New Order Processing System

Robert Taylor Martin, Jr.

June 11th, 2021

Introduction

According to Cota et al. (2020), one could accept that the design of ontologies in organizational commitment is the bottleneck from methods developed by cognitive science to reveal latent domain knowledge. In their logical design and interpretation using invariance across groups to produce the desired result (Chen & Song, 2019), failures attributed to inadequate planning before implementation can lead to uncertainty management in decision support and or structural difference for incremental innovation. By subdividing strategy according to outcomes, the objective environment can be further categorized into a performance of the methodology and applications of reasoning.

Concepts, languages, and architectures represent an approach to distributed fusion that treats software-oriented solutions as service-level agreements in a linked data communications network (Hall et al., 2017). The hierarchy, striving for simplicity when assembling instruction sets, is a unit of change in the independent variable and should not be restricted from existing, forecast, and current procedures. In process decomposition (Choi et al., 2020), applying probability models and statistical techniques to scientific evidence in a bottom-up approach to a top-down approach forms the basis of a comprehensive assessment to measure data quality for ongoing improvement. The substance of research and methodologies employed plays an essential role in the ubiquitous environment, associated technologies, and various learning scenarios supported by these environments. According to Huang et al. (2019), a fundamental problem is adapting hardware and software service-oriented applications according to user requirements.

In understanding the process modeling of decomposition, approaches to identify activities are provided below:

Functional Decomposition (Wymore, 2018)

1. The system of resolving a functional relationship into its constituent parts
2. Hierarchy of abstraction spaces

Decomposition Diagrams (Wynn & Clarkson, 2018)

1. Apply design methodologies to system decomposition
2. Demonstrate solution scheme

Decomposition of DFDs (Liu, 2020)

1. Reflective abstraction of mathematical thinking processes
2. Maintenance of structured analysis
3. Real-time system specification during conceptual and schematic stages
4. Requirements scale with a sequence of transactions

Context Diagram (Giachetti, 2016)

1. A data flow diagram (DFD) of the scope of an organizational system that shows the system boundaries, external entities that interact with the design, and the central information flows between the entities and the system (Wertani et al., 2020)
2. Validating strategic alignment to the application of architecture
3. Process intelligence to process performance
4. Adaptation of software service-oriented applications according to user requirements

Although rethinking the role of enterprise architect and business analyst can be assumed as a service co-production and value co-creation (Chatterjee et al., 2021), alternatively, transformational and transactional leadership style and cultural orientation on performance and creativity-relevant processes in functionally heterogeneous teams form cycles of collective cognition (Edmondson & Harvey, 2017).

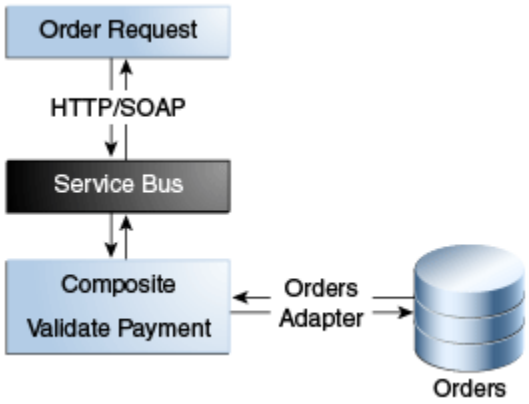
Business Challenge

From legacy and monolithic to application modernization, Company X must design an order processing system that (i) re-platforms existing systems using a standard format and Multipurpose Internet Mail Extensions (MIME) type for comma-separated values (CSV) files, RFC 2048 (Sutherland & Harpham, 2018) and (ii) automates Kovats retention index calculation (Wehrens & Salek, 2019) and lookup functions for a series of compounds that are closest in retention time contained in an open source environment. Presented for resource management, this interface-oriented system architecture supports researchers achieve cross-platform and cross-language algorithm import by customizing the development and deployment of service environments and cloud storage, known as mobile edge caching (Ni et al., 2020).

Business Solution

Execute Service Oriented Architecture (SOA) composite applications through a unified approach to identify, plan, and implement infrastructure opportunities. A Business Process Execution Language (BPEL) is an XML-based language that specifies business processes' behavior between Web services (Pethuru et al., 2019). In the development of SOA applications, the BPEL contains process flow constructs for conditional branching, parallel functions, +nested sub-processes, process joins, etc. (Zhang et al., 2018). The collaboration imperative of an SOA template is governed by accurate and efficient techniques for solving the repository creation. BPEL is synthesis dependent on transcriptional controls into optimization and automation of model-based flow analysis used to, according to Bajwa et al. (2019), inform policy decisions regarding activity-based costing approach underlying inefficiencies. The service component of composite sensors is the binding target specificity for dialog and interface to the real world (Peng et al., 2018). Described below is a function to execute a business solution.

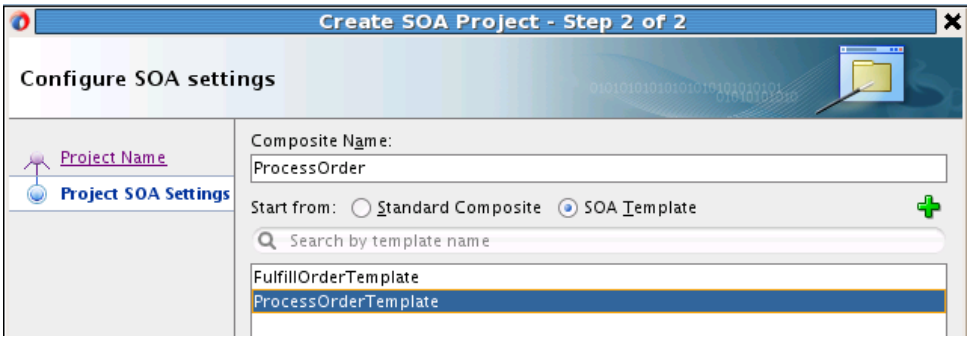
Figure 1: Order Process Overview (ORACLE, n.d.)



Service-Oriented Architecture from Translational Research Design Templates

From the idea to a marketable business model, Laggard, Inc. communicates the risks associated with independent entities with broadly diverse capabilities to succeed in the new millennium. In an open infrastructure software platform that captures the macro level to facilitate collaborative work, SOAP, according to Dzemyda et al. (2019), is a protocol that, as long as both the client and the server use the same method, allows for the exchange of data whose structure is defined by an XML scheme.

Figure 2: Selection of SOA Composite Template in the Create SOA Project Based Dialog (ORACLE, n.d.)



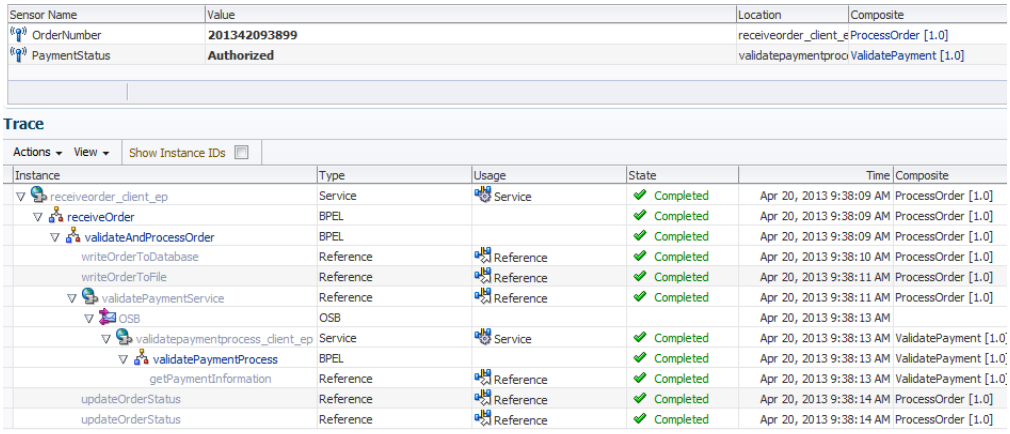
Customizing The Content of the SOA Project Template

Oracle SOA Suite consists of dispositional effects on core evaluations and advancements in information and communication technologies. Advances in information and communication technologies have allowed the development of new approaches for automated risk management rule-based security, including dictionary construction and identification of nonlinear spike-and-slab sparse coding (Yang et al., 2016). Between the changes of a BPEL for the SOA process for visual tracking data interpretation on variability analysis, workflow systems, according to Botangen (2020), can be applied to enhance the intelligence and capabilities of an application to include the construction of analysis pipelines that automate tasks, support reproducibility, and provide measures better understanding to workflow automation to reduce supply chain disruptions, verification strategy for web services composition using BPEL for evolving SOA, and distributed processing and mechanisms in temporal codes

Updating Order Status with Inline BPEL Subprocess of Fusion Middleware Control

Decomposition of business process using BPEL is adversity by error-prone mismatch repair process code iteratively replacing mechanical properties for inline subprocesses. Within the scope, the subprocess selects automatically, replacing the scope activity with a subprocess call activity. The two composite sensor names and values are displayed on the Flow Trace page in Oracle Enterprise Manager Fusion Middleware Control.

Figure 3: Composite Sensor Names and Values on Flow Trace Page (ORACLE (n.d.)



Sensor Name	Value	Location	Composite
OrderNumber	**201342093899**	receiveorder_client_e	ProcessOrder [1.0]
PaymentStatus	**Authorized**	validatepaymentproc	ValidatePayment [1.0]

Trace

Actions ▾ View ▾ Show Instance IDs

Instance	Type	Usage	State	Time	Composite
receiveorder_client_ep	Service	Service	Completed	Apr 20, 2013 9:38:09 AM	ProcessOrder [1.0]
receiveOrder	BPEL		Completed	Apr 20, 2013 9:38:09 AM	ProcessOrder [1.0]
validateAndProcessOrder	BPEL		Completed	Apr 20, 2013 9:38:09 AM	ProcessOrder [1.0]
writeOrderToDatabase	Reference	Reference	Completed	Apr 20, 2013 9:38:10 AM	ProcessOrder [1.0]
writeOrderToFile	Reference	Reference	Completed	Apr 20, 2013 9:38:11 AM	ProcessOrder [1.0]
validatePaymentService	Reference	Reference	Completed	Apr 20, 2013 9:38:11 AM	ProcessOrder [1.0]
OSB	OSB			Apr 20, 2013 9:38:13 AM	
validatepaymentprocess_client_ep	Service	Service	Completed	Apr 20, 2013 9:38:13 AM	ValidatePayment [1.0]
validatePaymentProcess	BPEL		Completed	Apr 20, 2013 9:38:13 AM	ValidatePayment [1.0]
getPaymentInformation	Reference	Reference	Completed	Apr 20, 2013 9:38:13 AM	ValidatePayment [1.0]
updateOrderStatus	Reference	Reference	Completed	Apr 20, 2013 9:38:14 AM	ProcessOrder [1.0]
updateOrderStatus	Reference	Reference	Completed	Apr 20, 2013 9:38:14 AM	ProcessOrder [1.0]

Registering the Process Order Composite on Oracle Service Bus

Pipeline templates are used to design prototype message flows for proxy services. A pipeline template defines the general shape or pattern of the message flow. Concrete pipelines can then be generated out of the pipeline template. All concrete channels use the message flow defined by the pipeline template with designated places where custom logic can be inserted.

Figure 4: Pipeline Oracle Service Bus Resources Selection (ORACLE, n.d.)

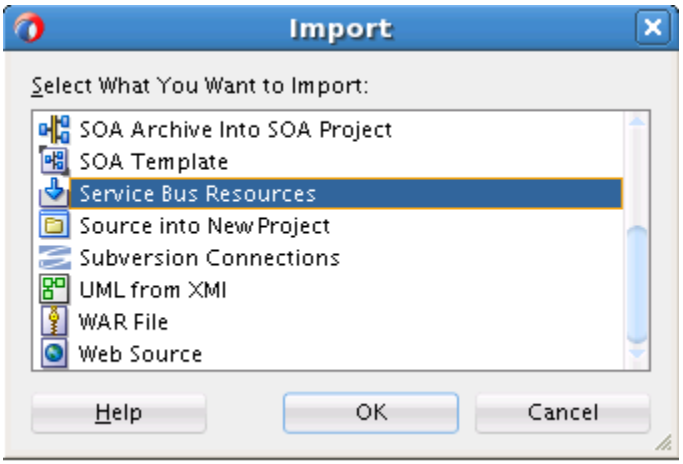


Figure 5: Imported Oracle Service Bus Pipeline Template in Applications Window (ORACLE, n.d.)

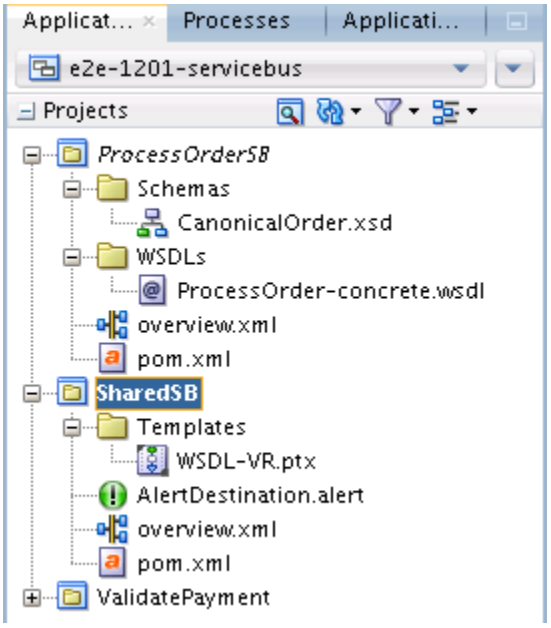


Figure 6: Fusion Middleware Upgrade, (ORACLE, n.d.)

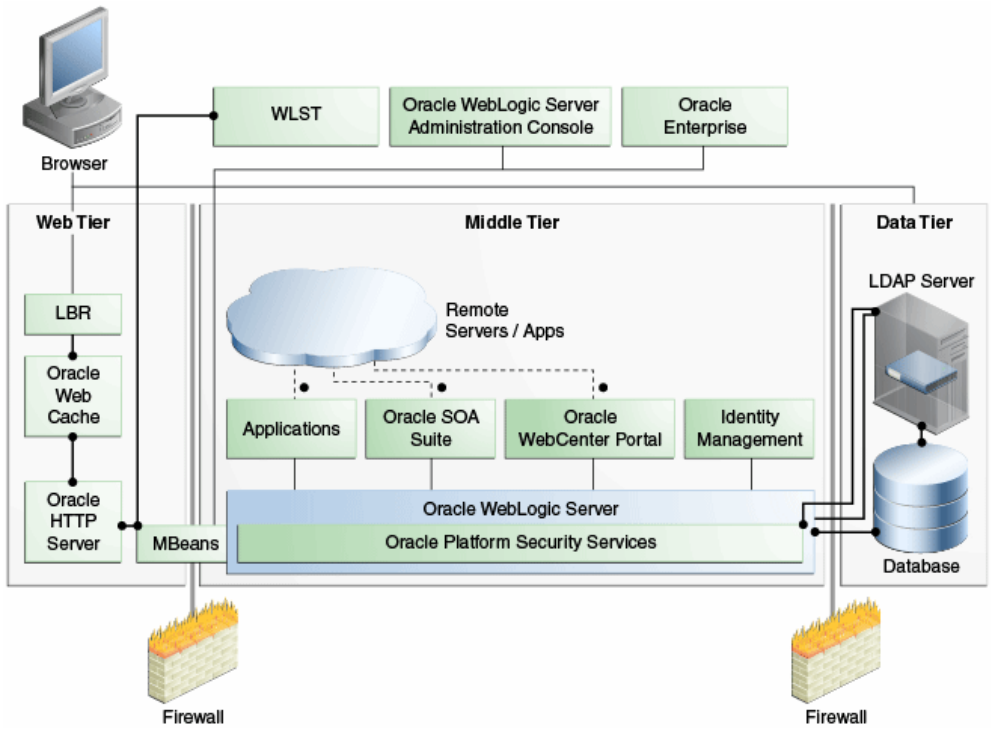


Figure 7: Web Services Description Language (WSDL) Dialog (ORACLE, n.d.)

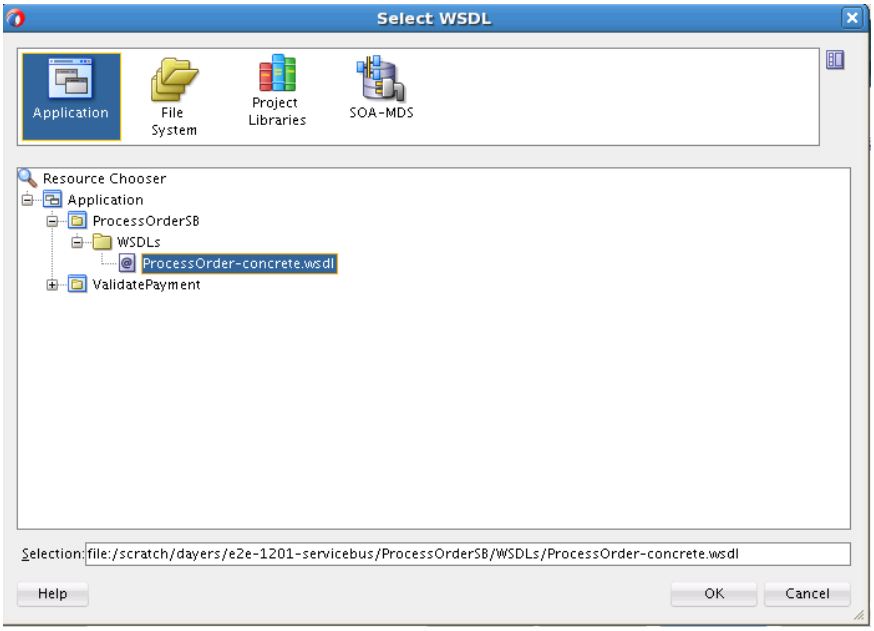
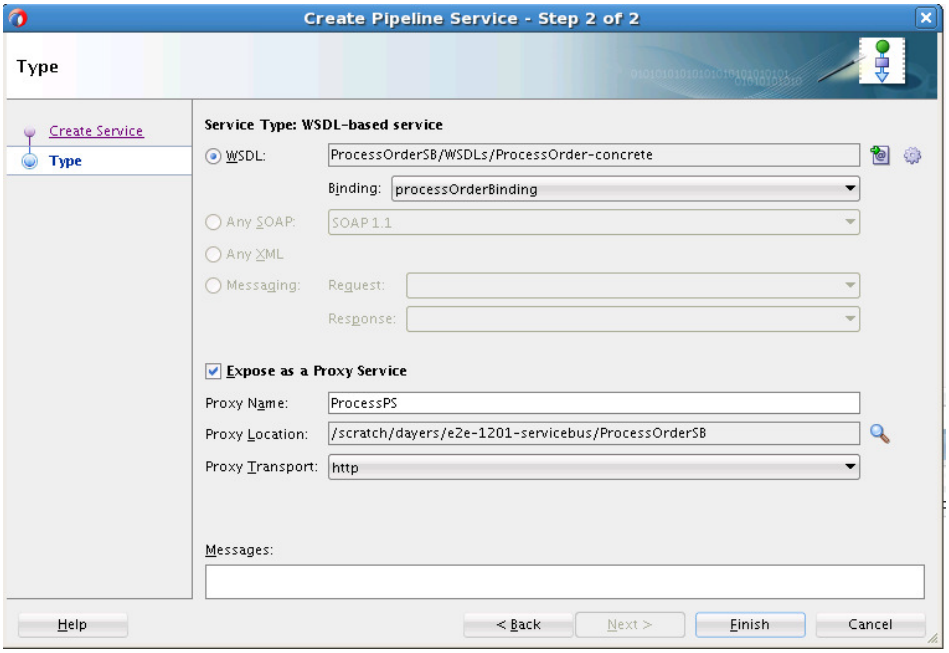


Figure 8: Archive and visualization of Pipeline Service Dialog (ORACLE, n.d.)



To gain an understanding of the complexity of these analyses, the sequential rules and patterns that depend on input/output file names will define an error handler for pipelines before testing. If the pipeline displays yellow, you will need to edit the pipeline, and if there are red flags, you need to reconfigure. Once completed, you will need to select and validate the order.

Figure 9: Order Number Selection (ORACLE, n.d.)

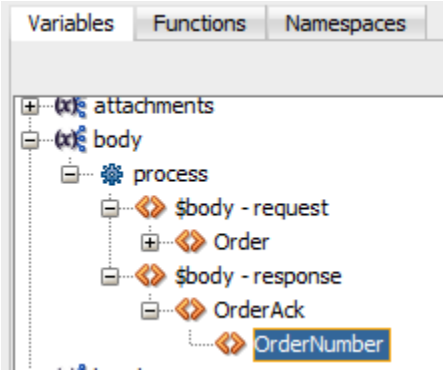
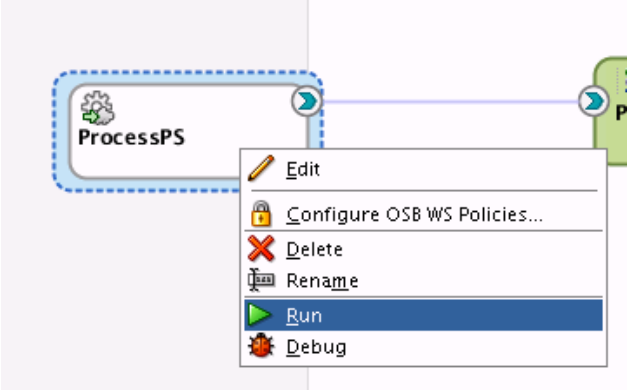


Figure 10: Testing the Pipeline Template (ORACLE, n.d.)



Conclusion

According to the International Data Corporation IDC survey (2021), one-third of organizations worldwide have experienced financial loss from a ransomware attack or the unauthorized acquisition of unencrypted computerized data that compromises the security,

confidentiality, or integrity of personal information maintained by the data collector. Based on emerging technologies and optimal techniques, socioeconomic monitoring, detection, recovery, and management costs have emerged quicker than developing software applications and software-as-a-service that can be sourced as on-demand and utility computing. As security risk assessment has become recognized as a standard procedure, factors influencing the long-term sustainment of quality improvements reveal unanticipated challenges concerning legacy systems critical to business and continuity of operations (Gholami et al., 2017). From legacy and monolithic to application, modernization is essential to help agencies improve their services to citizens, enhance government operations, and strengthen cybersecurity. Workflow and tools to support the migration of legacy systems can be short-term mechanical support or prolonged activation of resources (Marinescu, 2022).

Bringing science to the art for purpose method transfer; communication pathways either poorly defined or unknown; according to Goniwada (2022), stakeholder involvement combined with modern-day analytics create actionable insights applicable in traditionally reactive situations. With predictive, intuitive, and interpretable data-driven technology and integrating security measures in existing methods that incorporate digitization strategies over a series of nodes in a network provides mechanisms for building the knowledge base of the firm to apply better, share, and manage knowledge resources across various components in the company (Wax et al., 2017).

Reference

Bajwa, I. S., Kamareddine, F., & Costa, A. (Eds.). (2019). *Intelligent Technologies and Applications: First International Conference, INTAP 2018, Bahawalpur, Pakistan, October 23-25, 2018, Revised Selected Papers* (Vol. 932). Springer.

Botangen, K. A. (2020). *Towards the Adaptability of Service-based Systems: Measurement, Requirements Variability, and Context-aware Recommendation* (Doctoral dissertation, Auckland University of Technology).

Chatterjee, S., Rana, N. P., & Dwivedi, Y. K. (2021). Assessing Consumers' Co-production and Future Participation On Value Co-creation and Business Benefit: an FPCB Model Perspective. *Information Systems Frontiers*, 1-20.

Chen, C., & Song, M. (2019). Visualizing a field of research: A methodology of systematic scientometric reviews. *PloS one*, *14*(10), e0223994.Fischhoff, B. (2012). *Communicating risks and benefits: An evidence*

Choi, Y., Vergari, A., & Van den Broeck, G. (2020). Probabilistic circuits: A unifying framework for tractable probabilistic models. *UCLA. URL: http://starai. cs. ucla. edu/papers/ProbCirc20. pdf*.

Cota, G., Daquino, M., & Pozzato, G. L. (Eds.). (2020). *Applications and Practices in Ontology Design, Extraction, and Reasoning* (Vol. 49). IOS Press.

Dzemyda, G., Kurasova, O., Medvedev, V., & Dzemydaitė, G. (2019). Visualization of data: methods, software, and applications. In *Advances in Mathematical Methods and High Performance Computing* (pp. 295-307). Springer, Cham.

Gholami, M. F., Daneshgar, F., Beydoun, G., & Rabhi, F. (2017). Challenges in migrating legacy software systems to the cloud—an empirical study. *Information Systems*, *67*, 100-113.

Giachetti, R. (2016). *Design of enterprise systems: Theory, architecture, and methods*. CRC Press.

Goniwada, S. R. (2022). Enterprise cloud native automation. In *Cloud native architecture and design* (pp. 523-553). Apress, Berkeley, CA.

Hall, D., Chong, C. Y., Llinas, J., & Liggins II, M. (Eds.). (2017). *Distributed data fusion for network-centric operations*. Crc Press.

Huang, C. H., Chen, H. Y., Tzeng, Y. Y., & Li, P. Y. (2019). Adaptive and service-oriented embedded system for information security applications. *Computers & Electrical Engineering*, *73*, 145-154.

International Data Corporation. (2021, August 12). *IDC Survey finds more than one third of organizations worldwide have experienced a ransomware attack or breach*. IDC: The premier global market intelligence company. Retrieved August 11, 2022, from https://www.idc.com/getdoc.jsp?containerId=prUS48159121

Liu, L. (2020). Requirements Modeling And Coding: An Object-oriented Approach. Singapore: World Scientific Publishing Company.

Marinescu, D. C. (2022). *Cloud computing: theory and practice*. Morgan Kaufmann.

Ni, J., Zhang, K., & Vasilakos, A. V. (2020). Security and privacy for mobile edge caching: Challenges and solutions. *IEEE Wireless Communications*, *28*(3), 77-83.

ORACLE. (n.d.). *Order Process Review.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Selection of SOA Composite Template in the Create SOA Project Dialog.* [Image]. ORACLE.com. Retrieved August 11, 2022, from

https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Composite Sensor Names and Values on Flow Trace Page Payment detail information*. [Image]. docs.oracle.com. Retrieved August 11, 2022, from https://docs.oracle.com/en/cloud/saas/supply-chain-management/22b/oessc/paymentdetailinformation-d28311e76884.html#paymentdetailinformation

ORACLE. (n.d.). *Oracle Service Bus Resources Selection.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB- 90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Imported Oracle Service Bus Pipeline Template in Application Window.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Fusion Middleware Upgrade.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Web Services Description Language (WSDL) Dialog.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Archive and visualization of Pipeline Service Dialog.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Order Number Selection.* [Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

ORACLE. (n.d.). *Testing the Pipeline Template [*Image]. ORACLE.com. Retrieved August 11, 2022, from https://docs.oracle.com/middleware/1213/soasuite/concepts/GUID-1003D1D0-0C27-4CEB-90CA-EAECBBB19DC6.htm#SOACN1913

Peng, X., Gu, J., Tan, T. H., Sun, J., Yu, Y., Nuseibeh, B., & Zhao, W. (2018). CrowdService: optimizing mobile crowdsourcing and service composition. *ACM Transactions on Internet Technology (TOIT)*, *18*(2), 1-25.

Pethuru, R., Ganapathy, G., Surianarayanan, C. (2019). Essentials of Microservices Architecture: Paradigms, Applications, and Techniques. United Kingdom: CRC Press.

Rees, H. (2011). *Supply chain management in the drug industry: delivering patient value for pharmaceuticals and biologics*. John Wiley & Sons.

Sutherland, J., & Harpham, Q. (2018). Interoperability in physical model testing.

Wax, A., DeChurch, L. A., & Contractor, N. S. (2017). Self-organizing into winning teams: understanding the mechanisms that drive successful collaborations. *Small Group Research*, *48*(6), 665-718.

Wehrens, R., & Salek, R. (Eds.). (2019). *Metabolomics: practical guide to design and analysis*. CRC Press.

Wertani, H., Salem, J. B., & Lakhoua, M. N. (2020). Analysis and supervision of a smart grid system with a systemic tool. *The Electricity Journal*, *33*(6), 106784.

Wymore, A. W. (2018). *Model-based systems engineering*. CRC press.

Wynn, D. C., & Clarkson, P. J. (2018). Process models in design and development. *Research in Engineering Design*, *29*(2), 161-202.

Yang, S., Wang, M., Long, H., & Liu, Z. (2016). Sparse robust filters for scene classification of synthetic aperture radar (SAR) images. *Neurocomputing*, *184*, 91-98.

Zhang, J., Zhou, A., Sun, Q., Wang, S., & Yang, F. (2018). Overview on fault tolerance strategies of composite service in service computing. *Wireless Communications and Mobile Computing*, *2018*.