Transdisciplinary Engineering: A Paradigm Shift
C. Chen et al. (Eds.)

doi:10.3233/978-1-61499-779-5-67

Zachman Framework in the Agile Digital Transformation

Sergej BONDAR[a], John C. HSU[b], Alain PFOUGA[a] and Josip STJEPANDIĆ[a,1]
[a] *PROSTEP AG, Germany*
[b] *California State University, Long Beach, USA*

Abstract. Emergent behavior is behavior of a system that does not depend on its individual parts, but on their relationships to one another. Such behavior exists in biological systems, physical systems as well as in the human performance. It is an inherited nature of a System-of-Systems (SoS). A suitable framework is needed to guide the development of SoS architecture, which includes emergent behavior. Enterprise architecture (EA) is a discipline driving change within organizations. Aligning and integrating business and IT thereby belongs to strategic management. The management of EA change is a challenging task for enterprise architects, due to complex dependencies amongst EA models, when evolving towards different alternatives. In this paper, various architecture frameworks are explored for an application on SoS architecture. The use of Zachman Framework to guide the architecture development is described in step-by-step details in this paper. The agent-based simulation is recommended to develop the SoS architectural models following the Zachman Framework guidance. Ultimately, SysML and UML should be integrated with the agent-based model. An example with the collaborative engineering services for the global automotive supply chain is hereby described.

Keywords. Agile Digital Transformation; Architecture Reference Model; Systems of Systems Engineering; Engineering Collaboration; Zachman Framework

Introduction

There are numerous instances of literature describing the existence of emergent behaviors in human performance, physical systems, biological systems and economics [1]. Emergence is when some totally new phenomenon emerges out of the collective behavior of much simpler parts where the individual simpler parts are responding through simple rules to their local environment. The emergent behaviors exist in global grids due to Scale, Communications Locality, Element Simplicity, Feedback and Element Autonomy [2]. Change made to behavior in individual components could have pervasive and unexpected effects on global behavior. Emergent behaviors arise from the cumulative actions and interactions of the constituents of a System-of-Systems (SoS). The behavior and/or performance of the SoS cannot be represented in any form that is simpler than the SoS itself [3]. Understanding and harnessing these (emergent behavior) effects is crucial to success of SoS [4].

Digital transformation has multiple definitions. A broader one defines it as a consistent networking of all economic sectors and as adaption of actors to new

[1] Corresponding Author, Mail: josip.stjepandic@opendesc.com.

circumstances of the digital economy [5]. The ability to adapt quickly to change also stands out as an important capability. Experts have told, that agility is more important than technology skills. The 21st century is about agility, adjustment, adaptation and creating new opportunities [6]. The main finding of this study says that maturing digital businesses are focused on integrating digital technologies, such as social, mobile, analytics and cloud, in the service of transforming how their businesses work [7].

There are discussions and research about how to model and develop a SoS architecture including the emergent behavior. The architecture framework will have to be chosen first prior to modeling the SoS architecture. This goal is demonstrated by a use case from engineering collaboration with a high level of agility, which comprises impact from strategy, technology and global market [8].

The remainder of this paper is organized as follows. The section 1 gives the comprehensive overview of the most relevant commercial applications of the architecture framework, in particular in the Zachman framework for emergent behavior. The application of the agile digital transformation of a global supplier portal is described in section 2. Conclusions and outlook close the paper in section 3.

1. Commercial Applications

All the frameworks are applicable to a single system, but not all the frameworks are suitable for the SoS. The applicability of architecture frameworks to a SoS need to be explored.The most popular architecture framework is DoDAF (Department of Defense Architecture Framework) [9]. It is a sophisticated and well-defined framework with three views with its own products. The requirements of these products are detailed and structurally described. It leads to limited ways and means to comply. Emergent behavior is non-linear as explained above. “Interface” is closely related to integration that the missing of a system, subsystem or component will make the integration impossible. Thus, “Interface” is not applicable to the SoS with emergent behavior. For these reasons, DoDAF is not suitable as an architecture framework guide for the development of a SoS architecture.

MODAF (Ministry of Defense Architecture Framework) is an extension of DoDAF by adding the Strategic View and Acquisition View. The strategic view is to support the capability management process. The six (6) products are text, phasing, mapping to deployment and operations, and descriptions of the relationships between capabilities. They can be modeled for a single system or SoS. MODAF is not suitable as an architecture framework guide for the development of a SoS architecture.

The TOGAF (The Open Group Architecture Framework) Architecture Development Method (ADM) is flexible in that it may be used with a set of deliverables from another framework, or it may even be used in conjunction with the Zachman Framework [10][11]. The TOGAF ADM is designed to allow tailoring by an individual organization. It is intended that users of the TOGAF will use the TOGAF ADM as a guide in designing their architecture. Rather than being a restrictive method and forcing extraneous tasks to be performed, the TOGAF ADM allows an individual organization to choose to bypass, or tailor, any part of the process as required. The ADM is a generic method for architecture development, which is designed to deal with most system and organizational requirements. This suits well with loose coupling and emergent behavior characteristics of a SoS environment.

Zachman Framework is suitable for complex systems, such as SoS. Architecture is relative and there is a set of them, organized around the points of view taken by various players. They included (1) the planner or whoever is setting the agenda and strategy for an organization, (2) the owner who runs the organization, (3) the designer who wants to represent the business in a disciplined form, (4) the builder who applies specific technologies to solve the problems of the business and (5) the subcontractor. Each represents a perspective as a row in a matrix. Columns in the matrix represent the kinds of things people should be looking at. These include functions and data, as addressed by most methodologies. In addition, columns represent locations where business is done, the people and organizations involved, events, which cause things to happen, and the motivations, and constraints, which determine how the business behaves.

Is Zachman Framework suitable for developing a SoS architecture? It will be explored here. The Zachman Framework is a classification schema to organize "primitive" architectural information; no specific models, no methodology and no notation. The Zachman Framework can be used to describe any complex entity. It is product neutral and not like DoDAF and MODAF prescribing specific products. These basic features are advantages for developing SoS architectures with emergent behavior since the architecture developer will have the freedom to incorporate the latest modeling technique of emergent behavior.

The Systems-of-Systems (SoS) architecture is a layered architecture model which allows different developers to work in parallel and insure that changes in one layer of the protocol do not interfere with operations above and below that layer. Thus, layered architectures implement loose coupling between the services that makes up the overall SoS. System design including hardware and software will be based on architecture models in different levels. The first-layer is SoS level, the second-layer is component system level, the third-layer is subsystem level, and the fourth-layer is subsubsystem level, etc. There is only one first-layer for a single SoS but there are more than one second-layers depending on how many component systems etc. The layers are propagated in one-to-many relationships to the last layers. The first-layer of the Zachman Framework for a SoS is shown in Figure 1 [12].

2. Digital Transformation of a Global Supplier Portal

Canges in automotive supply chain lead to over 70% of the R&D shifted from OEMs to partners. This results in a big investment and development of R&D capacities [13]. System suppliers and joint ventures are thus growing in the innovation and value added networks into a new strategic role [14][15]. The creational performance of an OEM basically isn't measured by means of its own internal capabilities anymore. It depends on a collaborative SoS built from concert of its internal domains as well as these of its supply chain partners. These shifts essentially require partners to significantly expand their cooperation capability both vertically and horizontally, to redefine their strategic role within network structures and to speed up time-to-market of an underlying automotive SoS [16]. Efficient data communication in all phases is prerequisite for lean, agile and flexible collaboration; it is required to concentrate on differentiating strongly on development and technology competence [17]. Automotive suppliers that supply system components for a number of different OEMs or tier-1 suppliers are facing multiple challenges e.g. to ensure that they make components available according to customer-specific requirements and with a high level of resilience and reliability.

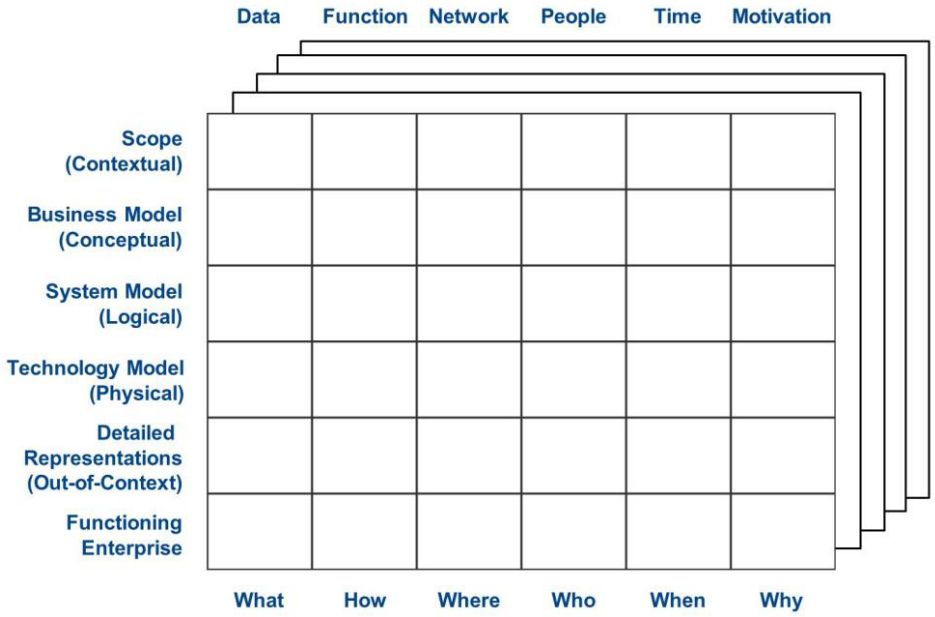

Figure 1. Zachman Framework in Layers.

As far as engineering data communication is concerned, automotive suppliers have to take numerous requirements into consideration that are not always easy to reconcile [18]. Multi CAD collaboration requirements are thereby specific for each development request coming from their various OEMs (Figure 2). On a collaboration perspective, each design change not only requires a comprehensive data conversion in each direction for identical product representations delivered to multiple OEMs. It involves taking multiple aspects into account related with substantially customized data, process, milestones, sites and tools aiming at improving the specific OEM collaboration experience; which very often are hardly available on premise. Any design iteration in change management extrapolates in requirements to managing the data exchange.

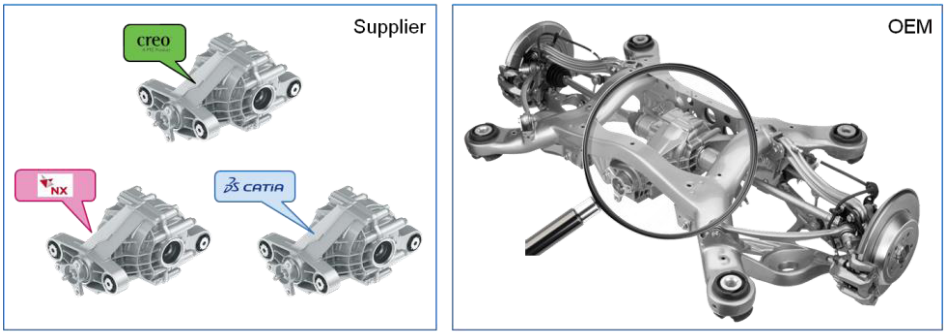

Figure 2. Multi CAD collaboration requirements [19].

OpenDESC.com is an industry-focused portal for engineering collaboration with features serving especially automotive throughout an extended enterprise. It is a holistic service, which includes both (1) translation of engineering data into a custom environment and (2) secured provision of engineering data to partners in an automotive

SoS. The conception shown below describes a high-level architecture of the platform on the perspective of 6 selected layers of the Zachman Framework (Table 1).

Table 1. High-Level Zachman Framework for OpenDESC.com.

	What	**How**	**Where**	**Who**	**When**	**Why**
Scope	Engineering product data provided by a customer or its partner are submitted according to specific constraints	Translation and exchange are thereby offered as cloud service	Global: All engineering sites of the involved parties are affected	OEMs and tier-X system suppliers of a customer	At each design change from an involved party	Achieve enterprise cross-collaboration without need for an infrastructure on premise
Enterprise Model	**Entity**=Business Organizations placing Order having Engineering Data and Tracing collaboration activities based on Report **Relationship**= Contract (SLA)	**Process**=Change Mngmt, On-boarding, Data Transmission, Data Reception, Support	**Node**=Business Locations (mostly engineering) **Link**=Connection	**People**= Sales, Design Department, Service Provide **Work**=Service Level Agreement, Process Order	On Data Submission, On Data reception	Enable engineering organizations to realize fast, easier and reliable communication demand
System Model (Logical)	**Entity=** Electronic DataSets contained in a Job submitted to Users in Organizations having a contractual Agreement **Relationship=**Person In Organization	**Process=** Set Up, Export Data, Import Data **I/O=**Web All parties must be registered to the platform and enabled to use services.	**Node**=see Figure 3 **Link**=Web Portal, OFTP over ISDN/ENX	**People**=Sales Person, Designer, Administrator, Sender, Receiver **Work**=Engineering Design, Send, Receive Data, Manage Services	On request	A party may be requested to provide, on a given date, data in a particular configuration to a customer.
Technology Model (Physical)	Model based Physical Data models with diagrams of the technology architecture, control structures, definitions and descriptions are realized using UML techniques. Users are interfacing directly with the service using (1) a web front end or (2) by means of an application programming interface for machine to machine communication. The UI is thereby described using graphical mock ups.					
Detailed Representations	By using templates, each customer of OpenDESC.com is easily set up as a node with its specific characteristics based thereby. Each node is logically separated from other customers in order to fulfill stringent security requirements. The operation of a node is conducted over an administration and operation interface, which allows an update of the node database.					
Functioning Enterprise	Each authorized user can create communication relationships, define rules for exchange, timing, quantity, desired data quality, level of detail and protection of intellectual property, add required metadata, distribute data sets to multiple users etc.					

Organizations who work together with different OEMs and tier-1 suppliers have to constantly cope with new requirements relating to exchange partners, data formats, system environments to be supported, quality and security requirements, etc. This

involves considerable administrative overhead in terms of time and money, which can on occasion have a negative impact on quality and adherence to deadlines [20]. Online collaboration is a big challenge in the field of international product development in a cross-language environment. It serves two aims: cross-language translation and design requirement clarification [21]. Scenarios are defined using planning techniques [22].

Aim of OpenDESC.com is to connect organizations within the extended enterprise, in a way that enables transfer and translation of engineering data submitted by a sender to a recipient based on specific-requirements of the receiver. The service is provided as cloud offering at global scale to customers and their OEM and tier-x partners anytime, when change orders are processed and handed over to partner [14]. Collaboration with OpenDESC.com is an alternative to not only cut costs but also facilitate making the exchange processes uniform and ensures a higher level of reliability, without having to invest in infrastructure. The results of research in the construction industry show that people collaborating using IT communication, at worst are as effective as people working face to face and are probably slightly more effective. This is a surprising result as IT communication lack rich and valuable non-verbal aspects of communication [23].

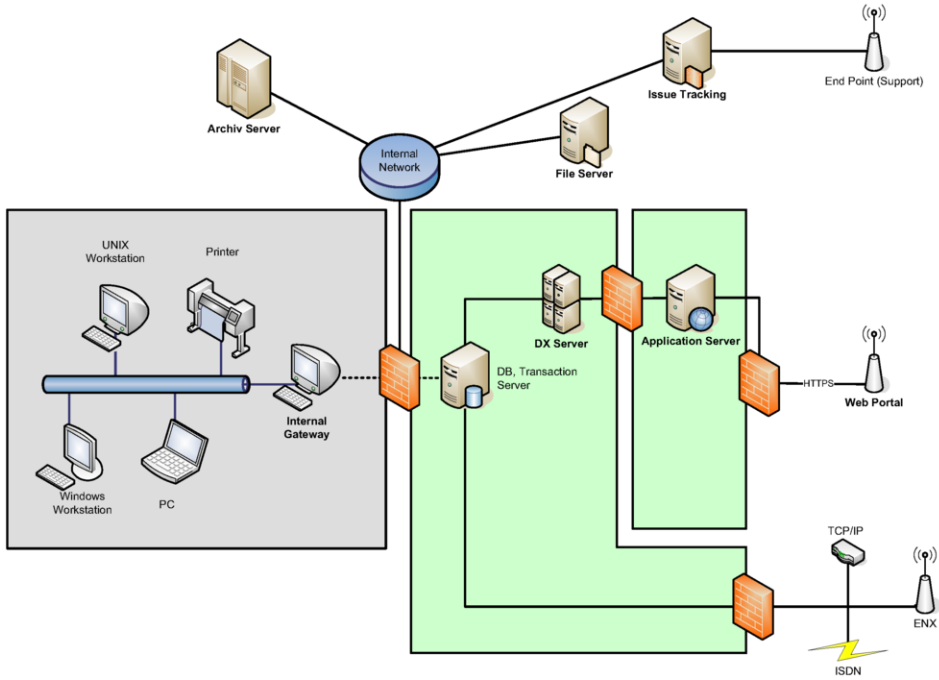

Figure 3. The platform conceptual architecture.

On a business model perspective OpenDESC.com allows business entities to process orders between development sites. The service offering is thereby governed by a Service Level Agreement (SLA) that covers processes like on-boarding new partners, transmitting and receiving data as well as support levels and decommissioning of registered users. Strategic agility remains the key factor in this contract. Having this in place, connections can be setup between engineering parties based on mechanisms and networks such as the European Network Exchange (ENX), the Odette File Transfer Protocol (OFTP) or by means of engineering portals just to name a few.

Each party in the collaborative SoS is provided with a platform on pay by use basis, enabling users to achieve fast, easier and reliable communication on demand, when an order is submitted [24]. The system model is built of users and units, the offering and jobs performed between users granted to use the service. Auxiliary infrastructure belonging to customers and OEMs are docked to this offering in a network realizing a system of collaborative systems in the various departments.The technology model of the platform is implemented with a redundant infrastructure taking stringent security aspects into account regarding access, data management and processing (Figure 3). On the user side, it provides a number of user frontends to services as well as machine-to-machine communication for an automatic processing of collaboration workflows [25]. The UI is described using graphical mock ups to visualize results at any stage.

Model based physical data models with diagrams of the architecture, control structures, definitions and descriptions are realized using UML. Use of node templates enable each customer of OpenDESC.com to be set up easily as a node instance with its own specific characteristics. Each node is logically separated from other customers in order to fulfill stringent security requirements. The operation of a node is conducted over an administration and operation interface, which allows its overall management. Each authorized user can create communication relationships, define rules for exchange, timing, quantity, desired data quality, level of detail and protection of intellectual property [26], add required metadata, distribute data sets to multiple users etc.

3. Conclusion

Emergent behaviors exist in biological system, physical system, human performance and economics. Emergence can be beneficial, harmful, or both. These behaviors exist in complex systems or SoS. Emergence is the primary mechanism for both operational success and operational failure in SoS. Success in SoS requires recognition, effective management, and exploitation of emergence. An accurate and complete system architecture model for a SoS is required to measure the existence, type, and level of emergent behavior of the SoS.

At the present time the Zachman framework is best suitable for developing a SoS architecture with the inclusion of emergent behavior. TOGAF and FEAF are also suitable since they are either adaptable to using and already include the Zachman framework features, respectively. DoDAF, MODAF and similar frameworks are not suitable to develop a SoS architecture including the emergent behavior.

Customized service-oriented value chains, which incorporate environment and social values, can be designed according to different conditions. Also, this system engages the advantages from both product-oriented and service-oriented companies to build a more comprehensive value network. Such a solution can serve as a decision support and benchmarking system because decision makers can develop different value networks according to various emphasized values [27].

Digital technologies, such as cloud computing and mobile apps, have the potential to transform significantly the way businesses run. The role of enterprise architecture is seen as important, especially in communicating business plans across the organization and defining a comprehensive framework. It could potentially be a good fit as one of the tools to be utilized in the latter stages of the Digital Transformation model (e.g. selecting one of multiple service offerings, [28]) but isn't definitely a solution to all of the different challenges, such as forgetting about the needs of the customer.

References

[1] J.C. Hsu and M. Butterfield, *Modeling Emergent Behavior for Systems-of-Systems*, 17th Annual INCOSE International Symposium, 2007.

[2] H. Van Dyke Parunak and R. S. VanderBok, *Managing Emergent Behavior in Distributed Control System,* ISA-Tech, Anaheim, California, USA, 1997.

[3] M. Maier, Architecting Principles for Systems-of-Systems, *Proceeding of the 6th Annual INCOSE Symposium*, 1996, pp. 567-574.

[4] D. Fisher, *An Emergent Perspective on Interoperation in Systems of Systems,* (CMU/SEI-2006-TR-003), Pittsburgh, PA, Software Engineering Institute, Carnegie Mellon University, March 2006.

[5] N. N., *Die Digitale Transformation der Industrie*, Roland Berger, München, 2015.

[6] G.C. Kane, D. Palmer, A.N. Phillips, D. Kiron and N. Buckley, Strategy, not Technology, Drives Digital Transormation: Becoming a digitally mature enterprise, MIT Sloan Management Review, July 2015.

[7] M. Hirt and P. Willmott, *Strategic principles for competing in the digital age*, McKinsey Quarterly, May 2014.

[8] S. Bondar, J.C. Hsu and J. Stjepandić, Network-centric operations during transition in global enterprise, *Int. J. Agile Systems and Management*, Vol. 8, 2015, Nos 3/4, pp. 355–373.

[9] Department of Defense (DoD) Architecture Framework Working Group, *DoD Architecture Framework*, Version 1, volume II: Product Descriptions, February 2004.

[10] The Open Group, The Open Group Architecture Framework (TOGAF), Version 8.1, Enterprise Edition, December 2003.

[11] J. A. Zachman, A framework for information systems architecture, *IBM Systems Journal*, Vol. 26. No 3, 1987.

[12] ANSI/AIAA G-043, *Guide for the Preparation of Operational Concept Documents,* Draft 2.0, 29 August 2006.

[13] A. Katzenbach, Automotive, in: J. Stjepandić et al. (eds.) *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, Springer International Publishing, 2015, pp. 607–638.

[14] Borsato, M. and Peruzzini, M. (2015) 'Collaborative Engineering', in: Stjepandić J. et al. (eds.): *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, pp. 165–196, Springer International Publishing Cham.

[15] R.C. Beckett, Functional system maps as boundary objects in complex system development. *Int. J. Agile Systems and Management,* Vol. 8, 2015, No. 1, pp. 53-69.

[16] Y, Lu, Industrial Integration-A Literature Review, *Journal of Industrial Integration and Management*, Vol. 1, No. 2, 2016, DOI: 10.1142/S242486221650007X.

[17] W. Shen, Q. Hao, H. Mak, J. Neelamkavil, H. Xie, J. Dickinson, R. Thomas, A. Pardasani and H. Xue, Systems integration and collaboration in architecture, engineering, construction, and facilities management: A review, *Advanced Engineering Informatics*, Vol. 24, 2010, pp. 196–207.

[18] A. Kiitam, A. McLay and T. Pilli, Managing conflict in organisational change, *Int. J. Agile Systems and Management*, Vol. 9, 2016, No. 2, pp.114–134.

[19] J. Sielemann, JT - Solving the "confusio linguarum" in a heterogeneous Multi CAD environment, *JT Day Regensburg*, Sep 22, 2016, ProSTEP iViP e.V.

[20] K. Wen, S. Tan, J. Wang, R. Li and Y. Gao, A model based transformation paradigm for cross-language collaborations, *Advanced Engineering Informatics*, Vol. 27, 2013, pp. 27–37.

[21] Y. Chen, Industrial information integration—A literature review 2006–2015, *Journal of Industrial Information Integration,* Vol. 2, 2016, 30–64.

[22] T. Suomalainen, R. Kuusela and M., Tihinen, Continuous planning: an important aspect of agile and lean development, *Int. J. Agile Systems and Management,* Vol. 8, 2015, No. 2, pp. 132-162.

[23] W.A. Hatem, A. Kwan, and J. Miles, Comparing the effectiveness of face to face and computer mediated collaboration, *Advanced Engineering Informatics*, Vol. 26, 2012, pp. 383–395.

[24] Y. Li, J. Shen, J. Shi, W. Shen, Y. Huang and Y. Xu, Multi-model driven collaborative development platform for service-oriented e-Business systems, *Advanced Engineering Informatics*, Vol. 22, 2008, pp. 328–339.

[25] J.P.T. Mo and W. Lorchirachoonkul, Lifecycle design and support of intelligent web-based service systems, *Int. J. Agile Systems and Management*, Vol. 9, 2016, No. 2, pp.135–153.

[26] J. Stjepandić, H. Liese and A.C. Trappey (2015) 'Intellectual Property Protection', in: Stjepandić, J. et al. (eds.): *Concurrent Engineering in the 21st Century: Foundations, Developments and Challenges*, pp. 521–552, Springer International Publishing Cham.

[27] Y.-T. Chen and M.-C. Chiu, A case-based method for service-oriented value chain and sustainable network design, *Advanced Engineering Informatics*, Vol. 29, 2015, pp. 269–294.

[28] W. Viriyasitavat, Multi-criteria selection for services selection in service workflow, *Journal of Industrial Information Integration*, Vol. 2, 2016, in press.