This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

Quantitative Assessment of Service Pattern: Framework, Language, and Metrics

Meng Xi, Jianwei Yin, Jintao Chen, Ying Li, and Shuiguang Deng

Abstract—For modern service industry (MSI), service pattern is a service provision approach to support the realisation of business model that involves participants from various domains and organizations. A comprehensive description and quantitative assessment of service patterns is of great significance for optimizing the organizational cooperation process in MSI and improving the competitiveness of enterprises. However, most relevant studies on service patterns stay at the level of business processes and qualitative analysis, lacking a comprehensive description of data, resources, and value exchanges among participants. Studies related to pattern assessment focus more on QoS (Quality of Service) rather than consideration of the utility of multi-participant collaboration. Hence, two issues need to be tackled for future development of MSI: a) How to systematically describe and distinguish service patterns with the same business processes. b) How to assess and compare service patterns quantitively and comprehensively.

In this work, we propose a service pattern assessment framework which consists of two parts. As part one, we complement the service pattern description language (SPDL) with extended elements and observable attributes to empower it with quantitative analysis, namely Quantitative SPDL (SPDL-Q). In part two, a set of service pattern assessment metrics are designed to assess not only the quality of the services but also the cooperation efficiency of the participants and the orchestration effect of the service patterns elements. The proposed framework was then further validated by a case study, of which four E-commerce service patterns were studied to reveal their evolvement processes. Correlation experiments were also performed to identify the pattern features that have the greatest impact on each metric, so to provide guidance and suggestions for pattern design. Finally, the innovation and significance of the work are outlined and discussed.

Index Terms—service pattern, modern service industry, business process, process design, process reconstruction.

1 Introduction

To meet the requirement of current industrialization development, MSI (Modern Service Industry) has been proposed which was built on information technology and modern management concepts. MSI is characterized by the need to coordinates data, resources, values, services, and participants from various domains and organizations. For instance, finance, logistics, seller, and customer and their services, resources, etc. together constitute a typical MSI business, E-commerce. To have a comprehensive description of all these elements and collaborations, service pattern is proposed.

Service pattern describes the method of service coordination, data transmission, resource allocation and value exchange among different participants in MSI. It focuses on collaboration and exchange of resources and values between multiple domains and cross-organizations. For the emerging MSI represented by E-commerce and Internet healthcare service, the design and assessment of service patterns have become important factors and means of determining business competitiveness [1]. Under those circumstances, relevant studies on service modeling and business models have been carried out.

Service modeling and business process management have been studied for long in the field of service computing. At present, most of the business process models can be grouped into two types: activity-centric modeling methods represented by BPMN (Business Process Modeling Notation) [2] and BPEL (Business Process Execution Language) [3], and data-centric modeling methods represented by artifact-centric models [4]. In order to depict the interactions between different participants or organizations, collaboration and choreography were proposed and studied as well [5], [6], [7]. However, existed methods focus more on arrangements of services and information interactions, but lack consideration of multi-domain collaboration and resource value exchange, thus cannot meet the need for systematic modeling, quantitative evaluation, and analysis of service patterns.

As for business models, they have been put forwarded by researchers in economics and management areas. Research on business models contributes mainly in three aspects: conceptual model, typology method, and industry model and typical cases. The conceptual model refers to abstraction and formalization for business models. The typology methods mainly focus on qualitative studies and classification methodologies. The research on industry models and typical cases are mostly empirical studies based on individual cases. Most of those research are qualitative analysis and lack computable modeling methods or quantitative analysis theory.

Service patterns of E-commerce have been evolved since

This work is supported by the National Key Research and Development Program of China(No.2017YFB1400601), the National Natural Science Foundation of China under Grant (No.61825205, No.61772459, No.U20A20173), and the National Science and Technology Major Project of China(No.50-D36B02-9002-16/19). (Corresponding author: Jianwei Yin and Shuiguang Deng)

- *Jianwei Yin is with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China.
E-mail: zjuyjw@cs.zju.edu.cn*
- *Shuiguang Deng is with the College of Computer Science and Technology, Zhejiang University, Hangzhou, China.
E-mail: dengsg@cs.zju.edu.cn*

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

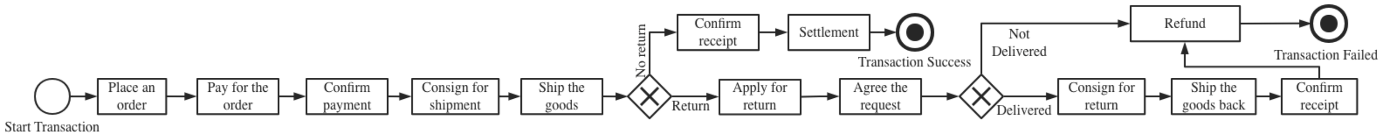


Fig. 1. The middleman pattern, platform pattern, and proprietary pattern share the same business process, which make it difficult to analyse the differences among the three.

the last few decades, among which four patterns are typical, i.e., middleman pattern, platform pattern, proprietary pattern, and new retail pattern. These four patterns are all based on business activities centered on the exchanges of goods by means of information network technology, which leads to similar workflow designs. The middleman pattern, platform pattern, and proprietary pattern even share the same workflow in most cases (see Fig. 1). That leaves two challenges for further development of E-commerce: *a) How to systematically describe the service patterns and distinguish the ones with the same business processes. b) How to assess and compare service patterns quantitively and comprehensively.*

In previous work, we have analysed and modeled the service pattern in detail and proposed the service pattern description language (SPDL) [8]. It partially overcomes the first challenge, but cannot provide the solution of the second. In this work, we expand the notations of SPDL and propose a Quantitative SPDL (SPDL-Q), based on which a service pattern assessment framework is built. Innovation in this work are reflected in the following aspects:

- A framework for service pattern assessment is proposed. It can transform service pattern computing from ambiguous and qualitative cognition to precise and quantitative analysis;
- We complement SPDL to SPDL-Q with extended elements and observable attributes. SPDL-Q can quantify service patterns from four perspectives: workflow, data flow, resource flow, and value flow;
- A set of service pattern assessment metrics are designed to evaluate the impact of different participants cooperation and service coordinations;
- Outline the benefits of the advantages created by platform pattern, proprietary pattern, and new retail patterns compared to middleman E-commerce pattern through a data-driven case study;
- Pattern features and correlation experiments are extracted and conducted to inspect the most influential factors on pattern metrics, which could be enlightening to relevant practitioners.

The rest of this paper is organised as follows. Section 2 briefly reviews previous research. Section 3 outlines the overall assessment framework. Section 4 introduces the SPDL-Q notations. Section 5 introduces the service pattern assessment metrics. Section 6 is a data-driven case study that reports our experimental results. Section 7 introduces eleven pattern features and studies the factors which influence each metric mostly. Section 8 carries out the comparison between SPDL-Q and other service modeling methods. Section 9 is the conclusion of the work.

2 Related Work

2.1 Service Modeling and Analysis

As one of the key technologies in the field of software engineering and service computing, service modeling has attracted attention from both industry and academics. The traditional business process modeling methods, among which BPMN and BPEL are the most widely used ones, mainly focus on service operation and process management. Based on those methods, extensions like semantics, human tasks, and data quality were also introduced [9], [10], [11]. In addition, efforts were put to further abstract and simplify the BPMN process models [12], [13].

Artifact-centric model is another typical way of modeling. It was proposed and defined as four dimensions of business process: business entity, lifecycle, service, and association [4], [14]. Artifact-centric models create a symbiosis between data and processes, which are available for model checking, verification, and validation [15]. The configurable modeling framework for artifact-centric business processes was proposed as well to support the business evolution [16].

Although current available approaches can describe business processes and information interactions nicely, they lack consideration of multi-domain collaboration and resource value exchange between participants. Therefore knowledge and technologies of business models and service patterns need to be introduced.

2.2 Business Model

Business model is a kind of holistic approach towards explaining the mechanisms of business [17], [18]. At present, the study on business model mainly focuses on the conceptual model, typology method, and industry model and typical cases. Conceptual model research mainly studies the concepts and composition of the business models. Amit et. al. did a systematic study on a theoretical basis and put forward the concept of business model integration for the first time [19]. After that, Leshub et. al. proposed a framework for modeling business model with UML [20].

Typology method research is to divide the business model into several different categories for classification. Madlberger et. al. established a business model analysis framework that includes three transaction stages of information acquisition, achievement, and implementation [21]. Liu et. al. conducted classification research on the business model of urban agriculture [22]. Furthermore, typology research on business models can also help to classify organizations and position how they create and capture values [23]. Model typology was developed for conceptual understanding of circular business models as well [24].

Industry models and typical case studies mainly work on specific cases to study the characteristics and contents

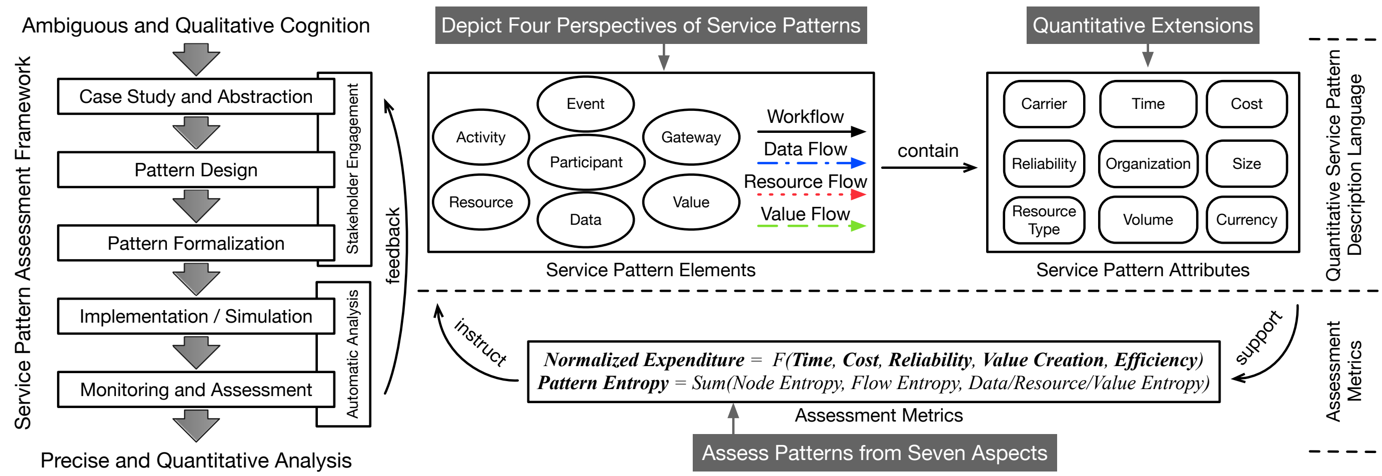


Fig. 2. The overall assessment framework.

of a specific business model. Many researchers applied the existing theoretical tools to some specific industries, such as personal entertainment services and mobile communities, and carried out applied research [25], [26]. In addition, Beynon et. al. analysed and discussed the business model of E-business and E-commerce and visualised them [27].

However, most of existing business model researches are qualitative analysis, and lack of deterministic and computable modeling methods. As a result, it could be hard, by those means, to analyse the impact of specific activities and objects in the business processes and guide the assessment and optimization of the service patterns.

2.3 Service Pattern

Different from the previous two types of patterns, service pattern was invented within a completely new strategy. Researchers have various understanding towards the definition of service patterns. Liang et. al. proposed the concept of service pattern from the perspective of users, and divided the patterns into three layers: user requirements layer, template layer, and instance layer [28]. Li et. al. considered the service pattern as a workflow-based exchange of resources and proposed a domain specific language to help enterprise managers to analyse basic business strategies [29]. Moreover, some other service pattern models and description languages were proposed to help with the reuse and value analysis of business processes [8], [30].

To improve service pattern strategies, research have been conducted from different perspectives. Duan et. al. proposed methods to improve the reusability and optimize the strategies of service patterns [31]. Yin et. al. proposed an economic analyse method on service patterns to support modifications towards better benefit [32]. To study the profitability of a business, service pattern with resource usage situation is extracted by the service description and system log [33].

However, the service patterns mentioned above could be considered merely as an abstraction or extension of the business processes. The relevant analysis methods also focused only on business process management and service quality analysis, which are not enough to support the conduction of comprehensive and quantitative analysis of service patterns on multi-domain cooperation and resource value exchange. In this work, we propose a service pattern assessment framework to depict and assess not only the business processes and services but also the cooperation of the participants and the orchestration of the service patterns elements.

3 Service Pattern Assessment Framework

For a better understanding of the content below, the overall assessment framework is outlined in Fig. 2. The left part of Fig. 2 shows the five phases of the framework. The first phase is to study the case and abstract the service elements. The second phase is to design the service pattern through workflow, data flow, resource flow, and value flow. The third phase is to formalize the service pattern through quantitative attributes. The fourth phase is to generate pattern data through practice implementations or simulation experiments. Finally, we monitor and assess the generated data, and provide feedback on the performance of the service pattern.

Among them, the former three phases are carried out by stakeholders on the basis of SPDL-Q. As shown in the top right corner of Fig. 2, SPDL-Q can describe the service patterns by workflow, data flow, resource flow, and value flow. Further quantitative analysis is supported by the extended service pattern attributes. The details about SPDL-Q are elaborated in Section 4.

The last two phases can be completed automatically through the assessment metrics designed as shown in the lower right corner of Fig. 2. To put forward an overall evaluation of the service pattern, the normalized expenditure is proposed based on the metrics from five aspects including time, cost, reliability, value creation, and efficiency. Pattern entropy is proposed in order to measure the chaos degree of multi-domain element coordination. The calculation methods and connotations of all metrics are given in Section 5.

The framework has been implemented to a service pattern-oriented computing platform developed by our team, on which the case study in Section 6 is deployed.

4 SPDL-Q: Service Pattern Description Language for Quantitative Computation

In this section, we briefly introduce service pattern description language (SPDL), the predecessor of this work, and discuss the differences and innovations made in SPDL-Q. Then we illustrate the elements of SPDL-Q in detail.

4.1 Service pattern description language

SPDL was proposed in 2016 to meet the needs of resource and data interaction management in modern service industry by depicting service patterns from three perspectives: workflow, data and resource [8]. It formalizes data and resource in process more comprehensively and support better analysis of data and resource exchange compared to classical business process models.

The SPDL-Q extends SPDL via the characterization of the run-time performance expectations and value flow. In particular, the SPDL-Q complements the SPDL as follows:

- Runtime properties (e.g., time, cost, and reliability) are added for activity, gateway and event to enable model designers to simulate and assess the model before deployment.
- The definition of service pattern has been extended, from an activity abstraction in SPDL to a convergence of participants, workflow, value flow, data flow and resource flow in SPDL-Q.
- Value is modeled independently, and the transformation between value and resource is considered.
- The types of participants are considered to depict the impact of introducing collaboration between participants from different organizations and domains.
- The observable attributes of data, resources and value are considered and modeled to support the estimation of communication, transmission and transaction efficiency.

4.2 Service elements in SPDL-Q

In this section, the definitions of the elements in SPDL-Q are given in a top-down way along with corresponding examples taken from middleman pattern of E-commerce (see Fig. 5a).

The service pattern Γ is an abstraction of business relationship among participants from four perspectives: workflow, data flow, resource flow and value flow (see Definition 1). Tab. 1 illustrates the service pattern from the example of middleman E-commerce pattern. As defined, the service pattern consists of three data flows, two resource flows, and five value flows based on the workflow between the four participants. Participants form an association through a workflow by declaring the nodes on which they operate. The data flows, resource flows, and value flows start and end at nodes in the workflow. Therefore they can also express the exchanges of data, resources, and values between participants.

Defintion 1. *A service pattern is a 6-tuple* $\Gamma = (id_\Gamma, \mathbb{P}, w, \mathbb{D}, \Theta, \mathbb{V})$*, where* id_Γ *is the identifier,* $\mathbb{P}$ *is the set of participants,* w *is the workflow,* $\mathbb{D}$ *is the set of data flows,* Θ *is the set of resource flows, and* $\mathbb{V}$ *is the set of value flows.*

TABLE 1
Service Pattern Example: Middleman E-commerce Pattern

Middleman E-commerce Pattern

$\mathbb{P}$:Participants	Seller, Customer, Logistics Company, Financial Institution.
w :Workflow	Online transaction workflow.
$\mathbb{D}$:Data Flows	Seller logistics data flow, Consumer logistics data flow, Return logistics data flow.
Θ :Resource Flows	Transaction resource flow, Return resource flow.
$\mathbb{V}$:Value Flows	Advance payment value flow, seller settlement value flow, logistics settlement value flow, return logistics value flow, return refund value flow.

TABLE 2
Workflow Example: Online Transaction Workflow

Online Transaction Workflow

$\mathbb{A}$:Activities	Place an order, Pay for the order, Confirm Payment, Consign for shipment,...
$\mathbb{G}$:Gateways	Whether to return, Whether need to send back.
$\mathbb{E}$:Events	Start transaction, Transaction success, Transaction failed.
$\mathbb{F}$:Connectors	{Flow_0s2tw9v, Start transaction, Place an order}, {Flow_0jrv32v, Place an order, Pay for the order},...

The workflow represents the business process based on which the data, resource, and value could be exchanged. There are three kinds of node in the workflow, namely activity, gateway, and event, which are linked by a set of connectors (see Definition 2).

Tab. 2 illustrates an example of online transaction workflow. The activities are used to represent the operations performed by the participants, such as place an order. The gateways can route workflows to different branches depending on the situation and eventually generate events for successful or failed transactions. All activities, gateways, and events are connected by flows.

Defintion 2. *A workflow is a 5-tuple* $w = (id_w, \mathbb{A}, \mathbb{G}, \mathbb{E}, \mathbb{F})$*, where* id_w *is the identifier,* $\mathbb{A}$ *is the set of activities,* $\mathbb{G}$ *is the set of gateways,* $\mathbb{E}$ *is the set of events, and* $\mathbb{F}$ *is the set of connectors.*

The data flow depicts the communication between participants and consists of name, type, size, and a connector (see Definition 3). The data type could be any common or custom data formats, like JSON and XML. The connector involved is used to depict the producer and consumer of data.

Tab. 3 illustrates a seller logistics data flow. That data flow carries 2048byte JSON formatted seller logistics data, which is generated in the "Ship the goods" activity and used in the "Agree the request" activity.

Defintion 3. *A data flow is a 5-tuple* $d = (id_d, n_d, \kappa_d, \epsilon_d, f) \in \mathbb{D}$*, where* id_d *is the identifier,* n_d *is the name,* κ_d *is the data type,* ϵ_d *is the size,* f *is the connector to indicate the transference of the data.*

The resource flow can illustrate physical assets that are delivered for trading between two participants on the basis of the workflow (see Definition 4). For a resource flow, the

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

TABLE 3
Data Flow Example: Seller Logistics Data Flow

Seller Logistics Data Flow

n_d :Data Name	Seller logistics data
κ_d :Type	JSON
ϵ_d :Size	2048 byte
f :Connector	{Flow_1cv14qy, Ship the goods, Agree the request}

TABLE 4
Resource Flow Example: Transaction Resource Flow

Transaction Resource Flow

n_θ :Resource Name	Goods
κ_θ :Type	Clothes
ϵ_θ :Weight	1000g
f :Connector	{Flow_0zgwtcp, Consign for shipment, Confirm receipt}

type κ_θ could be food, clothes, etc. The ϵ_θ would be the weight of the resource.

Tab. 4 illustrates a transaction resource flow. The resource flow represents a 1000g shipment of the clothing class, sent in the "Consign for shipment" activity and received in the "Confirm receipt" activity.

Defintion 4. *A resource flow is a 5-tuple θ = (id_θ, n_θ, κ_θ, ϵ_θ, f) $\in \Theta$, where id_θ is the identifier, n_θ is the name of the carried resource, κ_θ is the type of the resource, ϵ_θ is the weight, f is the connector to indicate the transference of the resource.*

The value flow refers to the money transaction between participants under the service pattern. It includes the value name, the currency used, volume, and the connector (see Definition 5). The connector f in a value object indicates the nodes where the value is provided and received. The transferences of the data, resource, and value occur only when all nodes of the connectors have completed execution.

Tab. 5 illustrates an advance payment value flow. The value flow represents the payment for goods of 200 CNY, which is paid in the "Pay for the order" activity and confirmed in the "Confirm the order" activity.

Defintion 5. *A value flow is a 5-tuple v = (id_v, n_v, κ_v, ϵ_v, f) $\in \mathbb{V}$, where id_v is the identifier, n_v is the name, κ_v is the currency type, ϵ_v is the volume, f is the connector to indicate the transference of the value.*

The connector can be used to describe the business logic in workflow, the transference of data, the transmission of resources, and the exchange of value. The source and target node of a connector should be linked to the activities, gateways, and events with respect to Γ (see Definition 6).

TABLE 5
Value Flow Example: Advance Payment Value Flow

Advance Payment Value Flow

n_v :Value Name	Payment for goods
κ_v :Type	CNY
ϵ_v :Volume	200
f :Connector	{Flow_06nc2to, Pay for the order, Confirm the order}

TABLE 6
Participant Example: Customer

Customer

κ_p :Type	Individual
$\mathbb{ID}_a$:Activities	Place an order, Pay for the order, Confirm receipt, Apply for return, Consign for return.
$\mathbb{ID}_g$:Gateways	Whether to return.
$\mathbb{ID}_e$:Events	Start transaction.

Defintion 6. *A flow connector is a triple f = (id_f, id_s, id_t), where id_f is the identifier, id_s and id_t are identifiers of source node and target node, respectively. A source node or target node could be one of the activities, events, or gateways.*

For the definition of participants, as shown in Definition 7, in addition to declaring the activities, gateways, events that they need to perform, participants own an attribute of type. The participant type κ_p is mainly used to represent the organization or superior of the participants and is allowed to refer to another participant. This enables the relationships between participants to form a tree structure to determine the domain of participants in a complex service pattern. For the participants who are no subordinate to another one, their types should be the reserved word "individual".

Tab. 6 shows an example of a consumer participant. The consumer participates in five activities, one gateway, and one event. In this example, since the type of both consumer and seller is "Individual", it obvious to tell that the shown middleman E-commerce pattern is also a C2C pattern (Consumer to Consumer).

Defintion 7. *A participant is a 5-tuple p = (id_p, κ_p, $\mathbb{ID}_a$, $\mathbb{ID}_g$, $\mathbb{ID}_e$) $\in \mathbb{P}$, where id_p is the identifier, κ_p is the type, and $\mathbb{ID}_a$, $\mathbb{ID}_g$, and $\mathbb{ID}_e$ are the identifiers of the activities, gateways, and events the participant participate in. Specifically, κ_p is mainly used to represent the organization or superior of the participants.*

As for the workflow nodes, i.e., event, activity, and gateway, we follow the basic definition in SPDL [8]. In short, the events mark the start, intermediate and end states of the service patterns. Each activity represents an API, a web service, or a functional component. The gateway is mainly responsible for routing and forwarding complex requests for web services.

To support the quantitative calculation to a service pattern, we extend the attributes of event, activity and gateway. The extended attributes include the carrier $m \in \mathbb{M}$, time $\mathcal{T}$, cost $\mathcal{C}$, and reliability $\mathcal{R}$. Among them, m and $\mathbb{M}$ represent the platform the nodes are deployed on and the full collection in the service pattern. $\mathcal{T}$ and $\mathcal{C}$ indicate the duration and money that the node may take in a single run. $\mathcal{R}$ is the probability of the node running successfully.

5 Service Pattern Assessment Metrics

Since the assessment of service patterns involves the exchange of data, resources, and values among multi-domain organizations, service pattern assessment metrics were designed to have time, cost, reliability, efficiency, value creation, normalized expenditure, and pattern entropy as participants collaboration. It is aimed to be able to comprehensively assessing and estimating service pattern based on

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

SPDL-Q during the design phase. Therefore, it is necessary to calculate the expected value of each metric based on the workflow in the service pattern.

In each assessment, we simulate progressively one execution of the pattern starting from the start event and estimate the metrics. The simulation is implemented by recursive algorithms to resolve the calculation to complex and nested workflow structures. When performing a recursion, there are four cases to consider: sequential case, parallel case, switch case, and base case. The calculate formulas of the metrics in each case are introduced hereunder.

5.1 Time

Unlike the traditional single-user single platform pattern, different web services are deployed on different servers and executed by different participants in one MSI business. Therefore, the service pattern time includes not only the web service execution time but also the cooperation time introduced by multi-participant cooperation and the interaction time introduced by multi-platform interaction. On the one hand, if the source and target participants of a connector f in the workflow are different, extra time will be consumed in the collaboration process of activity handover. For example, in the middleman E-commerce pattern, due to the traffic time required for the courier pickup, it usually takes hours between the seller "consign for shipment" and the logistics company "ships the goods", though they are consecutive activities. On the other hand, if the carriers of source and target are different, more information transmission time would be introduced. E.g., an invocation delay is approximately 25ms under AWS (Amazon Web Services) environment, which is usually lower than the normal cross-cloud connections [34].

To calculate the service pattern time $\mathcal{T}$, we introduce a recursive function $\mathbf{T}(n_i)$ to calculate the execution time required from node n_i to the end. The recursive logic varies depending on the type of node n_i. When n_i is an event or an activity (sequential case), the recursive computation is performed by Eq. 1a, where $\mathcal{T}_{n_i}$ and $\mathcal{R}_{n_i}$ are the time and reliability of the workflow node n_i resp., f_{i+1} and n_{i+1} are the adjacent subsequent connector and node of n_i resp., and $\mathcal{T}_{f_{i+1}}$ is the execution time of connector f_{i+1}. When n_i is a parallel gateway with K branches, $\mathbf{T}(n_i)$ is the sum of the time of node n_i and the largest time of all K branches (see Eq. 1b), where f_{i+k} and n_{i+k} are the k_{th} adjacent subsequent connector and node of n_i resp. When n_i is a non-parallel gateway (switch case), $\mathbf{T}(n_i)$ sum up the weighted branch times (see Eq. 1c), and the weight α_k indicates the probability of the k_{th} branch being called. When n_i is the end event of the pattern, Eq. 1d will be called as the base case. Based on Eq. 1a-1d, it is clear that the pattern time $\mathcal{T}$ can be obtained through $\mathcal{T} = \mathbf{T}(startEvent)$.

$$\mathbf{T}(n_i) = \begin{cases} \mathcal{T}_{n_i}/\mathcal{R}_{n_i} + \mathcal{T}_{f_{i+1}} + \mathbf{T}(n_{i+1}) & \text{(1a)} \\ \mathcal{T}_{n_i}/\mathcal{R}_{n_i} + \max\limits_{1\le k\le K}(\mathcal{T}_{f_{i+k}} + \mathbf{T}(n_{i+k})) & \text{(1b)} \\ \mathcal{T}_{n_i}/\mathcal{R}_{n_i} + \sum_{k=1}^{K} \alpha_k(\mathcal{T}_{f_{i+k}} + \mathbf{T}(n_{i+k})) & \text{(1c)} \\ \mathcal{T}_{n_i}/\mathcal{R}_{n_i} & \text{(1d)} \end{cases}$$

5.2 Cost

Since that the web services in MSI are mostly deployed and operated separately, the pattern cost should include the activity and gateway waiting cost in addition to the basic cost of service operation. For instance, in the process of E-commerce transactions, even if the consumer does not return the goods, the relevant services are still running on the servers, consuming energy and generating costs.

Similar to the calculation of time, we introduce a recursive function $\mathbf{C}(n_i)$ to calculate the execution cost required from node n_i to the end. $\mathcal{C}^b_{n_i}$ is used to represent the basic cost of node n_i. $\mathcal{C}^w_{n_i}$ and $\mathcal{T}^w_{n_i}$ are the waiting cost per unit of time and the waiting time resp. In sequential cases, i.e., n_i is an event or an activity, $\mathbf{C}(n_i)$ can be calculated through Eq. 2a, where n_{i+1} is the adjacent subsequent node of n_i. When n_i is a gateway (parallel and switch cases), $\mathbf{C}(n_i)$ is the sum of the weighted branch costs (see Eq. 2b), and the weight α_k indicates the likelihood of the k_{th} branch being called. The Eq. 2c is invoked as the base case when n_i is the end event. The pattern cost $\mathcal{C}$ can be estimated recursively from the start event, which means $\mathcal{C} = \mathbf{C}(startEvent)$.

$$\mathbf{C}(n_i) = \begin{cases} \mathcal{C}^b_{n_i} + \mathcal{C}^w_{n_i}\mathcal{T}^w_{n_i} + \mathbf{C}(n_{i+1}) & \text{(2a)} \\ \mathcal{C}^b_{n_i} + \mathcal{C}^w_{n_i}\mathcal{T}^w_{n_i} + \sum_{k=1}^{K} \alpha_k(\mathbf{C}(n_{i+k})) & \text{(2b)} \\ \mathcal{C}^b_{n_i} + \mathcal{C}^w_{n_i}\mathcal{T}^w_{n_i} & \text{(2c)} \end{cases}$$

5.3 Reliability

The reliability is the ratio of services running successfully, which is used to measure the probability of activities in the service process running as required. The reliability can affect the time and cost of the pattern, e.g., if the reliability of an activity was 0.5, its running time would double and the waiting cost of the subsequent nodes would also be affected.

The recursive function $\mathbf{R}(n_{i+1})$ is introduced to calculate the overall reliability of the pattern from node n_i to the end. In a sequential case, the reliability is the product of the reliabilities of all N nodes in the sequence (see Eq. 3a), where $\mathcal{R}_{n_i}$ is the reliability of node n_i. In a parallel case, the reliability is the minimum of the reliabilities of all K parallel substructures (see Eq. 3b). In a switch case, the reliability is the sum of the reliabilities of all K weighted substructures (see Eq. 3c). The Eq. 3d is the base case which will be invoked when n_i is the end event. The pattern reliability $\mathcal{R}$ can be obtained through $\mathcal{R} = \mathbf{R}(startEvent)$.

$$\mathbf{R}(n_i) = \begin{cases} \mathcal{R}_{n_i} * \mathbf{R}(n_{i+1}) & \text{(3a)} \\ \mathcal{R}_{n_i} * \min\limits_{1\le k\le K}(\mathbf{R}(n_{i+k})) & \text{(3b)} \\ \mathcal{R}_{n_i} * \sum_{k=1}^{K} \alpha_k(\mathbf{R}(n_{i+k})) & \text{(3c)} \\ \mathcal{R}_{n_i} & \text{(3d)} \end{cases}$$

5.4 Efficiency

The efficiency of a service pattern depends on the transmission efficiencies of the data, the resources, and the values. The calculation method is shown in Eq. 4. ϵ_o is the size/weight/volume of the data/resource/value object in flow o. $\mathcal{T}_o$ is the corresponding transit time of flow o. $f_{\mathbb{O}}$ represent the base functions which is used to normalize the three types of efficiencies according to their units. $|\mathbb{O}|$ is the number of elements in the set $\mathbb{O}$.

$$\mathcal{E} = \sum_{\mathbb{O}\in\{\mathbb{D},\Theta,\mathbb{V}\}} \sum_{o\in\mathbb{O}} \frac{1}{3|\mathbb{O}|} f_{\mathbb{O}}(\frac{\epsilon_o}{\mathcal{T}_o}) \tag{4}$$

5.5 Value Creation

Value creation refers to the additional value generated by the exchange of value and resources in the service pattern. The additional value arises from the fact that the same resource has different values for different participants. For example, a broken vase may be worthless to a seller, but it may be a priceless antique to a customer. Then the trade of the vase creates additional value to both the seller and the customer. In summary, the sum of the additional value generated by each participant through the exchange of resources and values in a service pattern constitutes the value creation of the pattern.

For a participant, his or her value creation is the difference between the sum of the value and resources he was expected to get and to spend under the pattern (see Eq. 6 and 7). $\mathcal{V}_p$ is the value creation of participant p. $\mathbb{V}^s$, Θ^s, $\mathbb{V}^t$, and Θ^t are the values and resources that are spent or gained. $\alpha_{v,p}$ and $\alpha_{\theta,p}$ are the ratios where value and resource transmission occur. $\Psi_{\theta,p}$ is the value conversion rate of resource θ to participant p. The sum of the value creation of all participants is that of the pattern (see Eq. 5).

$$\mathcal{V} = \sum_{p\in\mathbb{P}} \mathcal{V}_p \tag{5}$$

$$\mathcal{V}_p = SUMV(p,\mathbb{V}^t,\Theta^t) - SUMV(p,\mathbb{V}^s,\Theta^s) \tag{6}$$

$$SUMV(p,\mathbb{V},\Theta) = \sum_{v\in\mathbb{V}} \alpha_{v,p}\epsilon_v + \sum_{\theta\in\Theta} \alpha_{\theta,p}\Psi_{\theta,p}\epsilon_\theta \tag{7}$$

5.6 Normalized Expenditure

In order to be able to evaluate and compare service patterns comprehensively through a unified indicator, we propose the *pattern expenditure*. The pattern expenditure represents the loss per unit benefit per successful run of each node in the pattern. The mentioned loss is the sum of the logarithms of the time and cost consumed by the pattern. As for the mentioned benefit, it represents the product of data/resources/value transmission and value creation.

The calculating formula is shown as Eq. 8. N is the total number of workflow nodes, including activities, gateways, and events. From the proportional relationship between pattern expenditure and each metric involved, i.e., $\mathcal{L} \propto \{\mathcal{T},\mathcal{C},1/\mathcal{R},1/\mathcal{V},1/\mathcal{E}\}$, it can be obtained that the pattern expenditure will be reduced with decreasing pattern time and cost whereas increasing reliability, value creation, and efficiency. The pattern expenditure can be used to optimize service patterns by transforming the multi-objective optimization problem into a single objective optimization problem.

$$\mathcal{L} = \frac{\log(\mathcal{T}+1)+\log(\mathcal{C}+1)}{N*\mathcal{R}*\mathcal{V}*\mathcal{E}} \tag{8}$$

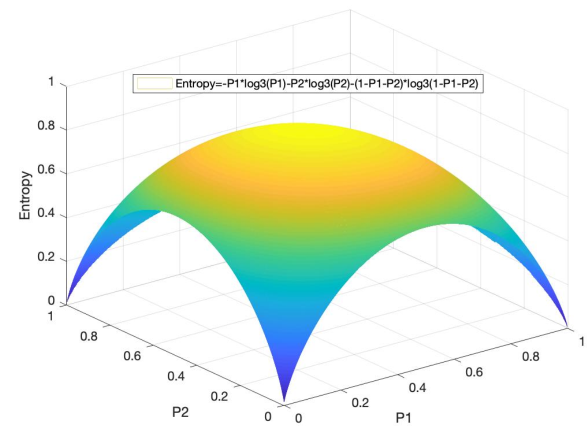


Fig. 3. Illustration of entropy function with 3 variables.

5.7 Pattern Entropy

The pattern entropy is introduced to indicate the chaos degree of service pattern. As shown in Eq. 9, the pattern entropy is the sum of the node entropy, the connector entropy, the data entropy, the resource entropy, and the value entropy.

For a better understanding of the metric, we draw the Shannon entropy function with 3 variables in Fig. 3, which apparently to be a convex function. The extreme value appears when the probabilities of all variables are the same. In other words, the smaller the value of $\mathcal{H}$, the more ordered the service pattern is.

$$\begin{aligned} \mathcal{H} = \;& -\textstyle\sum_{i=1}^{|\mathbb{M}|} P_i \log_{|\mathbb{M}|} P_i - \sum_{j=1}^{|\mathbb{J}|} P_j \log_{|\mathbb{J}|} P_j \\ & -\textstyle\sum_{\mathbb{O}\in\{\mathbb{D},\Theta,\mathbb{V}\}} \sum_{o\in\mathbb{O}} P_o \log_{|\mathbb{O}|} P_o, \\ s.t. \;\;& P_i = \frac{N_i}{|\mathbb{A}\cup\mathbb{G}\cup\mathbb{E}|}, P_j = \frac{N_j}{|\mathbb{F}|}, P_o = \frac{\epsilon_o}{\sum_{o\in\mathbb{O}}\epsilon_o}, \\ & \textstyle\sum_{i=1}^{|\mathbb{M}|} P_i = \sum_{j=1}^{|\mathbb{J}|} P_j = \sum_{o\in\mathbb{O}} P_o = 1 \end{aligned} \tag{9}$$

In Eq. 9, $-\sum_{i=1}^{|\mathbb{M}|} P_i \log_{|\mathbb{M}|} P_i$ represent the value of node entropy. N_i is the number of the workflow nodes run on the i_{th} carrier, $|\mathbb{A}\cup\mathbb{G}\cup\mathbb{E}|$ is the total number of the nodes, and $|\mathbb{M}|$ is the number of carriers involved in the service pattern. P_i represents the distribution probability of workflow nodes run on the i_{th} carrier. The node entropy obtains a smaller value when most of the nodes are running on a few carriers, which also means that most of the services in the pattern run on a uniform platform. For instance, in Fig. 5a, there are 3 types of carriers, 18 nodes in total, and 11, 5, 2 nodes for each type of carrier. Then the node entropy is $-\frac{11}{18}\log_3\frac{11}{18} - \frac{5}{18}\log_3\frac{5}{18} - \frac{2}{18}\log_3\frac{2}{18} \approx 0.82$. In Fig. 5b, all 18 nodels run one same carrier, so the node entropy is 0 since $\log_2 1 = 0$ (the base of the log takes 2 when there is only 1 carrier). Therefore, the service pattern whose nodes are more centrally deployed will obtain a smaller node entropy.

Likewise, $-\sum_{j=1}^{|\mathbb{J}|} P_j \log_{|\mathbb{J}|} P_j$ represent the value of connector entropy. Here, we define that the connectors connecting nodes belonging to the participants with the same types are of the same type. N_j is the number of the flows of the j_{th} type, $|\mathbb{F}|$ is the total flow number, and P_j indicates the probability distribution of j_{th} type of connectors. For

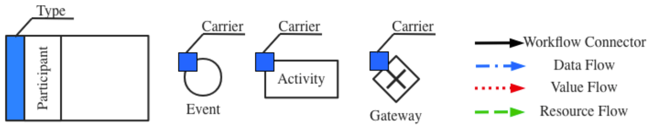


Fig. 4. The notations of SPDL-Q diagram.

instance, in Fig. 5b, there are 18 connectors of 12 types. Whereas in Fig. 5c there are also 18 connectors but in 4 different types. As a result, the connector entropy of Fig. 5c (≈ 0.92) is smaller than that of Fig. 5b (≈ 0.95). Because the seller, financial institution, and logistics company are of the same type (all acted by E-commerce) in Fig. 5c. So, it is clear that the connector entropy will increase if the participants in the service pattern need to cooperate more times and more variably.

As for data/resource/value entropy, they are calculated by the part of $-\sum_{\mathbb{O}\in\{\mathbb{D},\Theta,\mathbb{V}\}}\sum_{o\in\mathbb{O}} P_o \log_{|\mathbb{O}|} P_o$ in Eq. 9. ϵ_o is the size of the object in data/reousrce/value flow o, and P_o indicates the probability distribution of size in flow o. For instance, if there are two data flows of the same size in a service pattern, the data entropy value is $-\frac{1}{2}\log_2\frac{1}{2}*2=1$. If we merge two data objects and implement them through one data flow, then the corresponding pattern entropy is reduced to 0. In other words, to lower the value of those three terms, service patterns need to be designed to deliver larger size objects within fewer times but not the other way around.

In summary, the pattern entropy portrays the chaos degree of a service pattern from five aspects. By hooking the base of the log with the corresponding variation amount, the range of pattern entropy is constrained to [0, 5]. Thus it is possible to compare service patterns with a different number of activities, connectors, data flows, resource flows, and value flows under a unified metric.

6 Data-Driven Case Study

In this section, from the perspective of market regulators, we apply four typical patterns of E-commerce, i.e., middleman pattern, platform pattern, proprietary pattern, and new retail pattern, to SPDL-Q. Below is the mapping relationship between the graphic elements and the SPDL-Q elements (see Fig. 4). Each figure represents a service pattern Γ. Each lane represents a participant who is to execute or attend the activities, gateways, and events within. The participant's name is indicated on the left attached with a color block to present the type. The type is mainly used to represent the organization or superior of the participant and is allowed to refer to another participant. The activities, gateways, and events are represented by squares, diamonds, and circles respectively and connected by real lines. The color block in the top left corner indicates the carrier of the node, i.e., whose server the activity, gateway, or event is running on. In practice, the carriers share the same fields as the participant types. Data flows, value flows, and resource flows are represented by dot dash lines, dot lines, and dash lines, resp.

The graphical representations of the four patterns based on SPDL-Q are shown in Fig. 5. From Fig. 5a to Fig. 5d, they can be considered as the patterns experienced in different stages of E-commerce evolution. It is noticed that the patterns share the same or similar workflows as mentioned above in Fig. 1 if we inspect the workflow perspectives of them. However, the differences among the patterns, which are highlighted by red circles, can still be expressed explicitly through SPDL-Q. At the same time, through the assessment metrics proposed hereinbefore, we can make quantitative assessments and comparative analyses of the four patterns.

Here, we assume to trade about 1 kilogram of goods worth 200 CNY in four patterns. We set the attributes of the pattern elements in the four patterns in Fig. 5 through uniform (U), poisson (P), and exponential (E) distributions (see Tab. 7). For each pattern, 1000 runs were conducted to obtain a mean value and variance of the assessment metrics. The assessment results are shown in Fig. 6.

6.1 Middleman Pattern

The middleman pattern is an E-commerce pattern in which the E-commerce company is only responsible for building an online platform to broker transactions, without providing other additional services such as financial and logistics services (see Fig. 5a). In the middleman pattern, there are four participants. The consumer and the seller are different individuals, and the activities they use are provided by E-commerce companies. The financial institution and logistics company provide their own services to help the consumer and the seller complete the transaction together.

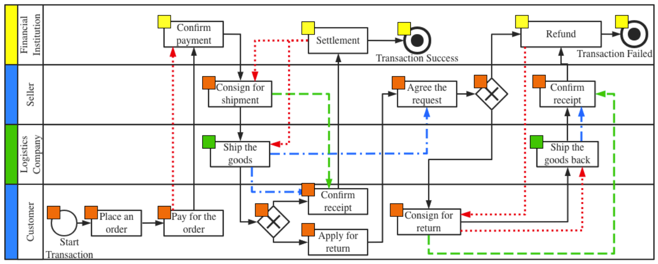
Under the middleman pattern, E-commerce companies are only responsible for building online shopping platforms to satisfy needs of consumers and sellers. The completion of value and resource exchange needs supports from the third-party logistics companies and financial institutions. As a result, users' data, resources, and values need to be exchanged among different platforms and applications. Consequently, although the logistics time is expected to be 3 days, it takes about 4.923 days to complete a transaction under this pattern on average. At the same time, the cost expected of manpower, logistics, and operation can reach 42.78 CNY per deal.

6.2 Platform Pattern

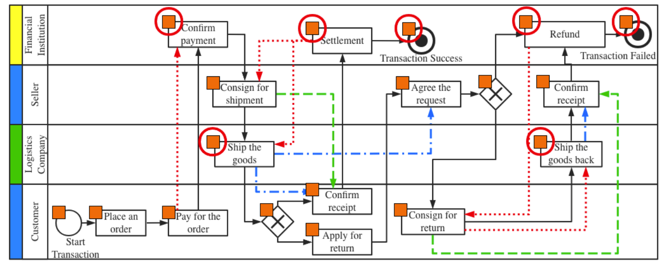
Unlike the middleman pattern, all services in the platform pattern are deployed on the unified platform of E-commerce, although there are still third-party financial institutions and logistics companies involved in this pattern (see Fig. 5b). For example, Taobao, a well-known E-commerce enterprise under Alibaba, connects to third-party logistics companies through Cainiao and traditional banks through Alipay. All services of Cainiao, Alipay and Taobao are running on Aliyun's cloud server. Such a unified deployment can reduce the time and cost of users' data/resource/value transmission between different platforms to some extent.

In terms of parameters, the platform pattern is consistent with the middleman pattern (see Tab. 7). However, due to the unified management of services, users' data does not need to be transferred between servers of multiple platforms or even multiple regions. Therefore, the extra time caused by service interactions can be greatly reduced. As a result,

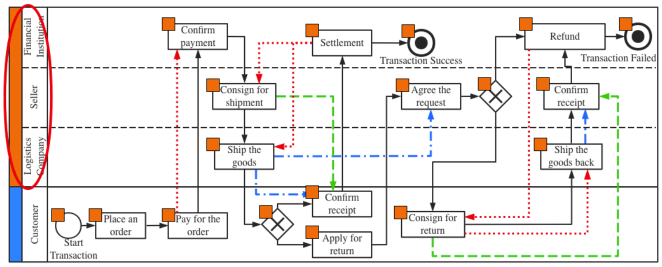
This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing



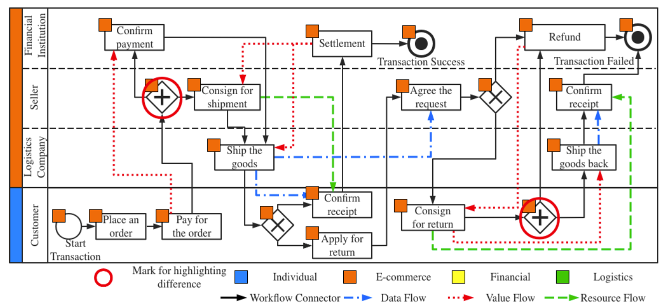
(a) **Middleman Pattern.** E-commerce is only responsible for building a platform to match up customers and sellers. Financial and logistics services are all provided and operated by third parties.



(b) **Platform Pattern.** E-commerce companies provide a unified service platform for sellers, financial institutions, and logistics companies to reside in, so that consumers can enjoy one-stop services.



(c) **Proprietary Pattern.** E-commerce not only provides a trading platform, but also connects sales, finance and logistics through acquisition or personnel injection to further improve the efficiency of collaboration among the three.



(d) **New Retail Pattern.** Further improve the transaction workflow by adding parallel mechanisms through online and offline collaboration.

Fig. 5. Pattern diagrams of four typical E-commerce patterns. The differences between the two adjacent patterns are highlighted by the red circles.

compared with the middleman pattern, the platform pattern reduces the average time by 0.175 days, that is, 4.2 hours. The cost of this pattern is also reduced by about 2.248 CNY per deal. The transfer efficiency of data, resources, and value have been improved accordingly (see Fig. 6). Finally, the average normalized expenditure of the pattern also decreased by about 6.67%. Because of the unification of the carriers, the average pattern entropy also decreased from 4.578 of the middleman pattern to 3.758.

6.3 Proprietary Pattern

In the proprietary pattern, the E-commerce enterprise provides a platform to sell its own goods through its own financial and logistics system (see Fig. 5c). That means the E-commerce enterprise plays the roles of seller, financial institution and logistics company through different departments simultaneously, which can further improve the collaboration efficiency of the supply chain. For example, JD, one of the largest E-commerce platforms in China, sells goods of its own stores, receives payment by its own financial services namely JD wallet, transits goods through its own JD logistics, and realizes a typical proprietary pattern.

In terms of parameters (see Tab. 7), due to the unified control on the source of goods, the seller can deliver the goods from the closest source to the consumer, which achieve a logistics time within 3 days. However, due to the large-scale warehouse stock and management needed by this pattern, the logistics cost increases 25% for each delivery.

As for the assessment results (see Fig. 6), it is obvious that the proprietary pattern greatly shortens the time to complete a transaction, which reaches 2.519 days on average. The transfer efficiency of data and resources have been improved by more than 1.5 times. The efficiency of value transfer has also been slightly improved. This is because of not only the reduction of logistics time but also the improvement of the collaboration efficiency of the supply chain. In addition, although the logistics cost has increased 25%, the overall cost of the proprietary pattern has still decreased from 40.53 CNY in the platform pattern to 31.05 CNY. Finally, through the integration of sellers, logistics, and finance, the normalized expenditure and pattern entropy of the proprietary pattern decreased by about 5.9% and 0.704% compared to the platform pattern, resp.

6.4 New Retail Pattern

The new retail pattern is produced based on the proprietary pattern to help consumers complete intra-city transactions in a short time (see Fig. 5d). It reliases the synchronization of goods delivery and payments through online and offline collaboration. For example, the FreshHemaa typical new retail business of Alibaba, mainly sells daily necessities and fresh food. The offline stores of FreshHema are deployed according to blocks. Each store is only responsible for the orders from nearby streets. If a user placed an order online, FreshHema would transfer the goods from the nearest store.

Due to the characteristics of intra-city transactions, the logistics process under the new retail pattern is usually completed within 30 minutes. Nevertheless the logistics cost has not been greatly reduced because of the frequent small batch distribution. The logistics cost will also increase at

TABLE 7
Assessment Parameters of Four Typical E-commerce Patterns

Parameter	Middleman Pattern	Platform Pattern	Proprietary Pattern	New Retail Pattern
Activity Time (Financial)	U(0.5,1)/second	U(0.5,1)/second	U(0.5,1)/second	U(0.5,1)/second
Activity Time (Logistics)	P(3)/day	P(3)/day	U(1,3)/day	U(10,30)/minute
Activity Time (Seller)	P(60)/second	P(60)/second	P(60)/second	P(60)/second
Activity Time (Customer)	P(120)/second	P(120)/second	P(120)/second	P(60)/second
Gateway Time	U(0.5,1)/second	U(0.5,1)/second	U(0.5,1)/second	U(0.5,1)/second
Flow Time (Basic)	P(50)/millisecond	P(50)/millisecond	P(50)/millisecond	P(50)/millisecond
Flow Time (Crossover)	E(0.5)/second	E(0.5)/second	E(0.5)/second	E(0.5)/second
Flow Time (Cooperation)	E(0.01)/second	E(0.01)/second	E(0.01)/second	E(0.01)/second
Activity Operating Cost (Financial)	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY
Activity Operating Cost (Logistics)	P(6)/CNY	P(6)/CNY	P(8)/CNY	U(4,10)/CNY
Activity Operating Cost (Seller)	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY
Activity Operating Cost (Customer)	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY
Gateway Operating Cost	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY	U(0.5,1)/CNY
Activity/Gateway waiting cost	U(0.5, 2)/CNY per day	U(0.5, 2)/CNY per day	U(0.5, 2)/CNY per day	U(0.5, 2)/CNY per day
Value	P(200)/CNY for goods, P(180)/CNY for seller, P(20) for logistics			
Resource	P(1000)/g			
Data	P(2048)/byte			
Reliability	U(0.995, 0.999)			

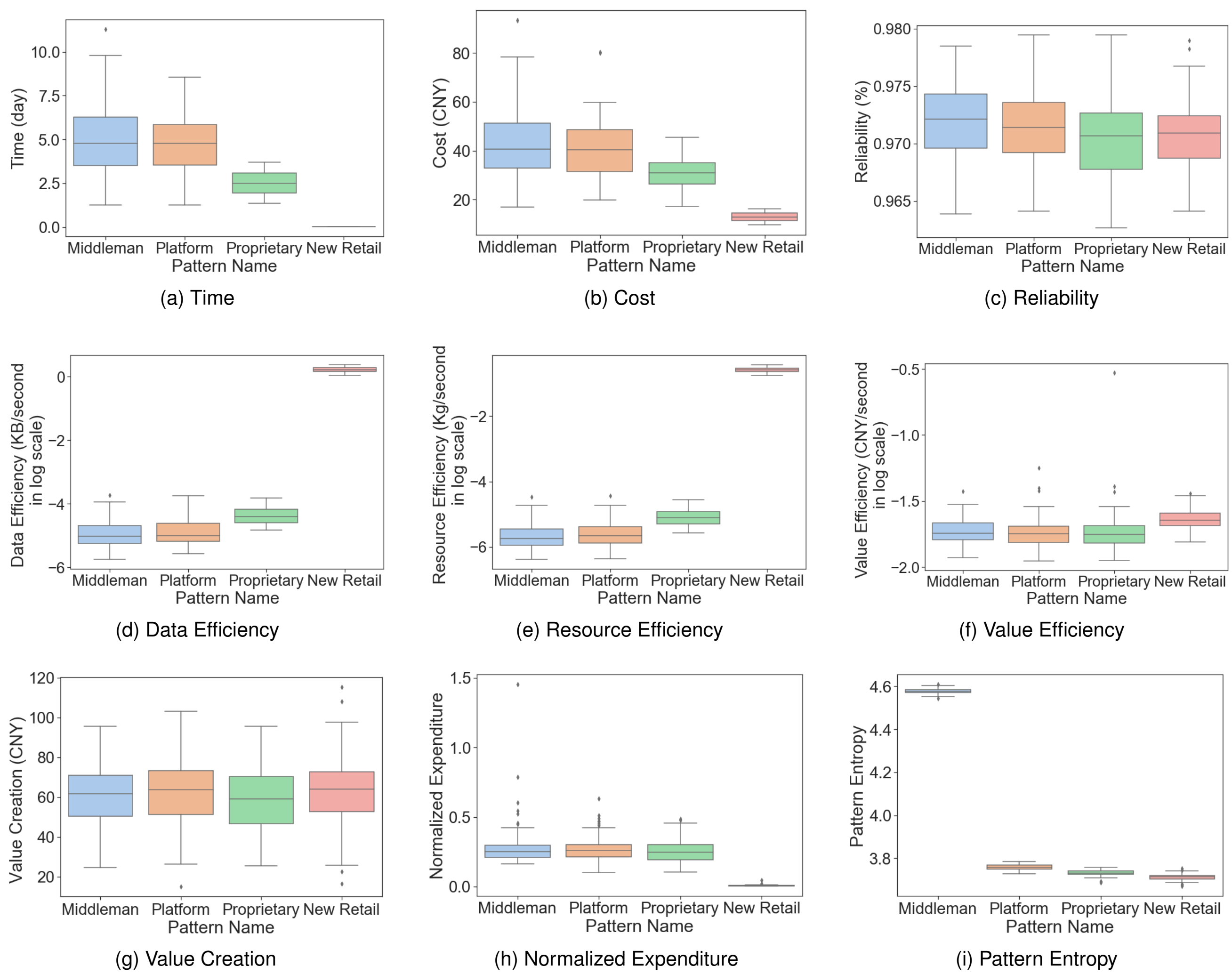


Fig. 6. Assessment results of four typical E-commerce patterns.

night and in bad weather. So the transportation cost of a new retail order is usually 4 to 10 CNY (see Tab. 7).

As a result, the new retail pattern significantly reduced the total time expectation consumed by each order, from 2.5 days in the proprietary pattern to 0.03 days, that is, 43 minutes. The cost per deal has also decreased from 31.05 CNY to 13.05 CNY, about 137.98% lower than the proprietary pattern. Because the logistics time is greatly shortened, the transfer efficiency of data, resources, and value has also been greatly improved. Finally, the normalized expenditure of new retail pattern has reached 0.0078, which is a huge improvement of E-commerce compared to 0.2541 of proprietary pattern. The pattern entropy also decreased slightly due to the refinement of parallel mechanisms of the pattern.

7 Correlation Experiment

In this section, we introduce 11 pattern features, including Depth, Breadth, Service Collaboration Times (SCT), User Collaboration Times (UCT), Execution Expect, Branch Number, Gateway Number, Node Number, Data Number, Resource Number, and Value Number, to try to find out the most influential factors in each metric. Below is an explanation of the pattern features:

- The Depth means the length of the longest path from the start event to the end event.
- The Breadth is the number of the structured continuum of events, tasks, gateways which should be connected by data flow, resource flow, and value flow.
- The Service Collaboration Times (SCT) can be obtained by counting the number of connectors between tasks.
- The User Collaboration Times (UCT) is the connector number that connects nodes of two different participants.
- The Execution Expect means the expectation of execution times of all tasks.
- The Branch Number is the number of structured continuums after removing the critical path (mostly the longest path) in the workflow.
- The Gateway Number is the number of the gateway nodes used in the workflow.
- The Node Number is the number of the events, tasks, gateways used in the workflow.
- The Data/Resource/Value Number is the object number transferred by data/resource/value flows.

In the following experiments, we investigate the correlations between the pattern metrics and features based on a service pattern dataset named S-SPD. S-SPD is a half-real dataset containing 850 service patterns constructed based on an existing dataset of IBM [35]. The IBM dataset constitutes the workflows of the patterns in S-SPD. Data flows, resource flows, value flows, participants, and other missing attributes are generated through unified distribution sampling methods to minimize the impact of the generated data and ensure the effectiveness of the results.

We investigated the correlations through Pearson correlation coefficient (r as abbr.), Spearman coefficient (ρ as abbr.), and Kendall's rank coefficient (τ_b as abbr.). The

TABLE 8
Correlation Coefficients between the Assessment Metrics and the Pattern Features.

Metric	Pattern Feature	r	ρ	τ_b
$\mathcal{T}$	UCT	0.7381**	0.9142**	0.7682**
	ExecutionExpect	0.6324**	0.8847**	0.7151**
	SCT	0.7052**	0.8647**	0.6834**
$\mathcal{C}$	Depth	0.6562**	0.8343**	0.6572**
	ExecutionExpect	0.5075**	0.7929**	0.6152**
	UCT	0.4858**	0.7068**	0.5174**
$\mathcal{R}$	Depth	-0.7116**	-0.8848**	-0.7370**
	ExecutionExpect	-0.5687**	-0.8887**	-0.7308**
	SCT	-0.6117**	-0.8104**	-0.6318**
$\mathcal{E}$	Breadth	0.1196**	-0.0968*	-0.0752*
	BranchNumber	-0.0569	0.0788*	0.0564*
	SCT	-0.0547	0.0675	0.0473
$\mathcal{V}$	ExecutionExpect	0.5650**	0.9363**	0.7828**
	GatewayNumber	0.1625**	0.8219**	0.6425**
	ResourceNumber	0.2072**	0.8180**	0.6573**
$\mathcal{L}$	ExecutionExpect	-0.1689**	-0.8259**	-0.6466**
	SCT	-0.3457**	-0.7980**	-0.6145**
	NodeNumber	-0.3522**	-0.7928**	-0.6090**
$\mathcal{H}$	UCT	0.7525**	0.8556**	0.6734**
	ValueNumber	0.7225**	0.8250**	0.6455**
	NodeNumber	0.7210**	0.8197**	0.6348**

* indicates $<$ .05 statistical significance, ** indicates $<$ .01 statistical significance.

coefficients of the most relevant three features of each metric are shown in Tab. 8. And the corresponding distributions of the features and the metrics are shown in Fig. 7.

As it can be seen, features that are mostly relevant to $\mathcal{T}$ are UCT, Execution Expect, and SCT. Those three features are found to have a strong positive correlation with the time of a service pattern. This is reasonable because the participants collaborate and service interaction could bring in extra cooperation time, and greater Execution Expect means more service execution time.

The correlations between $\mathcal{C}$ and Depth, Execution Expect, UCT are found to be positive under all three coefficients. For Depth, one possible reason for the positive correlation is that the long workflows need to spend more waiting cost compared to short ones. Correlation in Execution Expect is positive because more executions indicate more basic operating cost. The UCT is also positive to $\mathcal{C}$, which may be because that collaboration between participants would bring in more time and then result in more waiting cost.

The features most relevant to $\mathcal{R}$ are Depth, Execution Expect, and SCT. For Depth, the strong negative correlation is reasonable because the longer the workflow is, the weaker the task chain would be since any task exception could result in a pattern fault. The Execution Expect and SCT also has a negative correlation with $\mathcal{R}$, one possible reason is that the greater either feature is, the more tasks will be executed and connected, which means worse robustness since few tasks are perfectly reliable.

For Efficiency $\mathcal{E}$, the Breadth is found to have a weak positive but significant correlation around 0.1196 under Pearson correlation coefficient r. This is possibly because

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

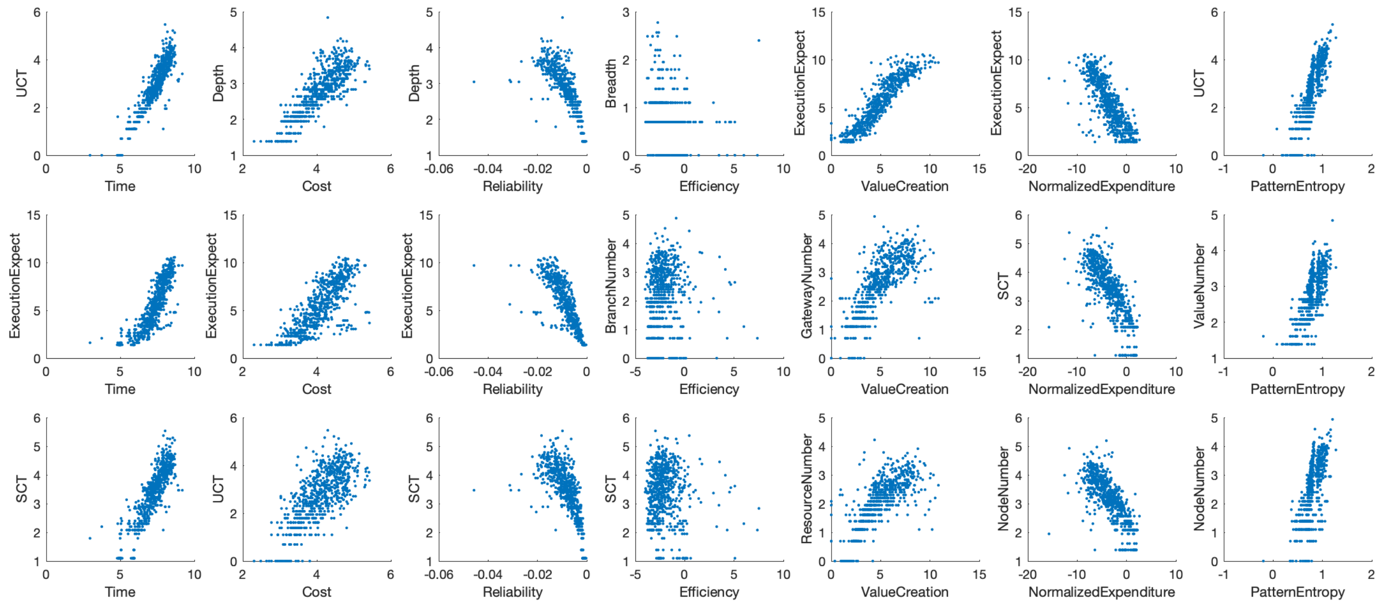


Fig. 7. Examples of the relationship between the Assessment Metrics and the Pattern Features. For better demonstration of the correlation, all the values used in the figures are in log scale.

more workflow in the pattern were performing simultaneously, so that data/resource/value were consumed as soon as it is produced. Besides, the weak positive correlation between Branch Number and $\mathcal{E}$ under ρ and τ_b could be of the same reason.

The correlations between $\mathcal{V}$ and Execution Expect, Gateway Number, and Resource Number are found to be positive under all three coefficients. For Execution Expect, the positive correlation is reasonable because a service pattern could create more value as it executes the corresponding tasks more. One possible reason for the positive correlation in Gateway Number is that the gateway could bring in loop structures, which may result in the create-value tasks to execute more. The Resource Number also has a positive correlation with $\mathcal{V}$. This is reasonable because the exchange of resources is an important source of value creation. In other words, the more resources there are in a service pattern, the more values could be created.

The features most relevant to $\mathcal{L}$ are Execution Expect, SCT, and Node Number. All the three features are found to have a significantly strong negative correlation with Normalized Expenditure $\mathcal{L}$. One possible reason is that the marginal cost would decrease as there are more nodes and connectors involved in a service pattern. As a result, the average expenditure for each node would be less.

For the Pattern Entropy $\mathcal{H}$, the UCT is found to have a significantly strong positive correlation with it under all three coefficients. This is reasonable because the increasing number of participants' collaboration would result in the diversity of connectors and a greater connector entropy. The Node Number is also found to be positive to $\mathcal{H}$. One possible reason is that more nodes not only means greater node entropy but also more connectors to serialize them which could result in a greater connector entropy. As for the Value Number, its positive correlation with $\mathcal{H}$ is reasonable because that more value objects indicate a greater value entropy. Though we omit Data Number and Resource Number, their correlation coefficients with $\mathcal{H}$ are quite close to that of Value Number (r are 0.7164 and 0.7063, ρ are 0.8036 and 0.8176, and τ_b are 0.6144 and 0.6486, resp.). One possible reason is that the more the data or resource object is, the greater the data entropy or resource entropy could be.

8 Comparison & Discussion

In this section, we make a comparison among SPDL-Q, SPDL, BPMN2.0, and artifact-centric BPM (Business Process Model), as shown in Tab. 9, to illustrate the main innovations and advantages of SPDL-Q. The SPDL-Q is of great significance in the following aspects.

1) A particular kind of result could be generated. The BPMN2.0 and artifact-centric BPM can only depict business from the perspective of processes, but cannot portray the resource flow and value flow in the model. Although SPDL complements describes the resource flow, it still lacks a comprehensive portrayal of the value flow. Moreover, none of the three modeling approaches supports quantitative description of the service pattern, and thus cannot provide a comprehensive analysis of the business.

SPDL-Q adds time, cost, reliability, data flow, resource flow, value flow and other attributes and elements to the traditional service process model. The service designer can fit the possible values of relevant attributes through different conditional probability distributions, and then measure the normal performance of the service pattern in time, cost, efficiency, and reliability. In addition, on the basis of these values, a particular kind of result on pattern could be generated.

2) A particular kind of decision could be enabled. The artifact-centric BPM, BPMN2.0, and SPDL are mostly used to instruct development. However, the lack of quantitative description of the business elements makes it impossible to

TABLE 9
Comparison between SPDL-Q and Other Modeling Methods

	Artifact-centric BPM	BPMN2.0	SPDL	SPDL-Q
Activity	tasks between states	atomic or choreography Task	individual services	individual services , expanded carrier, time, cost, reliability
Gateway	structured activities	controller of divergence or convergence	process controller	process controller, expanded carrier, time, cost, reliability
Sequence Flow	state sequence of artifact	logical relationships between nodes	logical relationships between nodes	node order relationship and contain the intermediate time
Data Flow	entity, pairwise data name and attribute	data input, output, and storage	data attributes and states added	quantifiable virtual sources, expanded data type and size
Resource Flow	none	none	available source of wealth	quantifiable physical sources, expanded category and weight
Value Flow	none	none	as a special case of resource flow	cash or virtual currency, expanded currency and volume

locate the bottlenecks and defects of the pattern. The optimization direction of service patterns nowadays is mainly worked out by the brainstorming of the project managers (PM), which usually requires a lot of time and manpower investment, yet the effect cannot be guaranteed. Though there are quantitative analysis methods like QoS (Quality of Service) proposed, they are still not enough to evaluate the resource and value flows of the business or to draw conclusions in the design phase.

The assessment metrics of SPDL-Q consists of time, cost, efficiency, reliability, value creation, normalized expenditure, and pattern entropy. The metrics are typical, consistent, operational, and comprehensive. They can not only reflect the main characteristics and states of the service pattern, but also reflect the strengths and weaknesses. With the help of SPDL-Q, when designing new service patterns, it is clear whether the pattern has achieved the expectations at all levels, where bottleneck is, and what the optimization direction of the pattern should be.

3) A particular kind of impact could be brought about. The quantitative analysis methods based on traditional modeling methods like artifact-centric BPM or BPMN2.0 are designed to be implemented after the service deployment. For instance, the neccessary attributes used in QoS analysis can only be obtained after the business is deployed. Let alone the methods cannot depict resource and value flows. SPDL, while allowing for a more comprehensive portrayal of the service pattern, still faces the same dilemma. This leads to insufficient analysis and slow iteration of service patterns.

Through SPDL-Q, MSI companies can simulate the service pattern through conditional probability distribution and historical data. They can judge whether the new function will have a positive impact on the system beforehand. Besides, a comprehensive quantitative analysis of the change could be made at the new pattern's design stage.

9 CONCLUSION

As the MSI business is developing quickly and diversely, service pattern modeling and assessment are becoming increasingly important. In this work, to empower the MSI with pattern computing and quantify the assessment and comparison of the service patterns, a service pattern assessment framework is proposed along with a pattern description language called SPDL-Q and a set of pattern assessment metrics. Case studies on four E-commerce patterns are conducted which validate the effectiveness of our framework and reveal the profound reason as to why and how the later patterns outperform the former ones. Besides, the most influential factors to each pattern metric are inspected through a correlation experiment to enlighten the relevant practitioners. In addition, a discussion is carried out to compare the SPDL-Q with existing modeling methods and illustrate the advantages.

In the future, service pattern selection and composition methods could be proposed on the basis of SPDL-Q. Also, with the assessment metrics as goals, the optimization algorithms can be designed as well. All those works could be profound to improve the efficiency of society and strengthen the innovation of the MSI.

REFERENCES

[1] J. Yin, B. Zheng, S. Deng, Y. Wen, M. Xi, Z. Luo, and Y. Li, "Crossover service: Deep convergence for pattern, ecosystem, environment, quality and value," in *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE, 2018, pp. 1250–1257.

[2] W. M. Van Der Aalst, A. H. Ter Hofstede, and M. Weske, "Business process management: A survey," in *International conference on business process management*. Springer, 2003, pp. 1–12.

[3] C. Barreto, V. Bullard, T. Erl, J. Evdemon, D. Jordan, K. Kand, D. König, S. Moser, R. Stout, R. Ten-Hove *et al.*, "Web services business process execution language version 2.0 primer," *OASIS Web Services Business Process Execution Language (WSBPEL) TC, OASIS Open*, 2007.

[4] K. Bhattacharya, C. Gerede, R. Hull, R. Liu, and J. Su, "Towards formal analysis of artifact-centric business process models," in *International Conference on Business Process Management*. Springer, 2007, pp. 288–304.

[5] M. Autili, P. Inverardi, and M. Tivoli, "Choreography realizability enforcement through the automatic synthesis of distributed coordination delegates," *Science of Computer Programming*, vol. 160, pp. 3–29, 2018.

[6] F. Corradini, A. Morichetta, A. Polini, B. Re, and F. Tiezzi, "Collaboration vs. choreography conformance in bpmn," *arXiv preprint arXiv:2002.04396*, 2020.

[7] ——, "Collaboration vs. choreography conformance in bpmn 2.0: from theory to practice," in *2018 IEEE 22nd International Enterprise Distributed Object Computing Conference (EDOC)*. IEEE, 2018, pp. 95–104.

[8] J. Yin, Z. Luo, Y. Li, and Z. Wu, "Service pattern: An integrated business process model for modern service industry," *IEEE Transactions on Services Computing*, vol. 10, no. 6, pp. 841–853, 2016.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication. Citation information: DOI 10.1109/TSC.2021.3091201, IEEE Transactions on Services Computing

[9] P. Y. Wong and J. Gibbons, "A process semantics for bpmn," in *International Conference on Formal Engineering Methods*. Springer, 2008, pp. 355–374.
[10] N. Russell and W. M. van der Aalst, "Evaluation of the bpel4people and ws-humantask extensions to ws-bpel 2.0 using the workflow resource patterns," *Bpm center report, Department of Technology Management, Eindhoven University of Technology GPO Box*, vol. 513, p. 142, 2007.
[11] A. Rodríguez, A. Caro, C. Cappiello, and I. Caballero, "A bpmn extension for including data quality requirements in business process modeling," in *International Workshop on Business Process Modeling Notation*. Springer, 2012, pp. 116–125.
[12] M. Ramos-Merino, L. M. Álvarez-Sabucedo, J. M. Santos-Gago, and F. de Arriba-Pérez, "A pattern based method for simplifying a bpmn process model," *Applied Sciences*, vol. 9, no. 11, p. 2322, 2019.
[13] E. Bazhenova, F. Zerbato, B. Oliboni, and M. Weske, "From bpmn process models to dmn decision models," *Information Systems*, vol. 83, pp. 69–88, 2019.
[14] M. Estanol, A. Queralt, M. R. Sancho, and E. Teniente, "Artifact-centric business process models in uml," in *International Conference on Business Process Management*. Springer, 2012, pp. 292–303.
[15] M. Estañol, M.-R. Sancho, and E. Teniente, "Ensuring the semantic correctness of a bauml artifact-centric bpm," *Information and software technology*, vol. 93, pp. 147–162, 2018.
[16] G. Kang, L. Yang, and L. Zhang, "Toward configurable modeling for artifact-centric business processes," *Concurrency and Computation: Practice and Experience*, vol. 32, no. 2, p. e5367, 2020.
[17] C. Zott, R. Amit, L. Massa *et al.*, "The business model: Theoretical roots, recent developments, and future research," *IESE Research Papers*, vol. 3, no. 4, pp. 1–43, 2010.
[18] C. Zott, R. Amit, and L. Massa, "The business model: recent developments and future research," *Journal of management*, vol. 37, no. 4, pp. 1019–1042, 2011.
[19] R. Amit and C. Zott, "Value creation in e-business," *Strategic management journal*, vol. 22, no. 6-7, pp. 493–520, 2001.
[20] A. Leshob, "Towards a business-pattern approach for uml models derivation from business process models," in *2016 IEEE 13th International Conference on e-Business Engineering (ICEBE)*. IEEE, 2016, pp. 244–249.
[21] M. Madlberger and S. Matook, "Creation of utilitarian value with online and offline transaction phases." in *CONF-IRM*, 2012, p. 13.
[22] S. Liu, "Business characteristics and business model classification in urban agriculture," Ph.D. dissertation, Master thesis for the chair group Rural Sociology submitted in fulfillment , 2015.
[23] S. Reinhold, P. Beritelli, and R. Grünig, "A business model typology for destination management organizations," *Tourism Review*, 2019.
[24] M. Donner, R. Gohier, and H. de Vries, "A new circular business model typology for creating value from agro-waste," *Science of the Total Environment*, vol. 716, p. 137065, 2020.
[25] R. Xia, M. Rost, and L. E. Holmquist, "Business models in the mobile ecosystem," in *2010 Ninth International Conference on Mobile Business and 2010 Ninth Global Mobility Roundtable (ICMB-GMR)*. IEEE, 2010, pp. 1–8.
[26] E. Balan, "Integrating community values in the design of a mobile application for parkour practitioners," 2013.
[27] P. Beynon-Davies, P. Jones, and G. White, "Business patterns and strategic change," *Strategic Change*, vol. 25, pp. 675–691, 11 2016.
[28] Q. A. Liang, J.-Y. Chung, S. Miller, and Y. Ouyang, "Service pattern discovery of web service mining in web service registry-repository," in *2006 IEEE International Conference on e-Business Engineering (ICEBE'06)*. IEEE, 2006, pp. 286–293.
[29] Y. Li, Z. Luo, J. Yin, L. Xu, Y. Yin, and Z. Wu, "Enterprise pattern: integrating the business process into a unified enterprise model of modern service company," *Enterprise Information Systems*, vol. 11, no. 1, pp. 37–57, 2017.
[30] C.-H. Lee, S.-Y. Hwang, and I.-L. Yen, "A service pattern model for flexible service composition," in *2012 IEEE 19th International Conference on Web Services*. IEEE, 2012, pp. 626–627.
[31] Y. Duan, A. Kattepur, H. Zhou, Y. Chang, M. Huang, and W. Du, "Service value broker patterns: Towards the foundation," in *2013 IEEE/ACIS 12th International Conference on Computer and Information Science (ICIS)*. IEEE, 2013, pp. 149–154.
[32] J. Yin, Z. Luo, Y. Li, B. Fan, W. Liu, and Z. Wu, "Towards a service pattern model supporting quantitative economic analysis," in *2014 IEEE World Congress on Services*. IEEE, 2014, pp. 95–102.
[33] Z. Luo, Y. Li, J. Yin, H. Gao, and Y. Yin, "Service pattern evaluation: Studying profitability from perspective of resource," in *2017 IEEE International Conference on Cognitive Computing (ICCC)*. IEEE, 2017, pp. 88–95.
[34] I. Pelle, J. Czentye, J. Dóka, and B. Sonkoly, "Towards latency sensitive cloud native applications: A performance study on aws," in *2019 IEEE 12th International Conference on Cloud Computing (CLOUD)*. IEEE, 2019, pp. 272–280.
[35] D. Fahland, C. Favre, B. Jobstmann, J. Koehler, N. Lohmann, H. Völzer, and K. Wolf, "Instantaneous soundness checking of industrial business process models," in *International Conference on Business Process Management*. Springer, 2009, pp. 278–293.

Meng Xi received his B.S. degree in computer science from Zhejiang University, China in 2017. He is now working towards a Ph.D. degree at the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. He has been a recipient of the Best Paper Award of IEEE SMDS 2020. His current research interests include Service Computing, Data Science, and Machine Learning.

Jianwei Yin received the Ph.D. degree in computer science from Zhejiang University (ZJU) in 2001. He was a Visiting Scholar with the Georgia Institute of Technology. He is currently a Full Professor with the College of Computer Science, ZJU. Up to now, he has published more than 100 papers in top international journals and conferences. His current research interests include service computing and business process management. He is an Associate Editor of the IEEE Transactions on Services Computing.

Jintao Chen is working towards a Ph.D. degree at the College of Computer Science and Technology, Zhejiang University, Hangzhou, China. She received her B.S. degree in Internet of Things from Central South University. Her research interests include service computing and information networks.

Ying Li received the B.S., M.S. and Ph.D. degrees in computer science from Zhejiang University, China, in 1994, 1997 and 2000, respectively. He is currently an associate professor with the College of Computer Science, Zhejiang University. His research interests include service computing, business process management and compiler.

Shuiguang Deng is currently a full professor at the College of Computer Science and Technology in Zhejiang University, China, where he received a BS and Ph.D. degree both in Computer Science in 2002 and 2007, respectively. He previously worked at the Massachusetts Institute of Technology in 2014 and Stanford University in 2015 as a visiting scholar. His research interests include Edge Computing, Service Computing, Cloud Computing, and Business Process Management. He serves for the journal IEEE Trans. on Services Computing, Knowledge and Information Systems, Computing, and IET Cyber-Physical Systems: Theory & Applications as an Associate Editor. Up to now, he has published more than 100 papers in journals and refereed conferences. In 2018, he was granted the Rising Star Award by IEEE TCSVC. He is a fellow of IET and a senior member of IEEE.