WEB SERVICE BUSINESS MODELS: BETWEEN THEORY AND PRACTICE

Maksymilian Iwanow*[1]

[1]*Wroclaw University of Economics and Business: Komandorska 118/120 Wroclaw 53-345 Poland: Department of Information Technology, maksymilian.iwanow@ue.wroc.pl*

Abstract

This paper examines the current state of web service business models as presented in the literature and identifies their main limitations in the context of today's web environment. The study applies a two-stage methodology: first, a scoping review is conducted to organize existing knowledge, and second, informal participatory observation is used to develop real-world use case illustrating practical gaps between theory and practice. The results reveal a clear mismatch between theoretical frameworks and the realities of complex, composite web services, particularly in areas such as service discovery, composition, and Quality of Service (QoS). The findings highlight the need for future research to focus on business models that better reflect the interdependent and dynamic nature of modern web services. The originality of this study lies in combining a structured literature review with empirical observations, offering a balanced view that connects theoretical understanding with real-world implementation.

Keywords

Business Model, Web Service, Service Provider, Service Requestor, QoS

JEL Classification

L86

DOI: https://doi.org/10.14311/bit.2025.02.02

Introduction

A web service is a self-describing, self-contained software module that can be accessed through a network, such as the Internet [1]. It is designed to perform various tasks, solve problems, and support transactions [2]. This type of technology is transforming the web from a simple information repository into a distributed computational platform [3]. A web service functions as a software component that operates independently of hardware or specific implementation, while offering a defined functionality. It can be viewed as a black box [4]; a component that takes input and responds with calculated output. Implementing web services, developers can create distributed systems and Internet-based applications. These services enable software components to communicate and cooperate across the Internet, regardless of their location or implementation details. To access these services, different protocols and interfaces are used, and their selection often depends on user preferences. For consumers, selecting a web service that matches their specific needs has become an important consideration [5].

Authors in [6] observed that there is no single, consistent definition of the term *business model*, and they tried to organize the various interpretations. A business model may be described as a compact and integrated structure centered on the customer. It may also be defined as a combination of customer-related strategies, key resources, and value chains ([7], [8], [9]). Another approach sees the business model as a mix of strategy, tactics, and operational activities that are essential for success in each business environment at a specific time. As the Internet continues to evolve and influence economies worldwide, it has become a driving force behind changes in traditional business roles and models. It affects how organizations create value [10], and it encourages the development of new and innovative forms of organizational and business structures.

This article aims to present the current state of web service business models as described in academic literature, and to identify their limitations. The research is based on a two-stage methodology. First, a scoping review [11] is carried out to examine the existing body of knowledge on business models related to web services. Second, informal participatory observation [12] is used to explore real-world use case that highlights where current theoretical frameworks may fall short. The originality of this study lies in the inclusion of real examples encountered by IT departments in the modern business landscape. The main outcomes include a structured overview of the current state of the art in web service business models, along with a use case that demonstrates areas where today’s business realities have become too complex for existing models.

The structure of the article is as follows. Chapter two reviews the current state of knowledge regarding web service business models in the academic literature. Chapter three discusses the theoretical limitations of these models in the context of modern companies that create value through web services. Chapter four presents real-life use case to illustrate the gaps identified in the previous chapter. The final chapter offers a summary of findings and outlines possible directions for future research.

A Review of the Current State of the Literature

The term *business model* is not consistently defined in the context of web services. According to [13], electronic commerce business models generally focus on three main aspects:

- What is the architecture of the product (such as a web service), how does information flow, and who are the key actors and what roles do they play in the given environment?
- What are the potential benefits for those business actors?
- What are the sources of revenue for the business unit, such as the web service provider?

Authors in [14] argue that businesses must continuously adapt to dynamic environment, particularly when operating within the digital technology ecosystem. Innovation in business models is essential to meet changing customer demands and to address environmental uncertainties. In [15] there are identified three core elements of a business model:

- Content – What tasks need to be performed?
- Structure – How are the tasks related, and in what order should they be carried out?
- Supervision – Who is responsible for performing the tasks?

A commonly used model of web services involves three main actors ([16], [17], [18]):

- Service requestor – Searches for a service that meets specific requirements using a service registry.
- Service provider – Publishes and offers services by registering them within a service registry.
- Service registry (directory) – A central place that maintains information about all available services.

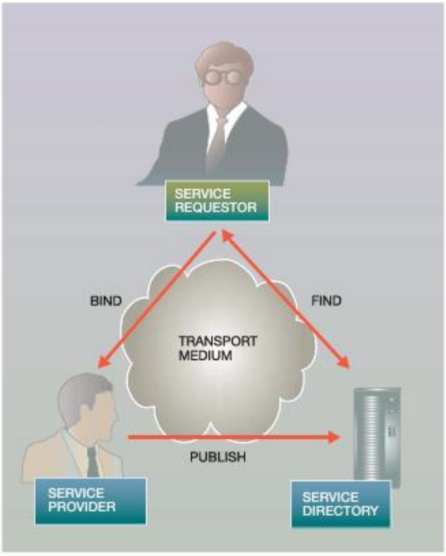

Figure 1: The most common web service business model (source: [16])

Authors in [19] proposed a model in which the service broker serves as an interface to the service registry database.

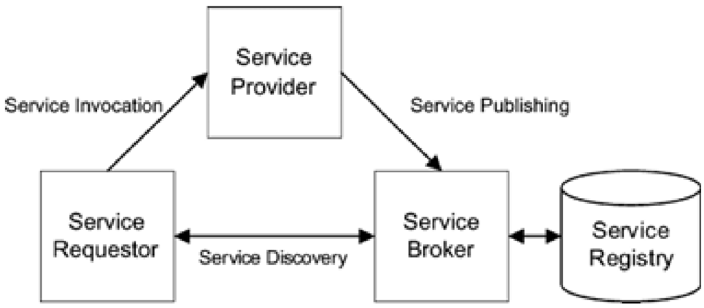

Figure 2: The service broker web service business model (source: [19])

A web service requestor looks for a solution that satisfies both functional and non-functional requirements. The benefits for the requestor can be considered on two levels. First, if functional expectations remain fixed, the user may benefit if the selected service provides better Quality of Service (QoS) parameters [20] – such as speed or reliability – compared to alternatives used previously. For example, when making an online payment, a service that processes the transaction instantly and charges the lowest commission would be the most attractive option.

Secondly, the requestor may gain additional value through improved functionality. In some cases, users are unaware of features that could enhance their experience. If a service offers new, useful functionality without additional cost or loss of quality, the user is likely to switch. For example, a platform that sends a purchased ticket directly to a user's email, eliminating the need to retrieve it manually, would likely be preferred by most people.

Service providers benefit when their services are used by many consumers. Increased usage leads to higher revenue from a single functionality. These consumers may range from individual users with specific needs to large organizations requiring continuous access to certain resources.

Common web service business models typically do not place the organization in a fixed physical location, which makes it more difficult to identify precise revenue sources [21]. In many cases, the business unit is simplified to the role of the service provider. These providers can generate income in various ways, as described by [22], including:

- Advertising-based model – The service is free for all users, both private and commercial. Revenue is generated through advertisements displayed within the application's interface [23].
- Subscription or pay-per-use model – Private users may pay a fixed fee per request, while larger businesses might subscribe to the service for unlimited use [24].
- Exclusive outsourced model – The service is provided exclusively to a single client or company and is not listed in a public registry. It is accessible only through the company's internal service broker, and the revenue depends entirely on the terms of the business contract [25].

Limitations of the Current State of Art

The most significant gap observed nowadays concerns complex web services, meaning services that are created through the combination or invocation of several smaller, functionally limited services [26], [27]. There remains an unaddressed space between the consumption of such complex services by the requestor and their provision by the provider. In practice, a service consumer may request functionality that the provider cannot deliver directly, as it must construct new logic that integrates its own internal components with external ones supplied by vendors. This creates a disconnection between all model participants, resulting in inconsistencies in how new functionalities are composed from multiple independent elements.

A key question arises: how should the service requestor behave in such a scenario? Should the consumer understand exactly which smaller sub-functionalities must be invoked and request them separately from the service registry? Or, perhaps more reasonably, should it be enough for the consumer to simply define the desired business outcome, leaving the technical composition process to the provider? On the other hand, should the service provider make all its developed functionalities publicly accessible? There are cases where certain functions are not usable without prior access to other components or data, which may not be available to the consumer. Furthermore, if some sub-functionalities are licensed from third-party vendors, they may not be publicly exposed due to contractual limitations. These questions raise a broader issue: what does service discovery truly mean in the modern web ecosystem? As mentioned in the Introduction, the web has evolved from being a static repository of information into a dynamic computational platform where users search for actions and processes, not just resources.

Currently, there are no well-established mechanisms for service discovery and static/dynamic composition [28] that address the needs of business units developing such complex solutions. It is not clearly explained how advanced functionalities should be structured in a way that provides a competitive advantage in the market. If certain internal services are meant to remain private and visible only within a defined subnet, the question arises of how to ensure this in practice. Similarly, what mechanisms should be in place when certain service instances become unavailable, interrupting the execution of processes that depend on them?

Another notable issue is the lack of QoS aware mechanisms that could make such systems more competitive. Current management practices focus mainly on individual services, Service Level Agreements (SLA) [29], and isolated performance metrics. As a result, there is a significant gap in handling QoS for composite services that combine both internal and third-party components. Existing frameworks rarely address these scenarios, treating them as exceptions rather than integral parts of the model. This is a critical omission, since today's web services primarily compete through quality, reliability, and responsiveness. Even the most advanced implementation will fail to attract users if it is unreliable, returns frequent HTTP [30] errors, or responds too slowly to meet real-time demands.

Empirical Use Case Highlighting Gaps in Theoretical Models

An individual wants to make an online reservation for a holiday in Turkey. The desired service is complex, as it must include multiple interconnected elements such as hotel accommodation, meals, flights, airport transfers, and travel insurance. Because an all-inclusive holiday represents a significant financial expense, the process should be highly reliable and capable of handling unexpected events without negatively affecting the customer's experience. In other words, even if something goes wrong during the booking process, the system should ensure that the client remains protected and informed.

The first question is: what kind of resource is the user seeking? The user is not looking for static information but rather for a service that performs a real-world task – booking and reserving physical resources. To achieve this, the user relies on a service registry, which in this case may take the form of a travel platform or website that aggregates offers from multiple travel agencies.

Secondly, travel agencies offering such comprehensive packages often rely on hidden or internal functionalities that are not visible to the end user. These include integrations with flight booking systems, hotel reservation interfaces, transport providers, and insurance agencies. Without these background components, agencies could not deliver the final travel package at all. They depend on multiple external systems and data sources that must work together seamlessly, even though the user interacts only with a single visible interface.

Thirdly, the user is unaware of the many sub-services involved in fulfilling the holiday booking request. The customer's business need is simple – to go on holiday – but the technical realization of that need involves a complex chain of service calls and data exchanges. If the customer was required to manually compose this entire process using a generic service registry or interfaces directory, the operation would almost certainly fail. The complexity of service composition and dependency management cannot be placed on the end user, who only expects a functional, easy-to-use experience.

Finally, the fourth point concerns QoS. The entire process must be stable, secure, and transparent. No user wants to face uncertainty about whether a payment was processed successfully, or to arrive at a destination only to find that the hotel reservation did not go through. Reliability, transaction integrity, and real-time communication between subsystems are crucial for maintaining user trust and satisfaction.

This example clearly illustrates the limitations of current web service business models, which often fail to address the dynamic nature of composite services. It highlights the gap between theoretical frameworks and practical reality – where services are built on top of multiple, interdependent systems that must cooperate smoothly to deliver a unified and dependable user experience. The three actors described in the literature (provider, requestor, and registry) and their relationships cannot adequately capture the complexity of modern service composition, Quality of Service (QoS) management, internal (private) registries, and the algorithms that enable these processes.

Conclusions

The analysis presented in this paper clearly shows that the reality of today's Internet and the way web services operate differ significantly from the theoretical business models described in the literature. Existing frameworks often fail to reflect the dynamic, interconnected, and composite nature of modern web services, leaving important areas – such as service composition, discovery, and QoS – insufficiently addressed or overly simplified. This gap highlights the need for a more systematic and comprehensive understanding of how web service business models function in practice.

Future research should focus on developing an extended business model that not only encompasses the key problems identified in this study but also provides adaptable structures for addressing new and evolving challenges. Such a model should aim to integrate theoretical precision with practical applicability, offering a unified approach that aligns academic perspectives with the realities of modern digital ecosystems.

References

[1] H. Kreger, “Web Services Conceptual Architecture (WSCA 1.0),” IBM, 2001.

[2] M. Papazoglou, Web services: principles and technology, Pearson Prentice Hall, 2007.

[3] D. Fensel and C. Bussler, “The Web Service Modeling Framework WSMF,” *Electronic Commerce Research and Applications,* vol. 1, no. 2, pp. 113-137, 2002.

[4] J. Zhu, P. He, Q. Xie, Z. Zheng and M. R. Lyu, “CARP: Context-Aware Reliability Prediction of Black-Box Web Services,” in *2017 IEEE International Conference on Web Services (ICWS)*, 2017.

[5] Y. Hao, Y. Zhang and J. Cao, “A novel QoS model and computation framework in web service selection,” *World Wide Web,* vol. 15, pp. 663-684, 2012.

[6] A. K. Wiśniewski and M. Raczyńska, “Strategia a model biznesu,” in *Współczesne Zarządzanie Przejawy, uwarunkowania i wyzwania Tom 2*, Wydawnictwo Naukowe Uniwersytetu Mikołaja Kopernika w Toruniu, 2019, pp. 155-166.

[7] M. Sako and E. Zylberberg, "Supplier strategy in global value chains: shaping governance and profiting from upgrading," *Socio-Economic Review,* p. 687–707. doi:10.1093/ser/mwx049, 2017.

[8] E. I. Stańczyk-Hugiet, "O istocie strategii wiedzy," *Przegląd Organizacji,* pp. 6-8. doi:10.33141/po.2004.78.01, 2004.

[9] A. Feller, D. Shunk and T. Callarman, "Value Chains Versus Supply Chains," *BPT Trends,* 2006.

[10] J. Wielki, “Modele wpływu przestrzeni elektronicznej na organizacje gospodarcze,” *Monografie i Opracowania Uniwersytetu Ekonomicznego we Wrocławiu,* vol. 211, 2012.

[11] H. Arksey and L. O'Malley, "Scoping studies: towards a methodological framework," *International Journal of Social Research Methodology,* vol. 8, no. 1, pp. 19-32, 2005.

[12] B. Glinka and W. Czakon, Podstawy Badań Jakościowych, Polskie Wydawnictwo Ekonomiczne S.A., 2021.

[13] P. Timmers, “Business Models for Electronic Markets,” *Electronic Markets,* vol. 8, no. 2, 1998.

[14] T. Haaker, H. Bouwman, W. Janssen and M. de Reuver, “Business model stress testing: A practical approach to test the robustness of a business model,” *Futures,* vol. 89, pp. 14-25, 2017.

[15] A. K. Wiśniewski, “Wartość W Koncepcji Modeli Biznesu,” *Przedsiębiorstwo we współczesnej gospodarce / Research on enterprise in modern economy,* 2016.

[16] F. Leymann, D. Roller and M.-T. Schmidt, “Web services and business process management,” *IBM Systems Journal,* vol. 41, no. 2, 2002.

[17] R. Verma and A. Srivastava, “A Novel Web Service Directory Framework for Mobile Environments,” in *2014 IEEE International Conference on Web Services*, 2014.

[18] M. Moran, M. Zaremba, A. Mocan and C. Bussler, “Using WSMX to bind Requester & Provider at Runtime when Executing Semantic Web Services,” in *Proceedings of the WIW 2004 Workshop on WSMO Implementations*, 2004.

[19] S. Dustdar and M. Treiber, “A View Based Analysis on Web Service Registries,” *Distributed and Parallel Databases,* vol. 18, no. 2, pp. 147-171, 2005.

[20] W. D. Yu, R. B. Radhakrishna, S. Pingali and V. Kolluri, “Modeling the Measurements of QoS Requirements in Web Service Systems,” *SIMULATION,* vol. 83, no. 1, pp. 75-91, 2007.

[21] K.-M. Chao, “E-services in E-business engineering,” *Electronic Commerce Research and Applications,* vol. 16, 2015.

[22] F. Welle Donker, “Public sector geo web services: which business model will pay for a free lunch?,” in *Sdi Convergence. Research, Emerging Trends, and Critical Assessment*, 2009.

[23] M. Tatsubori, “Towards an Advertising Business Model for Web Service Mashups,” in *International World Wide Web Conference*, 2009.

[24] S. Baravkar, O. Pellegrini, P. Gaikwad, E. Tilevich and Z. Song, “"How Can I Be of Service?"—A Comprehensive Analysis of Web Service Integration Practices,” in *2024 IEEE International Conference on Web Services (ICWS)*, 2024.

[25] N. Chou and M. Lee, “Outsourcing Business to cloud computing with communication services,” in *Annual Conference of KIPS*, 2010.

[26] D. B. Claro, J.-K. Hao and P. Albers, “Web Services Composition,” in *Semantic Web Services, Processes and Applications*, 2006, pp. 195-225.

[27] S. Dustdar and W. Schreiner, “A survey on Web services composition,” *International Journal of Web and Grid Services,* vol. 1, no. 1, pp. 1-30, 2005.

[28] Q. Z. Sheng, X. Qiao, A. V. Vasilakos, C. Szabo, S. Bourne and X. Xu, “Web services composition: A decade’s overview,” *Information Sciences,* vol. 280, pp. 218-238, 2014.

[29] L.-J. Jin, V. Machiraju and A. Sahai, “Analysis on service level agreement of web services,” Hewlett-Packard Company Technical Reports, 2002.

[30] L. Popa, A. Ghodsi and I. Stoica, “HTTP as the NarrowWaist of the Future Internet,” in *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*, 2010.