International Journal of Computer Trends and Technology
ISSN: 2231–2803 / https://doi.org/10.14445/22312803/IJCTT-V72I8P122

Volume 72 Issue 8, 147-154, August 2024

Original Article

Towards a Decision Support Tool for Service Identification in Service-Oriented Architecture

Iheb Abdellatif[1], Amar Ramudhin[2]

[1]*Department of MISA, State University of New York, New York, USA.*
[2]*Department of ISEM, Harrisburg University of Science and Technology, New York, USA.*

[1]*Corresponding Author : iabde002@plattsburgh.edu*

Received: 27 June 2024 Revised: 30 July 2024 Accepted: 14 August 2024 Published: 31 August 2024

Abstract *- Given the turbulent environment, the demands exerted by customers, and the pressures from competitors, companies express more and more the need for an information architecture that can adapt and respond quickly to their needs at lower costs. A Service-Oriented Architecture (SOA) is the leading choice for ensuring a flexible information system. Despite the current hype around SOA, there is no standard approach to setting up this new kind of architecture. However, a consensus is that a successful SOA is based on well-identified services. This paper introduces a systematic business process-based approach to assist SOA analysts in identifying the most convenient services for their enterprise. A case study illustrates the approach.*

Keywords - *SOA, Service identification, Business processes, UMN, BPEL.*

1. Introduction

SOA is a novel approach to creating flexible information systems that ensure intrinsic interoperability and increase enterprise agility. In addition, SOA promises a complete alignment of business processes and IT solutions, which is seldom realized with traditional information systems. This alignment is crucial as it allows for a more dynamic response to changing market conditions and customer demands, thereby facilitating a more agile and responsive business model.

The implementation of SOA presents new challenges to the enterprise as the SOA lifecycle is entirely different from the traditional IT solutions, and there is no standard approach to setting up such a novel architecture. A successful SOA deployment requires a departure from the monolithic designs of the past, favoring instead a modular approach where services are developed, deployed, and maintained independently.

There is, however, a standard set of principles to which services in SOA must adhere [1]. Some of these basic principles are service reusability, which aims to maximize the utility of service across different contexts and applications; service loosely coupling, which seeks to minimize dependencies to enhance service robustness and ease of integration; service composability, which allows services to be assembled into composite applications; service autonomy, ensuring that services have control over their environment; and service discoverability, which enhances the ability to find and bind to services at runtime [2]. These principles must be respected throughout the SOA lifecycle, starting with the first phase, namely, service-oriented analysis, in which the candidate services of the future SOA and their associated layers are identified.

Service identification can be considered the most crucial step in the SOA lifecycle [3]. It involves determining what services are required to be built and what logic should be encapsulated in each service. Service identification can be made from diverse sources of information, and this diversity of sources can be misleading to the SOA analyst. For instance, analysts may derive service candidates from business process models, existing system functionalities, or strategic business goals, leading to varied and potentially conflicting service definitions.

The challenge is determining an approach that best assists analysts in systematically searching and analyzing this vast amount of information. The approach should also contain guidelines and best practices for service identification to accelerate the identification process, minimize human interventions, reduce the risks of errors, and provide pertinent results. It must encompass mechanisms for effectively capturing and evaluating the complex interplay between business processes, the available IT infrastructure, and the organization's strategic objectives. Moreover, it must leverage existing assets wherever possible, promoting cost-efficiency and resource optimization.

The paper presents such an approach. The methodology introduced is a multi-step, semi-automatic approach that assists analysts in identifying candidate services for SOA. The starting point for this methodology is models describing the enterprise's business processes. These models are used not only as a blueprint for service identification but also as a communication tool to bridge the gap between business and IT, ensuring that the services identified align with business needs and are technically feasible to implement. The methodology leverages modern principles of service design, including domain-driven design and model-driven architecture, to establish a robust framework for SOA service identification and definition.

2. Problem Statement and Related Work

Numerous valuable initiatives for building SOA reflect organizations' diverse methodologies to achieve service orientation. While some methodologies take a holistic approach, addressing all phases of the SOA lifecycle [1], others concentrate on specific phases like service analysis, where the aptness and context of services are scrutinized. The varied approaches imply a spectrum of service hierarchies and analytical inputs, indicative of the SOA's multifaceted nature and adaptability to different enterprise requirements.
The literature provides extensive comparisons of service identification methods, illustrating the evolution and refinement of these methodologies over time [3, 4]. Senthil et al. [5] leverage interface design specifications, capitalizing on the concrete representations of user-system interactions to extrapolate service boundaries. Contrasting with this approach, DongSu, Chee-yang, and Doo-Kwon [6], as well as Yousef et al. [7], advocate for an ontology-based methodology. This ontological emphasis harnesses the power of semantic relationships within business process models, aiming to derive more meaningful and contextually aware service models.

Another significant contribution by Klose, Knackstedt, and Beverungen [4] delineates a phased approach to service identification involving preparation, analysis, and categorization. This method underscores the pivotal role of stakeholders, integrating their insights directly into the service development lifecycle—a testament to the participatory nature of effective SOA initiatives.

Despite these advancements, many methodologies must be revised, necessitating considerable human intervention. The reason often cited is the graphical representation of business processes, which, although intuitive for human understanding, poses a challenge for automated processing. Traditional Business Process Modeling (BPM) notations like EPC and BPMN are primarily visual and require manual translation to be operationalized within an SOA framework.

Arsanjani et al. [8] address this by utilizing the Service-Oriented Modeling and Architecture (SOMA) methodology applied to BPMN models, while Dwivedi and Kulkarni [9] introduce structured modeling that constructs a hierarchical representation of business processes, delineated in UML and serialized in XMI. This serialization facilitates automated processing and statistical analysis, making it a stepping stone towards semi-automated service identification.

Azevedo et al. [10] contribute to this progression with a systematic heuristic approach that combines semantic and syntactic analyses—probing deeper into the intent and structure embedded within process models. However, the abstraction inherent in these methodologies sometimes distances them from practical applicability, creating a gap between theoretical models and operational execution. This gap is evident in literature that provides guidelines and best practices without concrete implementation details [3, 4, 8, 11-13].

In contrast, our approach resonates with the previously discussed systematic and semi-automatic methods. Still, it diverges by eschewing structural pattern analysis in favor of a statistically driven model, as noted by Dwivedi and Kulkarni [9]. Our novel contribution lies in harnessing statistical patterns to inform service identification, steering away from purely theoretical constructs and closer to a data-driven, empirical foundation. This statistical approach quantifies the relevance and utility of potential services and introduces a measure of predictability and reliability into the service identification process.

In summary, our methodology embraces the systematic rigor of top-down approaches while integrating the agility of semi-automatic techniques. By focusing on the statistical relevance of business process components, we offer a pragmatic and robust framework for service identification, aligning theoretical principles with real-world applicability and setting the stage for a more efficient and responsive SOA lifecycle.

3. Proposed Approach

This section presents a six-step (Figure 1) procedure to identify services for all levels of granularity, from business process services to application services. UMN (*Unified Modeling Notation*) business process models are used as input [14-20]. These models are analyzed automatically in a preparation phase to select the business activities and the information that will be used to identify the appropriate services.

The proposed approach begins by identifying the entity services and the application services. It then moves to higher-level services and finally develops the orchestration layer. This method is heavily based on the analysis of 'information' used in each business process activity. The term '*information*' here means any information, be it paper or electronic-based (parts or whole documents, reports, windows, screens, etc.) required to execute an activity.

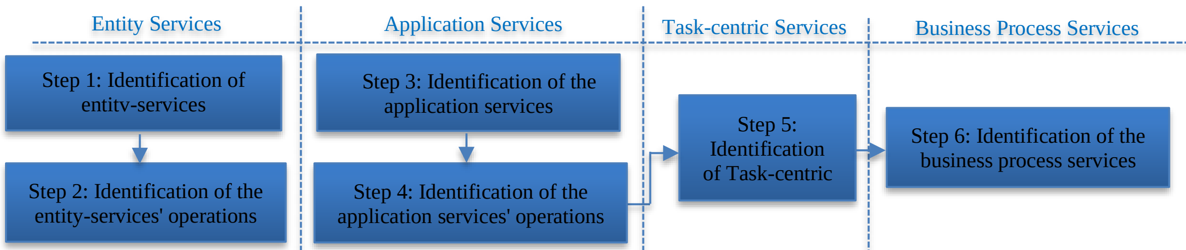

Fig. 1 The six steps toward service identification

Table 1. A prototype of the Matrix is used for the identification of the services

Information	Occurrences	Act1	Act2	Act3	Act4	Act5	Act6	Act7	Act8	Act9	Act10
		5	1	**6**	**10**	2	3	1	**6**	3	**5**
Inf1	8	0%	0%	40%	0%	0%	0%	0%	0%	0%	0%
Inf2	**12**	100%	0%	0%	67%	0%	0%	0%	100%	0%	0%
Inf3	10	0%	0%	0%	0%	0%	0%	100%	0%	0%	100%
Inf4	**34**	0%	0%	0%	0%	0%	100%	0%	0%	0%	0%
Inf5	5	0%	0%	0%	0%	0%	0%	0%	0%	0%	0%
Inf6	**10**	100%	0%	40%	0%	0%	0%	0%	0%	0%	30%
Inf7	10	0%	0%	0%	0%	0%	80%	0%	0%	0%	0%
Inf8	**8**	30%	0%	0%	80%	0%	0%	0%	0%	0%	0%

Highly solicited activities and information.

Stongly linked information

Key information

3.1. Preparation Phase

The proposed approach uses Business activities and information as the primary sources for service identification. In a business process, activities can be of 3 types: manual, automatic (without human intervention), or semi-automatic. Candidate activities for services are of the automatic or semi-automatic type. In addition, these activities must be conducted using a resource that can provide a service (software, website, etc.). Obviously, this resource must be under the company's ownership, thus having the right to customize and identify the logic that will be encapsulated in the identified services.

These services will be handled by computer systems rather than manually by human beings. For this, we will be interested just in information handled by the automatic and semi-automatic activities supported by computer systems.

3.1.1. Step 1: Entity Services' Identification

Entity services encapsulate and manage the business data entities (customer, account, etc.). They are identified from the key information manipulated in the business process models.

Sub-step 1.1: Identification of the Key Information

Information is considered vital if it meets at least one of the following conditions: i) its number of occurrences exceeds a given threshold, ii) it is strongly related to a highly solicited activity. These two conditions will ensure the identified entity services provide useful functionalities.

Sub-step 1.1.1: Select the most Recurrent Information in the Business Process Models

This sub-step will browse through all the information identified in the preparation phase and determine utilization frequencies. That information will be considered highly solicited if the number of occurrences exceeds a given threshold. The threshold is a parameter that determines the degree of reuse of the corresponding potential service and, hence, the quality of the SOA solution to implement. This parameter may be set by analysts according to their needs in terms of reusability. The following equation is proposed to set the threshold:

$$THLD = [average + (std\ deviation/y)] \quad (1)$$

Where [X] is the operator that rounds X to its nearest integer, and Y is an integer. The information identified in this sub-step will form the first lot of the vital information set, named E_KEY_INF. In our case, Y=2 was used after experimentation with the data set.

Sub-step 1.1.2: Identification of Solid Information Related to a Highly Solicited Activity

This sub-step begins with the selection of the highly solicited activities. If the number of occurrences exceeds a given threshold, it will be considered highly used for each activity. As in sub-step 1.1.1, this threshold determines the degree of reuse of the identified services and the quality of the future SOA solution. Analysts may set this metric according to their needs in terms of reusability. We recommend using the same equation (1) to determine this threshold. The activities identified in this sub-step will form a second set called E_ACT_SOLICITED.

A piece of information is closely linked to an activity if it is used in at least 50% of the occurrences of this activity. Pieces of information identified in this level will complete the set E_INF_KEY. To automate the identification of closely linked information, we create a matrix that gives the user's rate of information in each activity (Table 1)

Sub-step 1.2: Identification of the Entity Services

Once the key pieces of information are recognized, we group them semantically to identify the business entities handled in the business context of the enterprise. We will locate an entity service that will manage each entity. For example, suppose we have the following pieces of key information: "Client identifier," "Client demands," and "Client report." In that case, it is clear that there is a semantic link between these pieces of information, and we can group them to form a business entity named "Client" and identify the "Client" entity service. At the end of this stage, we have identified all entity services needed to manage the business entities used in the company's business context.

3.1.2. Step 2: Identification of the Entity Services' Operations

The entity service operations must be autonomous and provide reusable functionalities. They may be basic (getEntity, createEntity, updateEntity, etc.) or business. The basic operations will be identified from the related key information usage types (Read, Create, Modify, etc.). As for the business operations, they will be identified from the most solicited activities. We will surely get reusable and generic Operations using key information and the most solicited activities. The identification process of entity services' operations depends on the operation's type:

Basic Operations

For each piece of information that belongs to the set E_INF_KEY and is part of an entity service, identify the different usage types (Read, Create, or Update) made on this information. For each usage type, we associate the corresponding operation.

Business Operations

We browse the set E_ACT_SOLICITED of the most requested activities in the business models. If an activity satisfies the following characteristics: i) uses a single business entity (we use Table 1), ii) does not require human intervention, and iii) is not part of the basic operations already identified, then it will be considered as a business operation.

3.1.3. Step 3: Identification of the Application Services

Unlike entity services, identifying application services does not require business analysis expertise. The application services layer is a pure, service-oriented abstraction of an organization's technical environments, best defined by those who most understand these environments [1]. For this reason, it is sometimes challenging to identify all needed application services using a top-down approach.

During the identification of entity services, we note that many pieces of key information cannot form a data entity but require technical processing (e.g., generation of a document, conversion to another format, etc.). These pieces of key information are grouped according to the type of processing required. We associate an application service for each group that is responsible for the different treatments. This step must be done jointly with the IT specialists who better understand the enterprise's technical environment.

3.1.4. Step 4: Identification of the Application Services' Operations

The identification process of the application services' operations is similar to the process followed for the entity services (step 2), except that we do not identify business operations since, by definition, application services must include only purely technical methods. In fact, operations are identified from the pieces of key information that form the service. In the case of application services, operations must correspond to how we exploit these pieces of information and the technical constraints imposed by the application environment.

The identification process is as follows: for each piece of information operated by the application service and which belongs to the set E_INF_KEY, identify the different types of uses in the business process models (Read, Create, or Update). For every kind of use, associate the corresponding technical treatment (e.g., conversions, generation of documents, etc) that will be part of an operation.

3.1.5. Step 5: The Identification of Task-Centric Services

Task-centric services encapsulate the business logic specific to a coarse-grained task. The identification process for this kind of service is as follows:

Sub-step 5.1: Identify candidate activities to be task-centric services

In this step, we identify activities that involve more than one entity or application service and use the same services each time (regardless of the number of occurrences of these activities). This corresponds to the coarse-grained activities in which there is business logic that business analysts may judge to be more convenient to encapsulate in the activity than to show it in the business process models (we use Table 1).

Sub-step 5.2: Decide which Activity will be a Task-Centric Service

For each activity identified in sub-step 5.1, decide whether it must be divided (to have one service for each activity) or converted to a task-centric service that will control all the services invoked by the activity. Obviously, analysts can propose other task-centric services that follow from the business needs.

Sub-step 5.3: Define the Logic of the Service

As we said previously, the business logic to implement under the task-centric service is encapsulated in the activity from which we identify the service. Business process modeling notations do not provide details on the different interactions done within the activities. To define the logic encapsulated in the task-centric services, we use a UML sequence diagram to describe the various exchanges made within this activity. This will help to identify the operations to implement and the messages exchanged within the service.

3.1.6. Step 6: The Identification of the Business Process Services

This step is divided into three sub-steps.

Sub-step 6.1: Identifying the Processes to be Orchestrated

During this step, the analyst identifies which processes will be the subject of orchestration. Indeed, the enterprise can orchestrate all of its business processes, but they must be prioritized. The choice of the most appropriate business processes to orchestrate is made according to several criteria: the degree of automation of the process, the cost induced by the process, the expected ROI (*Return On Investment*), the security policy applied to the process, etc. For this purpose, the analysts can use multi-criteria analysis methods, like AHP (*Analytic Hierarchy Process*), WSM (*Weight Sum Method*), WPM (*Weight Product Method*), etc.

Sub-step 6.2: Design an Intermediate Model

Business processes may have many details that are not necessarily part of the orchestrations. Deriving BPEL orchestrations directly from business process models can introduce confusion and leave significant room for error. For this reason, we propose developing an Intermediate Model (IM) that ensures a shared vision of the process.

Table 2. Results of the study case

	Service	**Description**
Orchestration service	OpenAccount	Create a bank account by following the business rules outlined in the “create account” business process.
	CloseAccount	Close a bank account after a client requests it.
	PurshaseOrderForTitle	Process a purchase order for an investment title.
	OrdreVenteTitre	Process a sale order for an investment title.
	TrsctInvestPurchase	Once validated, the purchase order will be translated into a transaction that must be addressed by the bank system.
	TrsctInvestSale	Process a transaction for an investment sale order.
Task-centric services	ProcessingTransaction	Process a deposit and/or withdrawal transaction
	ActionPlanPreparation	This service covers the preparation of an action plan for a client.
	AccountChecking	Check the current situation of a given bank account.
	InvestRefusalNtceSub	Submit a notice of investment instruction refusal.
	AppointmentScheduling	When scheduling an appointment, the agent makes the necessary changes to the action plan and specifies the information related to the appointment.
	ProductSuggestion	Provide a list of suggested bank products to offer to the customer, taking into account his action plan.
Entity-services	Account	Manage the client account of the bank
	Investment	Manage the investments and their eligible funds.
	Client	Manage the client’s information.
	ActionPlan	Manage and monitor the action plan for a given customer.
	Appointment	Manage the appointment scheduling.
	InvestmentInstruction	Manage and verify the investment instructions.
Application	GenerateNotice	Generate notices (email, electronic document, etc.) to be exchanged between systems.
	PhoneSystemMgmt	Manage and treat the telephone instructions made by the customer.
	GenerateDocument	Generate documents (PDF, Word, etc.) whose content is stored in a native format (XML, database, etc.).

This model should provide only information exchanged by computer systems. In addition, it should contain system-supported activities that the orchestration engine can orchestrate.

Sub-step 6.3: Design of the Orchestrations

To reduce the errors introduced during the orchestration of business processes, we propose an algorithm that allows the automatic generation of BPEL orchestration from IMs. This algorithm identifies and translates into BPEL code the different structural patterns (sequence, flow, while, repeat, etc.) that can exist in an IM. The basic idea behind this algorithm is to identify components C that follow a predefined pattern and translate them into BPEL code. $[IM]_P$ denotes the set of an IM's nodes.

1. while ($[IM]_P \neq \emptyset$)
   a. select the current node (CN)
   b. If the CN marks the beginning of a component, then construct a new component by specifying its type (repeat, while, etc.): C:=new component(type)
   c. Else, if CN marks the end of an element, then translate the CC into BPEL code and C:= $\emptyset$ and go to 1
   d. if ($C \neq \emptyset$) then affect the CN to the component C
2. Output the BPEL code attached to the IM

Once the orchestrations are generated and implemented in an orchestration engine, they can be converted automatically to business process services.

4. Case Study

This case study elaborates on the experience of a prominent corporate bank that extends a wide array of financial products and services, such as checking accounts, savings, business loans, and diverse investment vehicles, across its extensive network of 50 branches. Within this financial institution's complex architecture, eleven core business processes were meticulously identified to encompass the entirety of the loan processing system. These processes were intricately modeled using the advanced xBPM® software, which revealed a comprehensive set of 452 distinct activities.

Delving into the bank's sophisticated business processes led to the identification of 21 distinct services, effectively forming a stratified Service-Oriented Architecture (SOA). This architecture spanned several layers, from foundational application services, instrumental in executing the day-to-day operations, to higher-level orchestration services that ensure these operations are seamlessly integrated and aligned with the bank's business strategies (as delineated in Table 2). While some of these services might seem intuitive to experienced SOA analysts, applying our proposed service identification approach allows for a more rigorous and justifiable selection of services tailor-made for the bank's unique operational landscape.

This approach's meticulous application made it apparent that while a systematic method to identify services is critical, it is also subject to the individual analyst's discretion. Different analysts may arrive at varying conclusions, underlining the inherently subjective nature of service identification. By introducing additional heuristics to the approach, we aimed to mitigate these disparities, striving for a more consistent and repeatable process across different analysts. Yet, it is essential to acknowledge that while the gap in interpretation can be narrowed, it must be partially bridged due to the complex judgment calls required during the process.

The extended analysis also allowed a deeper understanding of the bank's operational demands and customer engagement strategies. By dissecting each business process, we uncovered underlying patterns and dependencies that might otherwise remain obscured. This granular scrutiny was instrumental in pinpointing opportunities for optimization, automation, and innovation within the bank's service offerings. For instance, the role of digital transformation in loan processing was brought to the forefront, suggesting the potential for blockchain implementation to enhance security and transparency.

Moreover, the service identification process went beyond technical analysis; it engaged cross-functional teams to align IT capabilities with business goals. This collaboration was pivotal in shaping a customer-centric SOA where each identified service supported operational efficiency and enhanced customer experience. For example, services were designed to process transactions and provide personalized financial advice, leading to a more proactive and customer-oriented banking environment.

In summary, the comprehensive case study demonstrated the applicability and robustness of the proposed service identification approach. It highlighted the critical importance of analytical precision, stakeholder collaboration, and strategic alignment in developing a multi-layered SOA within the dynamic and complex banking domain.

5. Conclusions and Future Work

Service identification is acknowledged as the cornerstone in the lifecycle of Service-Oriented Architecture (SOA). Due to the intricate web of complex business processes, abundant informational assets, and diverse stakeholders, this critical process demands service-oriented analysts to possess a heightened sense of diligence and comprehension. Accurate service identification sets the stage for the ensuing phases of SOA, ultimately impacting the organization's operational efficacy and strategic adaptability.

In this research, we have crafted and delineated a systematic approach designed to equip service-oriented analysts with a robust framework for identifying services most conducive to the organization's objectives. Leveraging

Unified Modeling Notation (UMN) business process models as the foundational input, our method facilitates a top-down analysis that interweaves seamlessly with the UMN framework, furnishing analysts with a decisive decision support tool. This tool is engineered to streamline the service identification phase, enhancing precision and mitigating potential oversights.

The efficacy of our proposed methodology was empirically validated through a comprehensive case study within the banking sector. This practical investigation identifies and categorizes four distinct types of services, each integral to the bank's SOA strategy. The success of this case study has laid the groundwork for the subsequent phase of our research.

Looking ahead, our investigative trajectory is set toward automating the intermediary modeling stage. The envisioned automated process aims to convert business process models into intermediate models that can serve as a prelude to the actual service development. By automating this transition, we anticipate a significant acceleration in the SOA development lifecycle, a reduction in manual errors, and an overall increase in efficiency.

Future work will also explore integrating artificial intelligence and machine learning techniques to refine service identification further. This includes developing algorithms that recognize patterns within business process models that may allude to potential services. Moreover, these advanced techniques could offer predictive insights, suggesting services that align with current operational requirements and are scalable and adaptable to future business transformations.

Another promising direction for future research is expanding our methodology to encompass more diverse industry sectors beyond banking. This would validate the approach's versatility and ensure its applicability across different business domains, each with unique challenges and nuances.

Furthermore, we intend to establish a comprehensive set of metrics to assess the performance of the identified services quantitatively. Such metrics would evaluate various aspects of service performance, including reusability, granularity, and alignment with business objectives, thus providing a multifaceted evaluation of SOA success.

In essence, our pursuit is to automate and infuse cognitive processing into the early stages of SOA development. By doing so, we aspire to elevate the service identification process from a task of mere technical execution to one of strategic foresight, enabling organizations to harness the full potential of SOA.

References

[1] Thomas Erl, *Service-Oriented Architecture: Concepts, Technology, and Design*, Prentice Hall PTR, pp. 1-792, 2005. [Google Scholar] [Publisher Link]

[2] Asit Dan, Robert D. Johnson, and Tony Carrato, "SOA Service Reuse by Design," *Proceedings of the 2nd International Workshop on Systems Development in SOA Environments*, pp. 25-28, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[3] Rene Boerner, and Matthias Goeken, "Service Identification in SOA Governance Literature Review and Implications for a New Method," *3rd IEEE International Conference on Digital Ecosystems and Technologies*, Istanbul, Turkey, pp. 588-593, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[4] Karsten Klose, Ralf Knackstedt, and Daniel Beverungen, "Identification of Services - A Stakeholder-Based Approach to Soa Development and Its Application in the Area of Production Planning," *ECIS Proceedings*, pp. 1802-1814, 2007. [Google Scholar] [Publisher Link]

[5] Senthil Mani et al., "Using User Interface Design to Enhance Service Identification," *IEEE International Conference on Web Services*, Beijing, China, pp. 78-87, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[6] DongSu Kang, Chee-Yang Song, and Doo-Kwon Baik, "A Method of Service Identification for Product Line," *Third International Conference on Convergence and Hybrid Information Technology*, Busan, Korea, pp. 1040-1045, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[7] Rana Yousef et al., "BPAOntoSOA: A Generic Framework to Derive Software Service-Oriented Models from Business Process Architectures," *Second International Conference on the Applications of Digital Information and Web Technologies*, London, UK, pp. 50-55, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[8] A. Arsanjani et al., "SOMA: A Method for Developing Service-Oriented Solutions," *IBM Systems Journal*, vol. 47, no. 3, pp. 377-396, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[9] Vishal Dwivedi, and Naveen Kulkarni, "A Model-Driven Service Identification Approach for Process Centric Systems," *Congress on Services Part II*, Beijing, China, pp. 65-72, 2008. [CrossRef] [Google Scholar] [Publisher Link]

[10] Leonardo Guerreiro Azevedo et al., "A Method for Service Identification Frombusiness Process Models in a SOA Approach," *Enterprise, Business-Process and Information Systems Modeling*, vol. 29, pp. 99-112, 2009. [CrossRef] [Google Scholar] [Publisher Link]

[11] Dieter Van Nuffel, "Towards a Service-Oriented Methodology: Business-Driven Guidelines for Service Identification," *On the Move to Meaningful Internet Systems*, pp. 294-303, 2007. [CrossRef] [Google Scholar] [Publisher Link]

[12] P. Huysmans et al., "Towards Systematic Identification of Services: A Domain-Specific Approach," *ICSOFT 2007: Proceedings of the 2nd International Conference on Software and Data Technologies*, pp. 267-277, 2007. [Google Scholar]

[13] Srikanth Inaganti, and Gopala Krishna Behara, "Service Identification: BPM and SOA Handshake," *BPTrends*, pp. 1-12, 2007. [Google Scholar] [Publisher Link]

[14] A.R. Eric Chan, "An Evaluation of Various Business Process Modelling Techniques Applied to Healthcare," 2007. [Google Scholar]

[15] Mykola Beshley et al., "Traffic Engineering and Qos/Qoe Supporting Techniques for Emerging Service-Oriented Software-Defined Networks," *Journal of Communications and Networks*, vol. 26, no. 1, pp. 99-114, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[16] Yue Zhao et al., "Service Oriented Considerate Routing: Data, Predictions and Robust Decisions," *Predictions and Robust Decisions*, pp. 1-45, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[17] Amira G. Hosary, Ahmed Emran, and Basel El-Saghir, "Design of Service Oriented Architecture for an IoT Healthcare Management System," *Indonesian Journal of Electrical Engineering and Informatics*, vol. 12, no. 1, pp. 87-100, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[18] Ivo Pereira et al., "A Machine Learning as a Service (MLaaS) Approach to Improve Marketing Success," *Informatics*, vol. 11, no. 2, pp. 1-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]

[19] Intan Setya Palupi et al., "Analysis and Design of Oriented Architecture (SOA) with Service-Oriented Modeling and Architecture (SOMA) Method in Trucking Services Company (Case Study: PT Argo KencanaTransindo)," *Asian Journal of Social and Humanities*, vol. 1, no. 12, pp. 1175-1185, 2023. [CrossRef] [Google Scholar] [Publisher Link]

[20] Rafael Muñoz Terol et al., "Applying the Service-Oriented Architecture (SOA) Approach Andsearch Engine Optimization (SEO) Best Practices to the Development of a Video Streaming Meta Platform," *SSRN*, pp. 1-34, 2024. [CrossRef] [Google Scholar] [Publisher Link]