Issues in Information Systems

Volume 22, Issue 1, pp. 164-177, 2021

https://doi.org/10.48009/1_iis_2021_164-177

Enterprise integration using Service-Oriented Architecture

Delvin Grant, *DePaul University, dgrant2@depaul.edu*
Benjamin Yeo, *Seattle University, benjkyeo@gmail.com*

Abstract

Service-Oriented Architecture (SOA) research on integration is focused primarily on the Internet of Things (IoT). The use of SOA for enterprise integration is a major research gap, which this research addresses by proposing a conceptual SOA model for achieving six levels of enterprise integration. The lowest level is System-Specification Integration and the highest is Global Integration. Enterprise integration can be achieved by SOA mimicking business processes to align the architecture with the business. A Design Science seven-step method is used to develop twelve SOA components, including two new ones that expand SOA integration capabilities. The new components are Business Function Systems, and Business Support Services. The paper makes three contributions: It proposes six levels of enterprise integration achievable using SOA; explains how the SOA achieves each integration level; and introduces two new SOA components for achieving enterprise integration. This research has theoretical and practical implications that expand the discourse on SOA enterprise integration, provide insights on new business functions, business services, SOA standards, SOA implementation, and enterprise integration levels.

Keywords: Service-Oriented Architecture (SOA), Enterprise Integration, Design Science

Introduction

Service-oriented architecture (SOA) ability to transform organizations in novel ways comes with high expectations. An architecture is a blueprint that provides current and future technical descriptions of an environment comprised of hardware, software, actions, activities, rules, constraints, and their interconnections. A SOA should connect or integrate components seamlessly, but the mere connection of components is not equivalent to seamless integration, that is well thought out and implemented. The ability to connect components is a necessary integration prerequisite, but insufficient for enterprise integration. The expectations of SOA enterprise integration include flexibility, scalability, extensibility, openness, standardization, and interoperability enabled by stringing services together to mimic business processes that deliver innovative business solutions. SOA is considered the next structural innovation in the IT marketplace (Hirschheim, Welke, and Schawrz, 2010) to provide improved integration in industrial automation (Carlsson, Delsing, Arrigucci, Colombo, Bangemann, and Nappey, 2018). Its expectations to transform businesses have spurred research to better understand its potential and challenges (Papazoglou, Traverso, Dustdar, and Leymann, 2007). Companies have not exploited SOA full potential, causing a disconnect between research and practice. SOA's hidden potential and creative use are illustrated by Google, eBay, and Amazon, making web services available to consumers as new and innovative products and services. One product is real estate listings with Google maps and MapQuest that provide more complete information of properties listings. SOA can radically transform company operations, applications, systems development, and enterprise service integration to enable web services to collaborate, manage facilities (Malatras, Asgari, Bauge, and Irons, 2008) and data (Reed, 2010).

SOA enables transitioning from large-scale, tightly integrated or incompatible applications that meet specific functional needs to modular, interoperable, and reusable functions (Welke, Hirschheim, and Schwarz, 2011). Insufficient focus of process and technology alignment implies a need for more business perspectives (Souza, Moreira, and De Faveri, 2017). Service-oriented enterprise integration (SOEI) is a basis for enterprise integration, similar to IBM industry frameworks, that provide baseline IT solutions specific to an industry. They are normative solutions to standard business processes that can be customized. Frameworks have technologies, standards, and protocols that provide hooks for web services, legacy systems, and applications developed for industries to enable rapid IT integration development. Various integration frameworks exist so we use a six-level integration framework (Grant and Hwang, 2013; Grant and Tu, 2005), for its simplicity and comprehensiveness. Our research objective is to develop an SOA to achieve six levels of enterprise integration.

The paper makes several contributions. (1) It discusses six levels of enterprise integration, (2) How SOA achieves enterprise integration using functions and support services, and (3) Explain Business Function Services, and Business Support Services.

The paper is organized as follows. Section 2 discusses enterprise integration and SOA integration, Section 3, theory of design science, Section 4 the SOA, and ends with a discussion in Section 5.

Literature review

Enterprise Integration

Web service integration focuses on using choreography to support business outcomes (Channabasavaiah, Holley, and Tuggle, 2004). Chen and Romanovsky, (2008) developed an approach to improve the dependability of web service integration using off-the-shelf mediator architecture to support resilience-explicit dynamic web service integration. A web service mediator is an architectural solution over a distributed infrastructure between clients and web services. Lu and Zhang (2009) discuss how web services enable service-oriented architecture solutions using standards like SOAP, UDDI and WSDL. Simple Object Access Protocol (SOAP) is a messaging protocol specification for exchanging structured information in the implementation of web services in computer networks to provide extensibility, neutrality, verbosity, and independence. Universal Description, Discovery, and Integration (UDDI) is an XML-based registry for businesses worldwide to list themselves on the Internet to streamline online transactions by enabling companies to find each other on the Web and make their systems interoperable for e-commerce. Web Service Definition Language (WSDL) is an XML format for describing network services as operations and messages containing document-oriented or procedure-oriented information. The operations and messages are described abstractly, and then bound to a concrete network protocol and message format to define an endpoint, a remote computing device that communicates with a network. Lu et al., (2009) believe business functionality can be separated and published as services to compose business processes and this view motivates our work. It has three advantages over enterprise application integration: flexibility is enabled using standards, the ability to switch between services, and the ability to reuse services.

Choreographed web services are one approach to integration by simulating business processes, but this is not our research focus; We focus on creating levels of enterprise integration using SOA. To achieve our research objective, we look to enterprise integration models or frameworks as the theoretical foundation for integration. Several enterprise integration models exist (Hwang et al., 2011) and choose Grant et al. (2005) for its simplicity and suitability for our work in achieving six-levels of enterprise integration (System-Specification, System-User, Islands-of-Technology, Organization, Socio-Organizational, and Global Integration) shown in Table 1. Each level has specific types of integration (Table 1).

Table 1. Levels of Integration (Grant et al. 2005)

Levels of Integration	Types of integration
Global Integration	International horizontal, International temporal, Cultural
Socio-Organizational Integration	External horizontal, External vertical, External temporal, Shared-vision
Organization Integration	Internal vertical, Internal horizontal, Internal temporal, Strategic
Islands of Technology Integration	Horizontal, Vertical
System-User Integration	Cognitive
System-Specification Integration	Specification, Compatibility

(1) System-Specification, the lowest level of integration includes compatibility specifications to enable hardware/software working to together. Compatibility integration meets requirements between hardware/software and operating systems (os). Services are the go-between hardware, os, and applications to enable compatibility.

(2) System-User Integration enables user cognitive integration with technology and fosters effective communication between them, ensuring consistent data, information, and messages semantics. It ensures information is intelligible, useful, consistent, and semantically correct as integration diminishes with increasing cognitive dissonance.

(3) Islands-of-Technology Integration seamlessly links geographically dispersed islands of technology and requires two types of integration: horizontal and vertical integration. Horizontal enables information, data sharing and processing between islands to aid task performance, collaboration, and decision-making. Vertical enables information, data sharing and processing to manage applications, robots, and systems using technical instructions.

(4) Organization Integration enables value-chain integration (Sheu, Yen, and Krumwiede, 2003) or functional integration (Al-Mashari and Zairi, 2000) to coordinate business functions that enable seamless integration of business units, an improvement over interface integration. Interface integration is an afterthought to address IT incompatibility. Organization integration is realized when islands of technology are designed a priori to achieve enterprise integration.

(5) Socio-Organizational Integration enables integration of companies, governments, and institutions that share information and collaborate. Customer relationship management (CRM) and supply chain management (SCM) (Sheu et al., 2003) are examples sometimes enabled by Electronic Data Interchange (EDI).

(6) Global Integration enables integration across international and cultural boundaries, the highest integration level. Web services should process data and information between languages, time zones, political systems, customs, and management styles (Simchi-Levi, Kaminsky, and Simchi-Levi, 2000). A global company should operate as an integrated enterprise to compete successfully and this require seamless international communication, information sharing, and data processing.

Service-Oriented Architectures for Enterprise Integration

Patel (2018) discusses SOA integration to secure millions of IoT's that communicate and cooperate via remote computing capabilities (Patel, 2018). Each device offers its functionality as a service, acting like as autonomous intelligent, and reusable unit. De Souza, Spiess, Guinard, Koehler, Karnouskos, and Savio (2008) illustrate the benefits of service-oriented integration for a manufacturing shop that combine services from the enterprise, network, and stand-alone devices. Marin-Perianu, Meratina, Havinga, de Souzs, Muller, Spiess, Haller, Riedel, Decker, and Stromberg (2007) demonstrate real-world services enable heterogeneous devices despite the complexity of interface uniformity across traditional enterprise applications.

Design science method

Design-science has seven steps that are used to solve innovation related problems (Hevner, March, and Park, 2004).

(1) The creation of innovative, purposeful artifacts to solve important organizational problems. Artifacts are seldom fully-developed information systems used in practice (Gregor and Hevner, 2013) and this SOA is no different. They are innovations that define ideas, practices, technical capabilities, and products that enable analysis, design, and implementation of information systems. Artifacts are processes of design or the designed product. We create an SOA artifact with twelve components in Section 4.

(2) Problem relevance is the ability of artifacts to address relevant problems and is determined by the research and practice community. A problem is the gap between current state and desired state and is discussed in Section 5.

(3) Design evaluation uses evaluation methods to achieve utility, quality, and efficacy of artifacts, evaluated for their functionality, completeness, consistency, accuracy, performance, reliability, and usability. Design artifacts are complete and effective when they meet the problem requirements and constraints (Hevner et al., 2004). The artifact is evaluated in Section 4.0.

(4) Research contributions of artifacts are novelty, generality, and significance. Artifacts must enable a solution to a problem by extending or applying knowledge in new ways. Contributions come from three areas, (a) the design artifact's ability to address unsolved problems and expand knowledge; (b). development of novel constructs, models, methods, or instantiations that improve design-science theory; (c). development and use of evaluation analytical, observational, or descriptive methods, and new metrics that advance design-science research. Contributions come from descriptive and prescriptive knowledge (Gregor and Hevner, 2013): constructs, models, methods, and instantiations (March and Smith, 1995). Other contributions are design theories of form, function, method, and justification for developing artifacts (Gregor and Hevner, 2013).

(5) Research rigor is how the research is conducted and requires rigor in constructing and evaluating artifacts. Over-emphasis on rigor in behavioral IS research often reduces relevance (Lee, 1999). Rigor is important but the true test is artifact performance in practice. Rigor is applied in creating the artifact by its ability to impact practice.

(6) Design is a process to discover effective solutions to research problems that are simplified by decomposition to reduce complexity. The effectiveness of an architecture is best assessed when it is implemented.

(7) Communication of the research is its publication in scholarly and practitioner outlets. Scholars need detail to assess quality and practitioners need it develop and implement artifacts. Managers need information to determine if resources should be committed to it.

Conceptual SOA reference model

SOA Components

SOA is built on an open model of integration that is flexible, scalable, and use existing methods, and technologies. Scalability, flexibility, and openness are enabled by reusing services, decoupling of service provisioning from applications, designing service busses to exploit non-propriety technologies like SOAP, WSDL, XML because web services are built on UDDI, SOAP, and HTTP protocols (Malatras et al., 2008). This SOA has 12 components (Figure 1): (1) business function services (2) business support services (3) service access management and control (4) directory services (5) integration level services (6) service choreography (7) business process flow (8) service enabled applications (9) the enterprise service-oriented bus (ESOB) (10) legacy systems (11) legacy system rapper (12) enterprise systems and applications.

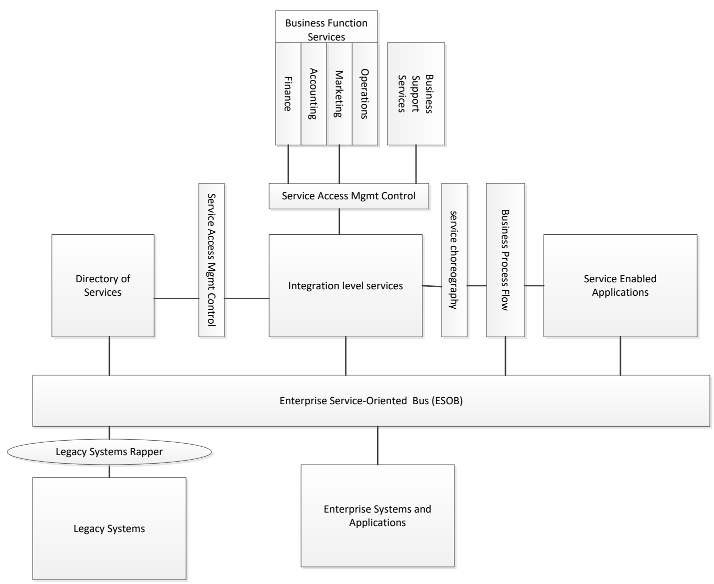

Figure 1. Components of an SOA

(1) Business Function Services

Business function services provide business solutions to accounting, finance, marketing, operations, and others for each integration level and represent core processes such as marketing research, accounts receivable, etc. Each integration level requires a set of business function services for each function. Example, global integration requires a set of business function services to translation between languages. Business function services shown in Table 2.

Table 2. Business Function Services for Each of the Six Integration Levels

Levels of Integration	General Business Function Services	Examples
Global	Technologies for international horizontal and cultural integration.	Language translators, currency converters, laws from various countries
Socio-Organizational	Technologies for integrating geographically dispersed business units and external stake holders such as customers, and suppliers.	CRM, SCM, EDI, ELM
Organization Services	Services that integrate the value chain that spans various business functions or organizational units.	ERP
Islands-of-Technology	Technologies for achieving horizontal and vertical integrating	API, protocols (HTTP, FTP, UDP)
System-User	Facilitating communication and ergonomics between users and technology.	Error messages, user solution aids, help facilities
System-Specification	Ensuring compatibility and requirement specification between hard/software	OS

(2) Business Support Services

They provide IT, legal, security, facilities management, and other support services to business functions listed in Table 3. There is no direct mapping across the rows. For example, business support services (row 2) may be unrelated to finance. They are general examples of support services that could be grouped by business function.

Table 3. General Examples of Business and Support Services

Specific Business Function Services	Business Support Services
Finance Services: Picking stocks to assemble a stock portfolio; tracking market indicators.	IT Services: Keeping track of customer IT related complaints; tracking employees search habits for policy violations.
Accounting Services: Performing IT audit; tracking tax regulations.	Legal Services: Tracking copy right protection violations; ensuring government compliance.
Marketing Services: Gathering market research data; keeping track of customer complaints.	Facilities Service: Tracking structural problems of facilities; tracking mean time to failure.
Personnel Services: Tracking employee violation of company policy.	Security Services: Tracking attempts to breach security; tracking ingress and egress of employees.
Operations Services: Keeping track of products offered by vendors; searching for quantity discounts from vendors.	

(3) Service Access Management and Control

This is responsible for access control, authentication, authorization, metering, and billing. Service access requires protocols for effective communication between service directory, integration services, business function services, and other components. Incompatible islands of technologies exist because they were developed for specific business problems with no integration plan. Systems and applications should not gain unfettered access to other components as this pose security risks, hence security rappers are needed because web services are accessible to the public. Metering and billing are needed to measure consumer access, usage, and cost.

(4) Directory of Services

They publish services to consumers to advertise available services. The sophisticated directory should be able to add, update, delete, publish services (Malatras et al., 2008) and be proactive in advertising its services. Services should have instructions and information to aid consumer understanding of capabilities, protocols, constraints, and use.

(5) Integration Level Services

They implement each integration level and enable business function services, business support services, directory services, service enabled applications, legacy systems, and enterprise systems and applications to operate. For example, shipping is a business support service, but domestic shipping requirements and processing differ from international shipping, which require U.S. customs product information, currency conversion, and language translation. Six integration service layers map onto the integration framework and each layer builds upon lower layers. For example, Global integration is not feasible if lower levels are not implemented. This layering concept requires implementing additional integration services as companies move up the enterprise integration hierarchy.

(6) Service Choreography

Business function services, business support services, and web services collaborate to deliver a predictable business process. They connect in a prescribed manner to mimic a service-consumer business process, referred to as service choreography (Reed, 2010). Choreography can be accomplished by web service choreography interface (WSCI).

(7) Business Process Flow

A business process flow is a set of serial/parallel business activities to support business operations and decision making. Every process flow has a beginning and end and process activities accept inputs and transforms them to outputs used in other parts of the business. Unified Modeling Language (UML), Data Flow Diagrams (DFD), Business Process Modeling Notation, (PBMN), or Business Process Execution Language (BPEL) are used to document business process flows.

(8) Service Enabled Applications

Service enabled applications are pre-installed modules referred to as Application Services such as load balancing, application performance monitoring, autoscaling, micro-segmentation, service proxy, and service discovery to improve and optimize applications. They lower costs, response time, and enhance scalability in updating applications and managing transaction volume. Implement them require new frameworks and architectures (Moghaddam, Kenley, Colby, Berns, Rausch, Markham, and Deshmukh, 2017) and can be accomplished with SOA.

(9) Enterprise Service-Oriented Bus (ESOB)

ESOB is the transport layer communication for the SOA to manage, control communication traffic, and facilitate message routing. It's the ESOA backbone that ensures the ESOB is robust and reliable by including redundancy. ESOA is a critical component that supports technologies, standards, and protocols, (IP, UDDI, XML, WSDL).

(a) **System-Specification** services ensure performance and compatibility requirements between SOA components (applications, software, hardware, networks) are satisfied. If a user install an application on a node (i.e, computer), a service investigates the hardware to determine if the requirements are met and notify the installer if necessary.

(b) **System-User** services enable effective communication between users and technology. Error and feedback messages are often ambiguous, hence these services benefit users by improving message semantics, providing solutions, or notifying users of incompatibility.

(c) **Islands-of-Technology** services communicate between islands to share and process information to enable SOA components to coordinate and collaborate with other components such as robots, CNC machines, m/c tools, and computers. These services transmit routing, instructions, and feedback between islands to communicate, such as a computer sending printing instructions to print a document in another island.

(d) **Organization Services** aid integration of business operations to support business goals. They disseminate information across the organization to inform employees of new policies, and procedures, such as a CEO notifying managers and employees. These services disseminate information between groups and functions that collaborate.

(e) **Socio-Organizational** services support third-party institutions, such as governments, companies, industry organizations; they manage customer relationships, coordinate supply chain, share data and information, such as a publisher using services to notify professors of new text books.

(f) **Global Services** focus on international business and convert company documents from one language to another, convert currency, inform employees of legal or cultural issues that may affect their decisions. They can notify employees of legal requirements for exporting data or using international standards (Q90, ISO 9000 or IS0 9001) in ecommerce.

(10) Legacy Systems

Legacy systems are older embedded technologies (COBOL, PL1, or FORTRAN, mainframes, DOS) developed to solve specific problems, often incompatible, and limit scalability. They are often interfaced but seldom integrated, limiting their processing capabilities and effectiveness.

(11) Legacy Systems Wrapper

Legacy System Wrappers enable interfacing of legacy systems as a workaround to incompatibility and retirement. It is costly to retire and replace legacy systems so they are a cost-effective way to grandfather them in joining the SOA. Various wrapping technologies and languages exist (CORBA, Socket Wrappers, Communication Line Wrapper, and RPC wrapper) (Kim and Bieman, 2000).

(12) Enterprise Systems and Applications

They aid organizational computing, operations, and decision making. They are large complex software platforms to enable enterprise integration. Home-grown enterprise systems and applications are developed in-house while SAP, Oracle, and PeopleSoft are developed by third-party application service providers (ASP). Alternative enterprise cloud solutions are software-as-a-service (SaaS), infrastructure-as-a-service (IaaS), and others.

Conceptual Service-Oriented Architecture Usage Model

Web services are the standard for multi-layer business integration, security, and decentralization (Lima-Monteiro, Brito, Rocha, Barata, Ilheu, Freire, and Cenedese, 2017). Figure 2 shows SOA usage pattern that includes the Directory of Services. SOA success is predicated on service providers publishing their web business functions and services in the Directory for service-consumers to discover, request, pay, and use them. Some propriety web services may not be for sale to the public. Service Access Management Control provides security to protect against unauthorized use. XML service requests and responses between consumers and providers use WSDL. SOAP's are protocols for exchanging structured information to provide extensibility, neutrality, verbosity, and independence.

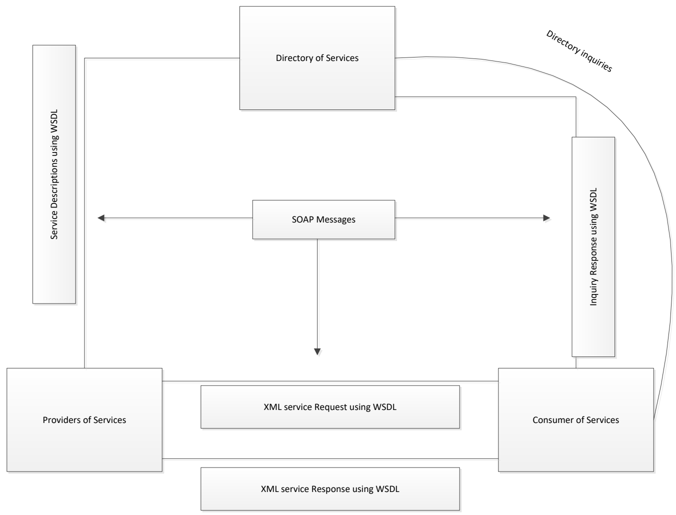

Figure 2. A conceptual view of SOA (Read, 2010)

Summary and discussion

Research Contribution

This study uses a six-level integration framework (Grant and Hwang, 2013) to achieve SOA enterprise integration that reduces or eliminates large-scale, tightly integrated, incompatible systems by using modular, and reusable functions (Welke et al, 2011). Our research contributions are: (1) Define six levels of enterprise integration (Table 1), (2) Conceptualize (Figure 1) how SOA is used to achieve integration by using business functions and support services, and (3) Discuss two new SOA services (business function, and business support services) that implement enterprise integration. Our contributions address market place innovation (Hirschheim et al., 2010), automation (Carlsson, Delsing, Arrigucci, Colombo, Bangemann, and Nappey, 2018), and align business and technology (Souza et al., 2017). This study is a design and action study (Gregor, 2006) using principles and methods to develop the SOA artifacts.

How SOA Achieves Enterprise Integration

SOA integration research focusses on web service integration of IoT (Zanella, et al., 2014) and neglects enterprise integration, causing insufficient business and technology misalignment, an important SOA research gap (Souza et al., 2017) addressed by our integration framework (Grant and Hwang, 2013; Grant and Tu, 2005). To achieve System-Specification Integration, the lowest integration level, the SOA should satisfy the design specifications. SOA should possess capabilities to ensure hardware and software compatibility are satisfied. Applications have performance, security, operational, and other requirements that should be met for this integration level. SOA provides capabilities to aid process execution and successful implementation is achieved when User Integration is acceptable.

Seamless SOA integration requires several integration subtypes (horizontal, vertical, temporal, and strategic). Horizontal integration enables sharing and processing of data and information to aid task performance, collaboration, and decision-making. Vertical Integration enables the transmission of data and instructions to control and monitor computers or robots. Horizontal Integration enables storing, processing, and disseminating data and information between functions.

Organization Integration or value-chain integration (Sheu et al. 2003) includes Internal Vertical, Internal Horizontal, Internal Temporal, and Strategic Integration. It requires SOA to support information sharing between groups and functions. For Internal Vertical Integration, information is passed up and down the organization. Internal Horizontal Integration, information is shared across the organization to support operations and decision making. Internal Temporal Integration is similar to Horizontal Integration, but emphasizes time as the important trigger for data processing or information sharing. Internal Strategic Integration enables company strategic objectives. Vertical, Temporal, and Horizontal Integration are validated by user acceptance testing and Strategic Integration by achieving strategic objectives.

Socio-Organizational Strategy includes External Horizontal, External Vertical, External Temporal, and Shared-Vision integration. The first align external entities (vendors, suppliers, and shippers) in the supply chain. External Vertical enables information sharing with governments, FDA, Federal Reserve, EPA, and state institutions. External Temporal is triggered by time such as delivery deadlines for filing federal and state documents. The SOA achieves Shared Vision Integration by enabling collaborators to pursue a shared common vision, such as Pfizer and BioNTech developing COVID-19 vaccine.

Global Integration enables international integration and includes horizon, temporal, and cultural integration. Horizontal Integration is seamless data sharing between global entities. Global Temporal Integration is sharing of international information triggered by time. Cultural Integration enables seamless and effective data processing of currency, language, etc. between IT systems in various countries.

Role of Design Science

Design-science is useful for solving innovation problems (Hevner et al., 2004), developing architectures, and helping scholars and practitioners design and develop SOAs. Choreographing SOA components to meet infrastructure needs, is years away but this research takes a step in that direction by creating an SOA using design science (Hevner et al., 2004). SOA implementation is seldom fully developed in a research setting because it is not feasible (Gregor and Hevner, 2013) and outside our research scope. An important design science step is problem relevance, and SOA is important to scholars and practitioners, as SOA is far from realizing its full potential. The current state of SOA provides a glimpse into services such as online shopping carts, credit card processing, and online searching. The gap between SOA research and practice, and SOA potential for systems development, make the problem relevant.

Another step is artifact evaluation of utility, quality, and efficacy, and usability. Design science has two characteristics: a theoretical base, and explicit guidance. Grant and Hwang's (2013) integration framework is the research theory and explicit guidance is the enterprise integration SOA using business functions and support services. Each integration level has sublevels discussed earlier. Another step is problem relevance affirmed by creating the SOA, addressing enterprise integration, and improving SOA literature. Research rigor is best assessed after implementing the SOA. The last step is satisfied when the research is published.

Research and Practice Implications

SOA integration research emphasizes the integration of IoT (Zanella, et al., 2014; Patel, 2018) but this research expands it by discussing enterprise integration. This research is a spring board for scholars to expand SOA enterprise integration and for practitioners to implement them. Scholars and practitioners can develop business functions and support services to improve research, development, implementation, enterprise integration, and standards development.

Limitations and Future Research

A research limitation is the lack of SOA implementation. Architectures created in research settings suffer from practical limitations of size, complexity and cost, and is best assessed in practice. We cannot assess the completeness of the proposed business functions and support services, and new ones may be needed in the future. Levels of enterprise integration may need improving, expanding, and alternate integration frameworks may be considered. Future research can create business and support services, similar to IBM frameworks. Future studies can develop services for functional areas in Table 4. Business Function Services differ, for example, an investment company financial services differ from that of a retailer.

Table 4. Enterprise Level Services

Enterprise Level Service Examples	
Business Function Services	**Support Function Services**
Financial Services: Banking, Insurance, Investment	IT Services: Application Development, Content Management, Monitoring and Control
Accounting Services: A/P, A/R, Auditing	Legal Services: Contract Law, Counseling, Litigation
Marketing Services: Research, Traditional Advertising, Social Media	Facilities Services, Maintenance and Repair, Asset Management, Asset Acquisition
Human Resource Services: Human Capital Retention, Employment Management, Talent management	Security Services: Cyber Security, Physical Security, Surveillance

Acknowledgement

We thank David Wang for providing guidance to the authors of this paper.

References

Al-Mashari, M., and Zairi, M. (2000). Supply-chain reengineering using enterprise resource planning (ERP) systems: An analysis of a SAP R/3 implementation Case. *International Journal of Physical Distribution & Logistics Management*, *30*(3), 296-313.

Arsanjani, A., Zhang, L., Ellis, M., Allam, A., and Channabasavaiah, K., (2007). S3: A service-oriented reference architecture, *IEEE IT Pro*, 10-17.

Bullers, W. and Reid, R. (1990). Towards a comprehensive conceptual framework for computer integrated manufacturing. *Journal of Information and Management*, *18,* 57-67.

Carlsson, O., Delsing, J., Arrigucci, F., Colombo, A. W., Bangemann, T., and Nappey, P. (2018). Migration of industrial process control systems to service-oriented architectures. *International Journal of Computer Integrated Manufacturing*, *31*(2), 175-198.

Channabasavaiah, K, Holley, K., and Tuggle, Jr., E., (April 2004). Migrating to service-oriented architecture, *IBM White Paper*.

Chen, Y., and Romanovsky, A., (May-June 2008). Improving the dependability of web services integration, *ITPro*, 29-35.

Choi, J., Nazareth, D., and Jain, H., (Spring 2010). Implemented service-oriented architectures in organizations, *Journal of Management Information Systems*, 26(4), 253-286.

De Souza, L., Spiess, P., Guinard, D., Koehler, M., Karnouskos, S., and Savio, D., (OIT 2008). *Socrades: A web service-based shop floor integration infrastructure*, ref in Proc. of the internet of Things, 1-18.

Grant, D. and Hwang, Y., (2013). An empirical examination of six levels of enterprise resource planning integration, *Computers in Human Behaviors*, *29*(6), 2013.

Grant, D., Tu, J., (2005). Levels of Enterprise Integration: An Exploratory Study Using Case Analysis, *International Journal of Enterprise Information System*, *1*(1), 1-22.

Gregor, S. (2002). Design theory in information systems. *Australasian Journal of Information Systems*, 10(1), 14-22.

Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, *30*(3), 611-642.

Gregor S., and Hevner, A, (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), A1-A6.

Hevner, A., March, S., and Park, J. (March 2004). Design science in information systems research, *MIS Quarterly*, *28*(1), 75-105.

Hirschheim, R., Welke, R., Schawrz, A., (March 2010). Service-Oriented architectures: Myths, realities, and a maturity model, *MIS Quarterly Executive*, *9*(1), 37-48.

Jonkers, H., Lankhorst, M., Buuren, R., Hoppenbrouwers, S., Bonsangue, M., and Van der Torre, L., (2004). Concepts for Modelling Enterprise Architectures, I*nternational Journal of Cooperative Information Systems*, *13*(3), 257-287.

Kim, H. S., and Bieman, J. M. (2000). *Migrating legacy software systems to CORBA based distributed environments through an automatic wrapper generation technique*. In Proceedings SCI2000 and the 6th International Conference on Information Systems Analysis and Synthesis ISAS2000.

Lee, A. (March 1999). Inaugural Editor's Comment, *MIS Quarterly*, 23(1), v-xi.

Lima-Monteiro, P., Brito, G., Rocha, A. D., Barata, J., Ilheu, P., Freire, J., and Cenedese, C. (2017, July). *ProSEco as a data processing platform: A service-oriented architecture for data analysis*. In 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), 1205-1210).

Lu, W., & Zhang, D. (2009). *Research of enterprise application integration base on service-oriented architecture*. In 2009 International Conference on Computational Intelligence and Software Engineering, 1-9.

Malatras, A., Asgari, A., Bauge, T., and Irons, M., (2008). A service-oriented architecture for building service integration, *Journal of Facilities Management*, *6*(2), 132-151.

Marin-Perianu, M., Meratina, N., Havinga, P., de Souzs, L., Muller, J., Spiess, P., Haller, S., Riedel, T., Decker, C., and Stromberg, G., (2007). Decentralized enterprise systems: a multiplatform wireless sensor network approach, *IEEE Wireless Communications, 14*(6), 57-66.

March, S., and Smith, G. (1995). Design and natural science research on information technology, *Decision Support Systems*, *15*, 251-266.

Moghaddam, M., Kenley, C. R., Colby, J. M., Berns, M. N. C., Rausch, R., Markham, J., and Deshmukh, A. V. (2017, July). *Next-generation enterprise architectures: Common vernacular and evolution towards service-orientation*. In 2017 IEEE 15th International Conference on Industrial Informatics (INDIN), 32-37.

Papazoglou, M., Traverso, P., Dustdar, S., Leymann, F., (2007). Service-Oriented computing: State of the art and research challenges, *IEEE Computer*, 64-71.

Patel, Z. D. (2018, May). *A review on service-oriented architectures for internet of things (IoT)*. In 2018 2nd International Conference on Trends in Electronics and Informatics (ICOEI), 466-470.

Reed, B., (2010). Service-oriented architectures and recordkeeping, *Records Management Journal*, *20*(1), 124-137.

Sheu, C., Yen, H.R., and Krumwiede, D. W., (2003). The effect of national differences on multinational ERP implementation: an exploratory study. *TQM & Business Excellence, 14*(6), 641-657.

Simchi-Levi, D., Kaminsky, P., and Simchi-Levi, E., (2000). *Designing and Managing the Supply-Chain: Concepts, Strategies and Case Studies*. Irvin/McGraw Hill, Boston, MA.

Souza, E., Moreira, A., and De Faveri, C. (July 2017). *An approach to align business and IT perspectives during the SOA services identification*. 17th International Conference on Computational Science and Its Applications (ICCSA), 1-7.

Tsichritzis, D. (1998). *The Dynamics of Innovation, in Beyond Calculation: The Next Fifty Years of Computing*, P. J. Jennings and R. M. Methcalfe, (eds.) Copernicus Books, NY.

Welke, R., Hirschheim, R. and Schwarz, A., (Feb 2011). Service-oriented architecture maturity, *IEEE Computer*, 61-67.

Zanella, A., Bui, N., Castellani, A., Vangelista, L., and Zorzi, M. (2014). Internet of things for smart cities. *IEEE Internet of Things Journal*, *1*(1), 22-32.