A Cross-Layer Integration Framework for the Development of a Platform Business Model, Applications and Services

Vjeran Strahonja, Dijana Oreški
University of Zagreb, Faculty of Organization and Informatics, Varaždin, Croatia
vjeran.strahonjaov@foi.hr, dijana.oreski@foi.hr

DOI
https://doi.org/10.18690/um.fov.4.2026.58

ISBN
978-961-299-152-4

Digital platforms have transformed contemporary economic structures by enabling multi-sided interactions, network effects, and ecosystem-based value creation. However, existing research frequently treats business models, technical architectures and service design as separate domains, resulting in conceptual fragmentation and practical misalignment. This paper proposes a cross-layer integration framework for the development of Digital Platform Business Models (DPBMs), platform applications, and platform services. The framework integrates lifecycle perspectives and construct mapping across business, service, and application layers. It introduces a bidirectional development logic that enables both top-down (business-driven) and bottom-up (technology-driven) innovation. By formalizing the transformation of Business Model Canvas (BMC) constructs into service design artifacts and Unified Modeling Language (UML) models, the framework enhances traceability, consistency, and alignment across layers. The study contributes to digital platform research by offering a structured methodological foundation for cross-layer integration and identifying future directions for tool-supported model transformation.

Keywords:
digital platform business models,
digital platform applications,
digital services,
service design artefacts,
digital platform design

University of Maribor Press

1 Introduction

The motivation for this research stems from the observation that the success of digital platforms (DPs) depends on the alignment of the business model (BM) with technical capabilities and service design. However, these dimensions are often examined separately in the literature. Business models originated as instruments of strategic management and are still most frequently viewed in this context. Nevertheless, contemporary research emphasizes their broader functional applications in entrepreneurship, operational management, financial planning, innovation management, digital transformation, and sustainable development.

Business model research focuses on value creation, delivery, and capture mechanisms, whereas technical architecture research emphasizes modularity, scalability, interoperability, and system performance. The gap between these perspectives may result in platforms that are technically sophisticated but commercially unviable, or strategically sound but technically infeasible

The subject of this research comprises business models based on digital platforms, as well as platform applications and platform services. Digital platforms have fundamentally transformed business models and organizational structures across industries by creating ecosystems that connect producers, consumers, and complementors through technology-mediated interactions. Understanding the lifecycles of these DPs, from both business model and technical architecture perspectives, is essential for practitioners and researchers seeking to design, implement, and sustain successful platform ventures.

The primary research questions are:

1. How can a business model be systematically propagated into the development of platform applications and services that support it?
2. How can a business model be digitally transformed or innovated based on platform applications and services that create new opportunities for value creation?

The ultimate goal of this research is the development of a cross-layer integration framework for the development of a platform business model, applications and services.

This paper emphasizes two aspects of the proposed framework:

1. Lifecycle aspects (Chapter 3);
2. Cross-layer mapping of constructs between business models and application/service architectures (Chapter 4).

2 Problem Positioning and Literature Review

To position the research problem, the literature review synthesizes research across four interconnected domains: (1) platform business models and their representation through the Business Model Canvas (BMC); (2) platform application architectures modeled using Unified Modeling Language (UML); (3) service lifecycle models; (4) integrated approaches that bridge business and technical concerns.

2.1 Digital Platform Business Models and the Business Model Canvas

The origins of business model development can be traced back to early twentieth-century economic theories concerning how companies create and deliver value and generate profit through innovation and market mechanisms. Systematic approaches to modern business model development emerged in the early 2000s, when Osterwalder and Pigneur (2010) introduced the Business Model Canvas (BMC), a widely adopted framework for conceptualizing and communicating business models.

The BMC comprises nine building blocks: customer segments, value propositions, channels, customer relationships, revenue streams, key resources, key activities, key partnerships, and cost structure. In the context of digital platforms, the BMC has been adapted to capture the specific characteristics of multi-sided markets, network effects, and ecosystem orchestration.

Traditionally, business models serve as instruments of strategic management. They define and connect the fundamental structural elements of a business and support business innovation, including the redefinition of value propositions, revenue logic,

cost structures, and customer relationships. Foss and Saebi (2017) conceptualize business model innovation (BMI) as organizational transformation that does not necessarily require changes in the underlying technology or product. BMI is particularly relevant in conditions of disruptive change and digitalization, where competitive advantage increasingly depends on the configuration of activities rather than solely on resource ownership.

Beyond strategic management, the business model concept serves multiple purposes. In entrepreneurship and startup development, the business model functions as a dynamic learning and validation instrument rather than a static document. Lean and agile startup approaches emphasize iterative hypothesis testing and validation (Blank, 2013).

In operational management and organizational alignment, the business model performs an integrative function. Zott, Amit, and Massa (2011) define it as a system of interconnected activities aimed at creating value for customers.

From a financial perspective, the business model describes and operationalizes the logic of revenue generation and cost management. Morris, Schindehutte, and Allen (2005) argue that the business model reflects fundamental economic mechanisms, including revenue structure, profit margins, and market positioning.

In the domain of sustainable development, sustainable business models emphasize environmental and social values, resources, and activities. Bocken et al. (2014) propose archetypes of sustainable business models that support the transition toward circular economy principles and long-term sustainability.

In the context of digital transformation and platformization, the digital platform business model (DPBM) plays a central role in the monetization of innovation. Teece (2010) emphasizes that the business model defines the manner in which a company appropriates value from technological innovation. In platform companies, the business model incorporates mechanisms such as network effects, digital ecosystems, and multi-sided market structures.

Steffen et al. (2023) identified 30 platform success factors and integrated them into the BMC framework, demonstrating its utility for platform design. These factors span corporate value integration, platform value definition, and essential IT considerations, providing a comprehensive view of what differentiates successful platforms from failures.

However, in the context of DP development, the function of BMC is primarily descriptive and configurational, not processual or architectural. Since DPs are based on multilateral interactions, network effects, ecosystem orchestration and digital infrastructure, their elaboration requires methodological approaches that go beyond the static representation of business logic. Therefore, BMC is complemented by approaches such as "lean startup" and agile development, as well as modeling methods that provide a stronger foundation for the design and operationalization of platform services and applications. The Activity System Perspective (Zott and Amit, 2024) is the concept of a business model as an activity system, where the focus is on content, structure and transaction management. From the perspective of DPs, it enables the mapping of interdependent activities of different actors, explains how the configuration of activities generates network effects and analyzes governance mechanisms (control, access rules, algorithmic management).

Unlike traditional linear business models, DMss are based on network effects, ecosystem orchestration and multi-sided markets (Parker, Van Alstyne, 2023; Daradkeh, 2023).

2.2 Digital Platform Architectures and Applications

A digital platform is an Internet-based socio-technical system that leverages digital technologies to connect participants and facilitate interactions, communication, transactions, and value exchange. From a functional perspective, digital platforms represent scalable, user-centric ecosystems rather than single, isolated applications.

Digital platform applications (DPA) require alignment between the business model and the IT architecture. Enterprise modeling approaches, such as ArchiMate, provide layered representations of business, application, and infrastructure components. Compared to the BMC, which operates at the level of business logic,

ArchiMate offers formal links between business processes and software implementation (Lankhorst, 2017).

The Unified Modeling Language (UML) serves as a standardized notation for modeling software-intensive systems. UML diagrams—such as use case diagrams, class diagrams, sequence diagrams, activity diagrams, component diagrams, and deployment diagrams—enable architects to visualize, specify, construct, and document the structure and behavior of platform systems.

Scheer (2023) utilized UML to model platform composition, detailing technical architecture layers that support application composition. The work demonstrates how UML can link development lifecycle phases to technical models, facilitating communication between business stakeholders and technical teams. However, although UML provides strong support for technical modeling, it does not explicitly represent business model constructs such as value propositions or revenue mechanisms.

2.3 Design of Digital Platform Services

Service design is inherently multidisciplinary, integrating insights from service research, design, marketing, operations management, information systems, and interaction design (Prestes Joly et al., 2019). Contemporary service design is grounded in the value co-creation perspective, which challenges traditional provider-centric approaches and aligns with service-dominant logic (Yu et al., 2018). This perspective emphasizes resource integration and collaborative value creation between providers and customers.

The theoretical foundation of contemporary service design rests on the value co-creation perspective, which challenges traditional product-oriented and provider-centric approaches to service development. This perspective aligns with service-dominant logic, which emphasizes resource integration and collaborative value creation between providers and customers (da Fontoura Vieira et. al., 2025).

Recent research highlights the need for systemic and holistic approaches to service design. Cardoso et al. (2024) propose that service design should first be understood holistically, incorporate user-centered practices for structuring development steps,

and employ co-creation mechanisms with partners to make services transparent and meaningful to users.

2.4 Cross-Layer Transformation

Ensuring traceability and consistency between the business model and platform applications and services requires the transformability of concepts and constructs across layers. The Business Model Canvas frequently serves as the primary source framework for such mappings.

Key elements frequently mapped include:

- Value Propositions: Mapped to service specifications, application functions, and service design value propositions (Koyama et al. 2023; Meertens et. al., 2012; Fritscher and Pigneur, 2020)
- Customer Segments: Aligned with personas, user roles in UML, and ArchiMate business actors (Fritscher and Pigneur, 2020; Blaschke et al., 2016)
- Channels: Corresponded to service delivery channels, application interfaces, and customer touchpoints (Allweins et al., 2021), (Koyama et al. 2023)
- Key Activities: Mapped to business processes in ArchiMate, UML activity diagrams, and service blueprint activities (Meertens et al., 2012; Fritscher and Pigneur, 2020)
- Key Resources: Aligned with ArchiMate resources, UML components, and service design resources (Meertens et al., 2012), (Lê, 2016). Platform-specific extensions like the Platform Canvas introduce additional constructs such as platform sides, network effects, and ecosystem governance mechanisms, which require specialized mappings to architectural elements (Allweins et al., 2021).

3 Lifecycle Aspects of the Cross-Layer Development of Business Models, Applications and Services

There is a clear need for integrated lifecycle approaches that align business model evolution with technical architecture and application development, as well as the design of platform services. Existing studies predominantly focus on either business model lifecycles or technical lifecycles, with limited integration. The need for integrated approaches is particularly acute in platform contexts, where business model decisions (e.g., pricing strategies, partnership models) have direct implications for technical architecture (e.g., API design, data governance), and vice versa. The following sections examine lifecycle models in each domain before returning to the integration challenge.

The proposed integration framework includes a lifecycle model and process patterns that define the phases, processes, and activities of platform business model development, as well as the development of the platform itself and its associated applications and services.

The lifecycle approach is bidirectional. Development may proceed top-down, from an existing or newly designed platform business model toward the development of services and applications. Alternatively, development may proceed bottom-up, from a platform idea or prototype toward the formulation or transformation of the business model. Iterative development cycles analogous to software engineering methodologies may be applied.

Figure 1. shows the development of a business model. It starts with a phase of inspiration and idea development. Various approaches and methods of Design Thinking, a discipline that has been developing since the middle of the last century, are applied here. General approaches and methods are specialized in service design. Service Design Thinking (SDT) is a human-centered, co-creative, iterative approach that treats services as end-to-end experiences across touchpoints, frontstage/backstage activities, actors, and supporting systems (Stickdorn, Schneider, 2012). It helps co-design across multiple participant groups (e.g., consumers, providers, partners).

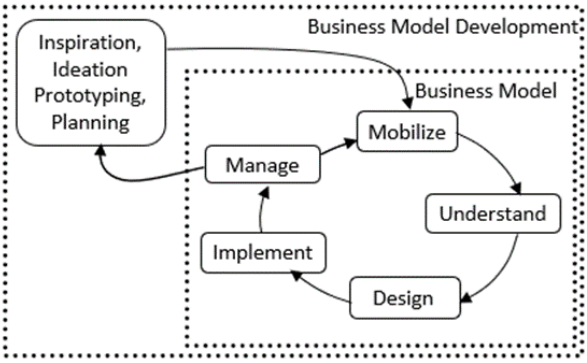

Figure 1: Business Model Development Cycle
Source: Own

According to Alexander Osterwalder and Yves Pigneur (2010) and BMC, the business model development cycle includes mobilization, understanding, design, implementation, and management phases. The transition from the business model layer to the service design layer entails the development of platform applications and customer-centric service models focused on user experience.

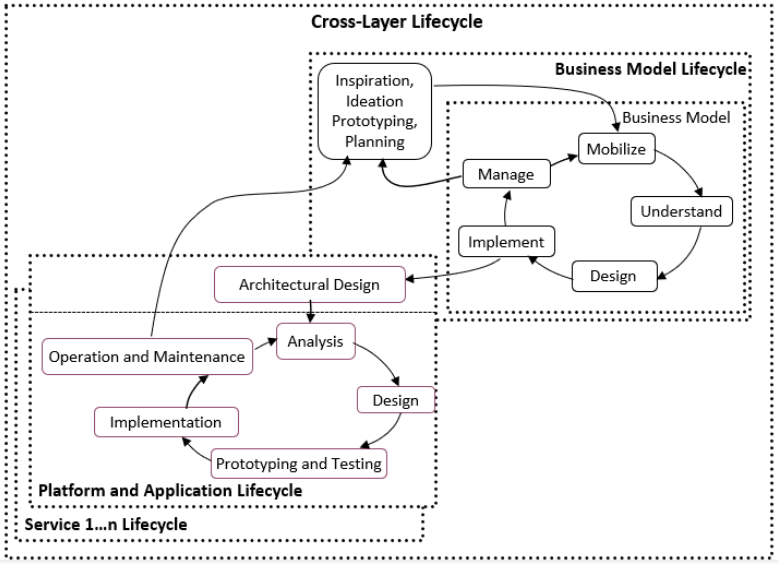

Figure 2: Cross-Layer Lifecycle
Source: Own

Platform applications and services are the implementation of platform business models, as shown on the Cross-Layer Lifecycle (Figure 2).

Methodological approaches of different authors or companies, differ in the number, names and content of the phases, but it is mostly possible to map Lifecycles among different methodological approaches.

Architectural design of digital platform applications encompasses the definition of system components, interfaces (APIs), data flows, infrastructure, security mechanisms, and scalability strategies. Analogously, the architectural design of a platform service encompasses defining the system structure, key components, interfaces (APIs), data flows, security mechanisms, infrastructure, and scalability strategies to ensure reliable, secure, and efficient service delivery.

Platform application and service lifecycles typically include analysis, design, prototyping and testing, implementation, and operation and maintenance. All these phases are carried out in accordance with the chosen methodology of business modeling, and the development of software and digital services.

Development is typically incremental and iterative, in accordance with agile methodologies that combine best practices from systems and software engineering.

The overall cross-cycle development spans three distinct lifecycles, each with its own focus and progression. The development can be conducted in an incremental and iterative manner, consistent with agile development principles.

The figure does not show who initiates development cycles and government mechanisms, as well as the layers of other management processes, such as management of changes, configuration, releases, quality, etc.

Changes in the business model are most often driven by the strategic management of the organization. However, in cyclical development, the impetus for business model development can come from the operation and maintenance phases of the development cycle of platform applications or services, where evaluation can indicate that modifications are needed within the BM. The new development cycle

may also be triggered by technological advancements, the emergence of new tools and platforms, as well as by the redesign or improvement of platform services.

4 Cross-Layer Mapping of Constructs Between Business Models, Application and Service Architectures and Models

Business models, such as those based on the BMC framework, provide high-level representations of value creation, delivery, and capture. Since key platform service design models contain similar constructs, such as value, activity, resource, partner communication, only elaborated in more detail, the conversion of BM constructs into SDM constructs will not be presented in more detail in this article.

However, platform application development requires formal modeling constructs capable of specifying system structure, behavior, and interactions. Unified Modeling Language (UML) offers such formalization through standardized diagrams (e.g., use case, class, sequence, activity, and component diagrams).

This section analyzes how key BMC elements (Value Proposition, Customer Segments, Channels, Customer Relationships, Key Partners, Key Resources, and Key Activities) are transformed into UML artifacts during platform application development (Fritscher and Pigneur, 2020). The objective is to clarify which business-level concepts serve as direct inputs for UML modeling and how they are translated into system-level abstractions.

Table 1 summarizes the mapping of BMC constructs to Platform Architecture layer constructs, ArchiMate constructs and UML concepts with a description of the transformation.

Some constructs are directly transformed from BMC to UML (e.g., Customer Segments → Actors; Key Activities → Activity Diagrams; Key Resources → Domain Classes).

Table 1: Mapping of BMC constructs to Platform Architecture, ArchiMate constructs and UML constructs and concepts

BMC Construct	Platform Architecture Layer Construct	ArchiMate Construct	UML Concept(s)	Transformation into UML (Detailed Description)
Value Proposition	Platform Service / Platform Capability	Business Service, Value, Capability	Use Case, Service Interface, Component	Value propositions are decomposed into platform services. Each service becomes a Use Case. Services exposed via APIs are modeled as UML interfaces. At the structural level, they become Components implementing those interfaces.
Customer Segments	Platform Actor Roles (e.g., Provider, Consumer, Admin)	Business Actor, Business Role	Actor (Use Case Diagram), Actor Generalization	Customer segments are abstracted into system roles. Each role becomes a UML Actor. If multiple subtypes exist (e.g., Premium User), Actor generalization hierarchies are introduced.
Channels	Interaction Layer (Web App, Mobile App, API Gateway)	Application Interface, Application Component	Boundary Class, Interface, Component	Channels are operationalized as interaction interfaces. In UML, they become boundary classes or provided interfaces in component diagrams. They define system entry points.
Customer Relationships	Engagement Mechanisms (Subscription logic, Reputation System)	Business Process, Application Function	State Machine Diagram, Sequence Diagram	Relationship mechanisms (e.g., subscription lifecycle) are transformed into state machines describing user states and transitions. Interaction rules are formalized in sequence diagrams.
Key Activities	Core Platform Processes (Matchmaking, Payment Processing)	Business Process, Application Function	Activity Diagram, Sequence Diagram	Activities become workflows. Each key activity is modeled as an activity diagram. Interactions among components during execution are captured in sequence diagrams.
Key Resources	Platform Assets (Data, Algorithms, Infrastructure)	Data Object, Application Component, Artifact	Class Diagram, Component Diagram	Data resources become domain classes. Algorithms become service components. Infrastructure becomes deployment nodes (in Deployment Diagrams if extended).
Key Partners	External Systems / Ecosystem Actors	Business Actor, Application Collaboration	External Actor, External Component, Dependency	Partners (e.g., Payment Provider) become external actors or components. Dependencies are modeled explicitly in component diagrams and interaction flows in sequence diagrams.

Source: own

Other constructs require derived transformations (e.g., Value Propositions → Use Cases; Customer Relationships → State/Interaction Models; Channels → Interface Definitions).

Revenue streams and cost structures function primarily as contextual inputs for requirements engineering and are not directly mapped to UML structural elements.

A major challenge for future research lies in the development of computer-supported tools enabling systematic cross-layer transformation between BMC components, service design artifacts, and UML architectural models.

5 Conclusion

Digital platform development requires systematic alignment between business logic, service design, and technical architecture. The proposed cross-layer integration framework aligns lifecycle phases and formalizes transformation rules between business model constructs and architectural artifacts. By bridging strategic and technical perspectives, the framework enhances methodological rigor and practical applicability. Future research should focus on meta-model formalization and computer-supported transformation tools to enable automated traceability across layers.

The proposed framework is founded on the cyclical and iterative co-development and consolidation of three interrelated domains: (1) business model development through the application of the Business Model Canvas framework, (2) service development employing selected service design methods, and (3) platform (application) development using Unified Modeling Language. The primary objective of the framework is to reduce the methodological discontinuities among these domains by enabling the systematic integration and cross-utilization of artifacts generated within each domain throughout the development process. In addition, the framework seeks to establish coordinated and synchronized development cycles, thereby facilitating the continuous alignment of business, service, and technological perspectives during the iterative refinement of the overall system.

The proposed framework reduces discontinuities between strategic, operational, and technical modeling domains by enabling cyclical co-development and traceable artifact propagation across business model, service design, and platform engineering activities.

Acknowledgment

The authors would like to acknowledge the support given by the European Commission through the Interreg Danube Region Programme Digital services for Circular Economy - a Toolbox for Regional Developers & SME (DECIDE).

References

Allweins, M. M., Proesch, M., & Ladd, T. (2021). The platform canvas—conceptualization of a design framework for multi-sided platform businesses. *Entrepreneurship Education and Pedagogy*, *4*(3), 455-477.

Blank, S. (2013). Why the lean start-up changes everything. *Harvard Business Review, 91(5)*, 63–72.

Blaschke, M., Cigaina, M., Riss, U. V., & Shoshan, I. (2016). Designing business models for the digital economy. In *Shaping the digital enterprise: Trends and use cases in digital innovation and transformation* (pp. 121-136). Cham: Springer International Publishing

Bocken, N.M.P., Short, S.W., Rana, P., & Evans, S. (2014). A literature and practice review to develop sustainable business model archetypes. Journal of Cleaner Production, 65, 42–56.

Cardoso, A. M. K., Canciglieri Junior, O., & Benitez, G. B. (2024). Service design: a critical examination and future research directions in servitization literature. *Journal of Business & Industrial Marketing, 39*(9), 1950-1968.

Daradkeh, M. (2023, May). Exploring the boundaries of success: A literature review and research agenda on resource, complementary, and ecological boundaries in digital platform business model innovation. In *Informatics* (Vol. 10, No. 2, p. 41). MDPI.

da Fontoura Vieira, J. F., Echeveste, M. E. S., Tinoco, M. A. C., Marcon, A., Marcon, É., & Lermen, F. H. (2025). Enhancing value cocreation orientation in service innovation: a new service development and service design integrated process. *Humanities and Social Sciences Communications*, *12*(1), 31.

Foss, N.J., & Saebi, T. (2017). Fifteen years of research on business model innovation. Journal of Management, 43(1), 200–227.

Fritscher, B., & Pigneur, Y. (2020). A visual approach to business IT alignment between business model and enterprise architecture. In *Sustainable Business: Concepts, Methodologies, Tools, and Applications* (pp. 543-566). IGI Global.

Koyama, S., Tsuji, Y., Masuda, R., Onoda, H., Ono, H., Natsuyama, M., ... & Yamamoto, S. (2023). A Technique to integrate service business models with archimate. *Procedia Computer Science*, *219*, 479-485.

Lankhorst, M. (2017). Enterprise Architecture at Work: Modeling, Communication and Analysis. Springer.

Lê, L.S., (2016). Enterprise Architecture. Mapping of BMC, DEMOand ArchiMate. *Master Thesis Leiden Institute of Advanced Computer Science (LIACS), theses.liacs.nl/pdf/Le-Minh-non-confidential.pdf*

Meertens, L. O., Iacob, M. E., Nieuwenhuis, L. J., van Sinderen, M. J., Jonkers, H., & Quartel, D. (2012). Mapping the business model canvas to ArchiMate. In *Proceedings of the 27th annual ACM symposium on applied computing* (pp. 1694-1701)

Morris, M., Schindehutte, M., & Allen, J. (2005). The entrepreneur's business model: Toward a unified perspective. Journal of Business Research, 58(6), 726–735.

Osterwalder, A., & Pigneur, Y. (2010). *Business model generation: a handbook for visionaries, game changers, and challengers* (Vol. 1). John Wiley & Sons.
Parker, G., & Van Alstyne, M. (2023). Platforms: Their structure, benefits, and challenges. In *Introduction to Digital Humanism: A Textbook* (pp. 523-542). Cham: Springer Nature Switzerland.
Prestes Joly, M., Teixeira, J. G., Patrício, L., & Sangiorgi, D. (2019). Leveraging service design as a multidisciplinary approach to service innovation. *Journal of service Management*, *30*(6), 681-715.
Scheer, A. W. (2023). Application composition platform architecture. In *Digital Business Engineering* (pp. 47-58). Springer. https://doi.org/10.1007/978-3-658-42483-1_4
Stickdorn, M., & Schneider, J. (2012). *This is service design thinking: Basics, tools, cases*. John Wiley & Sons.
Steffen, M., Hunke, F., Hagen, S., & Schlereth, A. (2023). Success factors of digital platform design. *ITM Web of Conferences*, 51, 05001. https://doi.org/10.1051/itmconf/20235105001
Teece, D. J. (2010). Business models, business strategy and innovation. Long Range Planning, 43(2–3), 172–194.
Yu, E., & Sangiorgi, D. (2018). Service design as an approach to implement the value cocreation perspective in new service development. *Journal of Service Research*, *21*(1), 40-58.
Zott, C., Amit, R., & Massa, L. (2011). The business model: Recent developments and future research. Journal of Management, 37(4), 1019–1042.
Zott, C., & Amit, R. (2024). Business models and lean startups. Journal of Management, 50(8), 3183-3201.