Meta-system (Aspect 20)

VESBM adapts the VSM Third Axiom of Management and the Law of Cohesion to extend the meta-systemic scope of ESB. This can benefit the management of multiple ESB recursions through ITSM framework used in IT infrastructure.

In the VSM, a meta-system is a system that is over and beyond a system of lower logical order to which higher-level authority might not be applied. The Third Axiom of Management states that: “*The variety disposed by System FIVE equals the residual variety generated by the operation of the Second Axiom.*” Residual does not imply to small, but anything leftover, which may also be considerably large. In VESBM, this means that there would be potential service candidates, which might be suspended for integration through ESB at a given time. Though, their functionality might be required for the execution of service representations that are part of business models, the residual variety implies to the need for actions that must be taken by the senior management prior such integrations take place. To apply this equally to more abstract service candidates, such as multiple ESB instances in IT infrastructures, the Law of Cohesion must be adapted: “*The System One variety accessible to System Three of Recursion x equals the variety disposed by the sum of the meta-systems of Recursion y for every recursive pair.*” This statement is a generic interpretation of the First Axiom of Management, which implies the applicability of an ITSM framework not only to services, but also to ESBs that integrate them in IT

infrastructure.

O. Algedonic Signals (Aspect 21)

VESBM adapts the VSM Algedonic Signals for the crisis signaling capability purposes. This can benefit the management of asynchronous events that take place in ESB and because of time bias not realized in the ITSM used in IT infrastructure.

In the VSM, System FIVE is prone to occupational hazards that can put it in a somnolent state. As in ESB, System FIVE may accidentally ignore particular events, because all the filters on the main axis may drown the entire system. Thus, System FIVE might just 'fall asleep' and ignore the events that require attention. To wake up the system VSM employs a special alarm signal, known as Algedonic (stands for both pain and pleasure), which performs non-analytical regulation in the viable system. It divides the signals ascending from System ONE, which enter meta-systemic filtrations and uses its own Algedonic filter to decide whether the System FIVE shall be alerted or not. VESBM adapts the Alegodnic Signal to define procedures that undertake two roles: at the managerial level they solve the time bias problem of asynchronous events management, through the employment of procedures that would identify services that are either operating improperly or require close attention prior next use in the business process automation; at the technical level they are supported by automated events management services present in ESBs, so that the system can be woken up to pay attention to a particular event in the pool. These procedures can then be aligned with ITSM framework to further contribute to the integrity of IT infrastructure.

V. EXPERIMENTAL SETTING

The VESBM current research application is within an Australian governmental department that includes a case study of adaption of VESBM to the ITSM practices to identify the managerial gaps in the services that are integrated within the organisation's infrastructure and extend them as required. The experiment is comprised of five stages:

1. Analyzing existing ITSM frameworks used in organisation (e.g. ITIL, COBIT, customised, etc.) to identify the managerial procedures
2. Mapping the VESBM to the ITSM framework to reveal the gaps in the procedures
3. Extending the ITSM framework through VESBM according to the gaps identified
4. Running the VESBM-enabled ITSM framework in the practical setting
5. Awaiting for feedback on the use of VESBM for the ITSM

Once the experiment is conducted and the feedback is received, the VESBM enrichment of ITSM framework can undergo additional iterations until the framework meets the levels that satisfy the service management requirements.

VI. CONCLUSION

This paper provided the results of the mapping of the cybernetic principles of VSM to ESB. This mapping is one of the steps towards creation of the model of VESBM that aims to become a roadmap for the generation of a vendor-neutral blueprint for the management of distinct middleware infrastructures, such as the ESBs, which drive the integration of the services that form the IT infrastructure. VESBM is the result of the symbiosis of the cybernetic model of VSM and the service-oriented model of ESB. The need for VESBM is especially actualized in the era of Cloud. VESBM aims to fill a specific niche by combining distinct ESB implementations available on the market under the managerial umbrella of VSM, without compromising their design.

The goal of this experiment was to identify gaps and extend the management principles of service-oriented integration with design propositions based on the principles of cybernetics. It provides a solid foundation for future research as the model can be combined with decision support systems, work systems, service management methodologies and other best practices that can aid the management of complex service-oriented integration projects.

REFERENCES

- [1] J.-Y. Chung and K.-M. Chao, "A View on Service-oriented Architecture," *Serv. Oriented Comput. Appl.*, vol. 1, no. 2, pp. 93–95, 2007.
- [2] M. M. Lankhorst, "Enterprise Architecture Modelling—The Issue of Integration," *Adv. Eng. Inform.*, vol. 18, no. 4, pp. 205–216, 2004.
- [3] J. van Bon, *IT Service Management: An Introduction*. Van Haren Publishing, 2007.
- [4] S. Geric, "The Potential of Service-oriented Architectures," in *32nd International Conference on Information Technology Interfaces (ITI)*, 2010, pp. 471–476.
- [5] B. Bygstad and T.-M. Gronli, "Service Oriented Architecture and Business Innovation," in *44th Hawaii International Conference on System Sciences (HICSS)*, 2011, pp. 1–10.
- [6] C. Abrams and W. R. Schulte, "Service-Oriented Architecture Overview and Guide to SOA Research," *Gart. Res.*, 2008.
- [7] C. Baroudi and F. Halper, "Executive Survey: SOA Implementation Satisfaction," *Hurwitz Assoc.*, 2006.
- [8] A. Arsanjani, "Service-oriented Modeling and Architecture," *IBM Dev. Works*, 2004.
- [9] T. Erl, *Service-Oriented Architecture (SOA): Concepts, Technology, and Design*, vol. 31. Prentice Hall, 2005.
- [10] M. Papazoglou and W.-J. Van Den Heuvel, "Service Oriented Architectures: Approaches, Technologies and Research Issues," *VLDB J.*, vol. 16, no. 3, pp. 389–415, 2007.
- [11] W. Roshen, "Enterprise Service Bus with USB-Like Universal Ports," in *9th IEEE European Conference on Web Services (ECOWS)*, 2011, pp. 177–183.

- [12] M. Keen, A. Acharya, S. Bishop, A. Hopkins, S. Milinski, C. Nott, R. Robinson, J. Adams, and P. Verschuere, *Patterns: Implementing an SOA Using an Enterprise Service Bus*. IBM, International Technical Support Organization, 2004.
- [13] T. Erl, *SOA Design Patterns*. Pearson Education, 2008.
- [14] F. Menge, "Enterprise Service Bus," in *Free and Open Source Software Conference*, 2007, vol. 2, pp. 1–6.
- [15] D. A. Chappell, *Enterprise Service Bus: Theory in Practice*. O'Reilly Media, Inc., 2004.
- [16] M.-T. Schmidt, B. Hutchison, P. Lambros, and R. Phippen, "The Enterprise Service Bus: Making Service-oriented Architecture Real," *IBM Syst. J.*, vol. 44, no. 4, pp. 781–797, 2005.
- [17] S. Ortiz, "Getting on Board the Enterprise Service Bus," *Computer*, vol. 40, no. 4, pp. 15–17, 2007.
- [18] P. De Leusse, P. Periorelli, and P. Watson, *Enterprise Service Bus: An overview*. University of Newcastle upon Tyne, Computing Science, 2007.
- [19] S. Desmet, B. Volckaert, S. Van Assche, D. Van Der Weken, B. Dhoedt, and F. De Turck, "Throughput Evaluation of Different Enterprise Service Bus Approaches," in *Proceedings of SERP2007, the 2007 International Conference on Software Engineering Research & Practice (part of the 2007 World Congress in Computer Science, Computer Engineering, and Applied Computing)*, 2007, pp. 378–384.
- [20] A. T. Manes, "Enterprise Service Bus: A Definition," *Burton Group*, 2007.
- [21] M. Luo, B. Goldshlager, and L.-J. Zhang, "Designing and implementing Enterprise Service Bus (ESB) and SOA Solutions," in *IEEE International Conference on Services Computing*, 2005, vol. 2, p. xiv–vol.
- [22] J. L. Bower and C. M. Christensen, *Disruptive Technologies: Catching the Wave*. Harvard Business Review, 1995.
- [23] D. S. Linthicum, *Cloud Computing and SOA Convergence in Your Enterprise: A Step-by-step Guide*. Pearson Education, 2009.
- [24] A. Fox, R. Griffith, A. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, and I. Stoica, "Above the Clouds: A Berkeley View of Cloud Computing," *Dept Electr. Eng Comput Sci. Univ. Calif. Berkeley Rep UCBECS*, vol. 28, p. 13, 2009.
- [25] S. J. Andriole, "Seven Indisputable Technology Trends That Will Define 2015," *Commun. Assoc. Inf. Syst.*, vol. 30, 2012.
- [26] T. Dillon, C. Wu, and E. Chang, "Cloud Computing: Issues and Challenges," in *24th IEEE International Conference on Advanced Information Networking and Applications (AINA)*, 2010, pp. 27–33.
- [27] L.-J. Zhang, J. Zhang, J. Fiaidhi, and J. M. Chang, "Hot Topics in Cloud Computing," 2010.
- [28] NIST, "Final Version of NIST Cloud Computing Definition, <http://www.nist.gov/itl/csd/cloud-102511.cfm>, Accessed: March 12, 2014," 2011.
- [29] P. Banerjee, C. Bash, R. Friedrich, P. Goldsack, B. A. Huberman, J. Manley, C. Patel, P. Ranganathan, and A. Veitch, "Everything as a Service: Powering the New Information Economy," *Computer*, vol. 44, no. 3, pp. 36–43, 2011.
- [30] G. Mateescu, W. Gentzsch, and C. J. Ribbens, "Hybrid computing—where HPC meets grid and cloud computing," *Future Gener. Comput. Syst.*, vol. 27, no. 5, pp. 440–453, 2011.
- [31] R. Buyya, C. S. Yeo, S. Venugopal, J. Broberg, and I. Brandic, "Cloud computing and emerging IT platforms: Vision, hype, and reality for delivering computing as the 5th utility," *Future Gener. Comput. Syst.*, vol. 25, no. 6, pp. 599–616, 2009.
- [32] B.-G. Chun and P. Maniatis, "Augmented Smartphone Applications Through Clone Cloud Execution," in *HotOS*, 2009, vol. 9, pp. 8–11.
- [33] I. Sriram and A. Khajeh-Hosseini, "Research agenda in cloud technologies," *ArXiv Prepr. ArXiv10013259*, 2010.
- [34] I. Gat and G. Succi, "A Survey of the API Economy," *Cut. Consort.*, 2013.
- [35] G. Succi and T. Remencius, "Profiting in the API Economy, <http://www.cutter.com/itjournal/fulltext/advisor/2013/itj131002.html>, Accessed: March 22, 2014," *Cut. Consort.*, 2013.
- [36] Layer7, "API Security and Management Today, www.layer7tech.com/infographic, Accessed: March 24, 2014," *Layer7*, 2013.
- [37] B. Golden, "Cloud Computing Calls for Rebuilding Enterprise IT, http://www.cio.com/article/702585/Cloud_Computing_Calls_for_Rebuilding_Enterprise_IT_, Accessed: March 24, 2014," 2012.
- [38] B. Sotomayor, R. S. Montero, I. M. Llorente, and I. Foster, "Virtual Infrastructure Management in Private and Hybrid Clouds," *Internet Comput. IEEE*, vol. 13, no. 5, pp. 14–22, 2009.
- [39] S. Yan, B. S. Lee, G. Zhao, D. Ma, and P. Mohamed, "Infrastructure Management of Hybrid Cloud for Enterprise Users," in *5th International DMTF Academic Alliance Workshop on Systems and Virtualization Management (SVM)*, 2011, pp. 1–6.
- [40] T. Liu, Y. Katsuno, K. Sun, Y. Li, T. Kushida, Y. Chen, and M. Itakura, "Multi Cloud Management for Unified Cloud Services Across Cloud Sites," in *IEEE International Conference on Cloud Computing and Intelligence Systems (CCIS)*, 2011, pp. 164–169.
- [41] C. Krantz and R. Wolski, "Unified API Governance in the New API Economy, <http://www.cutter.com/itjournal/fulltext/advisor/2013/itj131002.html>, Accessed: March 29, 2014," *Cut. Consort.*, 2013.
- [42] S. Beer, *Brain of the Firm: The Managerial Cybernetics of Organizations*. London: Allen Lane and Penguin Press, 1972.
- [43] S. Beer, *Brain of the Firm (2nd ed.)*. Chichester: John Wiley & Sons, 1981.
- [44] S. Beer, *The Heart of Enterprise: Companion volume to Brain of the Firm*. Chichester: John Wiley & Sons, 1979.
- [45] S. Beer, *Diagnosing the System for Organizations*. Chichester: John Wiley & Sons, 1985.
- [46] T. Haslett and R. Sarah, "Using the viable systems model to structure a system dynamics mapping and modeling project for the Australian Taxation Office," *Syst. Pract. Action Res.*, vol. 19, no. 3, pp. 273–290, 2006.
- [47] A. Golnam, G. Regev, and A. Wegmann, "On Viable Service Systems: Developing a Modeling Framework for Analysis of Viability in Service Systems," in *Exploring Services Science*, Springer, 2011, pp. 30–41.
- [48] R. Vidgen, "Cybernetics and business processes: using the viable system model to develop an enterprise process architecture," *Knowl. Process Manag.*, vol. 5, no. 2, pp. 118–131, 1998.
- [49] O. O. Oduntan and N. Park, "Enterprise Viability Model: Extending Enterprise Architecture Frameworks for Modeling and Analyzing Viability under Turbulence," *J. Enterp. Transform.*, vol. 2, no. 1, pp. 1–25, 2012.
- [50] S. Buckl, F. Matthes, and C. M. Schweda, "A Viable System Perspective on Enterprise Architecture Management," in *Systems, Man and Cybernetics, 2009. SMC 2009. IEEE International Conference on*, 2009, pp. 1483–1488.
- [51] C. Herring and S. Kaplan, "The Viable System Architecture," in *System Sciences, 2001. Proceedings of the 34th Annual Hawaii International Conference on*, 2001, p. 10–pp.
- [52] M. E. Zadeh, G. Millar, and E. Lewis, "Mapping the Enterprise Architecture Principles in TOGAF to the Cybernetic Concepts – An Exploratory Study," in *System Science (HICSS), 2012 45th Hawaii International Conference on*, 2012, pp. 4270–4276.
- [53] E. Lewis and G. Millar, "The Viable Governance Model: A Theoretical Model for the Corporate Governance of IT," in *System Sciences, 2009. HICSS'09. 42nd Hawaii International Conference on*, 2009, pp. 1–10.