Towards a Service Pattern Model Supporting Quantitative Economic Analysis

Jianwei Yin
Computer Science
Zhejiang University
Hangzhou, China
Email: zjuyjw@cs.zju.edu.cn

Zhiling Luo
Computer Science
Zhejiang University
Hangzhou, China
Email: luozhiling@zju.edu.cn

Ying Li
Computer Science
Zhejiang University
Hangzhou, China
Email: cnliying@zju.edu.cn

Binbin Fan
Computer Science
Zhejiang University
Hangzhou, China
Email: fanbinbin@zju.edu.cn

Weiwei Liu
Computer Science
Zhejiang University
Hangzhou, China
Email: lww319@zju.edu.cn

Zhaohui Wu
Computer Science
Zhejiang University
Hangzhou, China
Email: wzh@cs.zju.edu.cn

***Abstract*—The research on the business models is a hot topic in recent years. It is an interesting problem to study the business model of the service. There are three kinds of models related: classical service models, Business process (BP) models and the enterprise business (EB) models in management. However, none of them covers all the properties of the service business model. In this paper, we define the business model of the service as the combination of four kinds of strategies and name it as the service pattern. We also propose a language named Service Pattern Description Language (SPDL) covering all the elements involved in these strategies. We formulate the language syntax and two basic extraction rules assisting economic analysis. Furthermore, we extend Business Process Model Notation (BPMN) to support SPDL, which is named BPMN for Service Pattern (BPMN4SP). The example of Mobile Application Platform is studied in detail for a better understanding of SPDL.**

***Index Terms*—service pattern, service model, business model, business process, value creation and economic analysis;**

I. Introduction

Nowadays the competition between businesses is not the competition of products but the competition of business models, just as Peter F. Drucker, the master in management, had said.

Taobao is a typical modern service company in China, which is established in 2003. The online mall (http://www.taobao.com) is the primary business of *Taobao*. *Taobao* does not sell any goods, but it provides a platform service to sellers and attracts buyers. This business model is called C2C (Customer to Customer) in e-business. *Taobao* does not charge sellers for the basic platform service. The advertisement and the value-added services (e.g. Web site decorating service) are the major sources of income. Since the trading volume arrives 17.2 billion yuan (nearly 2.7 billion dollars) in a single day, *Taobao* now is the biggest online mall in China. The trading data (logs) is now valuable helping to analyze the customer behaviors. The magic of this business model attracts lots of researchers studying the business model in recent years. There are three basic questions on the business model:

- Q1: What is the business model of the service (e.g. *Taobao*)?
- Q2: How to analysis the business model of a special service?
- Q3: How to modify and re-design the business model of the service for a better benefit?

To address Q1, we define the business model of the service as the combination of four kinds of strategies: *Resources Allocating (RA)*, *Activities Organizing (AO)*, *Shareholders Coordinating (SC)* and *Productions Designing (PD)*. We name the business model of the service as the **Service Pattern**.

There are three kinds of models related to address Q2.

- *Classical service model* concentrates on the service function and quality. And it does not cover the business strategy and business process.
- *Business process (BP) model* defines the business process. Both BPEL (Business Process Execution Language) [1] and BPMN (Business Process Model and Notation) [2] [3] solve the problem of business modeling and process optimization, achieve a great success in the last few years. However, the economic elements (e.g. Resources) are not defined clearly in BP.
- *Enterprise Business (EB) model* in management is a hot topic. EB do good in assisting analyzing the business strategy while poor in process analysis. And EB cannot figure out the relation between the business processes with the business strategy.

To address Q2, we extract four basic elements from these four strategies: *resource* from RA, *activity* from AO, *role* from SC and *entity* from PD. In this paper, we propose a language named Service Pattern Description Language (SPDL) covering these elements. Furthermore, we propose two basic analyzing tools based on SPDL. The SPDL is a high-level language

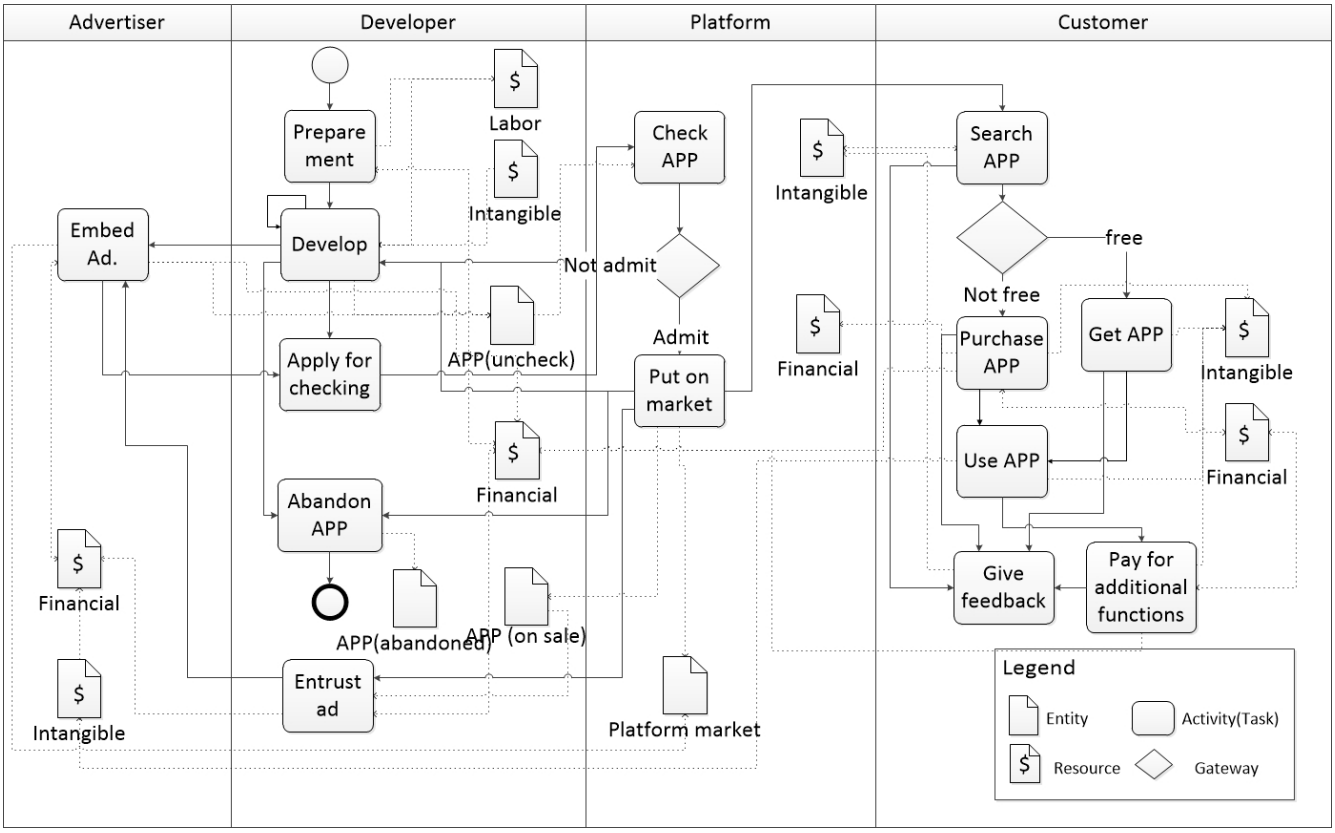

Fig. 1. The Swim Lane Diagrams of Mobile Application Platform Service example

which can bridge the gap between BP and EB. For better implementation, BPMN4SP is introduced which extends the basic BPMN with the elements of SPDL. We replace the definition of resource and activity in BPMN. We have studied the basic idea of Artifact-Centric Business Process Model (Artifact BP) [4] and introduced the entity life cycle into our model.

To solve Q3, we need to address Q1 and Q2 at first. Both Q1 and Q2 are discussed in this paper while Q3 is left for the future work.

The paper makes the following technical contributions:

- The concept of service pattern is defined as the business model of the service and define service pattern by 4 kinds of strategies (RA, AO, SC and PD).
- A service pattern modeling language SPDL and two basic economic analyzing rules are proposed in this paper. SPDL covers four kinds of basic elements (Resource, Activity, Role and Entity) in strategies (RA, AO, SC and PD). Furthermore, SPDL bridges the gap between the business process and the enterprise business model.
- BPMN4SP is introduced as an extension of BPMN and Artifact BP. BPMN4SP implements SPDL and extends the usage scope of BPMN.

The rest of this paper is organized as follows. Section II presents a motivating example of our research. The Service Pattern Description Language (SPDL) is formulated in section III. Two basic analyzing rules are proposed in Section IV. Section V introduces the BPMN4SP and the detail comparison with BPMN. Section VI discusses the complete service example of Mobile Application Platform. Section VII reviews the related works. Finally, section VIII concludes this paper and provides an outlook on possible continuations of our work.

II. A Motivating Example

Considering a *Mobile Application Platform Service* (MAP, for short). The *Mobile Application* (*APP*, for short) is a kind of executable program deployed on *Smart Mobile Phones* (such iPhone and Android Phone). An application can (in most cases) only be deployed on one kind of Mobile Phone Operating System (e.g. IOS, WP or Android). Applications are developed by **Developers**. The **Developer** is a *role* in our discussion.

The concept of role is quite different from participator. A programmer or software engineer is a participator instead of a role. More discussions can be found in the next section. In order to distinguish the role with a specific participator, we use the bold style to identify the name of the role.

Along with the **Developer**, another important role is the **Customer** who purchases for APPs. However, in most cases, the **Developer** does not sell its APPs to the **Customer** directly. The **Developer** puts its APPs on a market, which is the third

important role: the **Platform**. The **Platform** collects huge amounts of APPs, presents them in sequence of their rankings, and provides the purchasing channel for the **Customer**. Once a **Customer** has bought an APP from the **Platform**, the fee will be allocated to **Platform** and **Developer** by some ratio. A famous example of **Platform** is the *APP Store* of Apple Inc. There are over 40 billion downloads from *APP Store* and over 500 million active users.

The **Advertiser** is another interesting role in this service. The **Advertiser** puts some *advertisements* in the APP by purchasing to the **Developer**. Sometimes, the software company places the advertisements about its own APPs. In this situation, the software company plays two roles: the **Developer** and the **Advertiser**. Fig. 1 presents the business process of MAP service by the notations of BPMN. We use the mark of resources in this figure, which is defined in BPMN4SP and not in BPMN. If we use BPMN to model this process, the **Advertiser** would have only one activity and we could not get more information about how the **Advertiser** earns money from this activity. However, once we declared the resources, we can find out that, **Advertiser** earns by investing money on the presentation of advertisements and promoting the benefit of the advertisement. Therefore, we need a new model with resources and roles defined clearly to support quantitative service pattern analysis.

III. Service Pattern Description Language

This section formulates the key syntaxes and basic notions of SPDL. Fig. 2 presents the relation of basic notions by UML diagram. There are 4 basic elements (Resource, Activity, Role and Entity) in service pattern strategies (RA, AO, SC and PD). A role class owns many resources and an entity class has many attributes and states. The concept of activity is redefined by adding roles and attributes. The step denotes the execution order, from one activity to another.

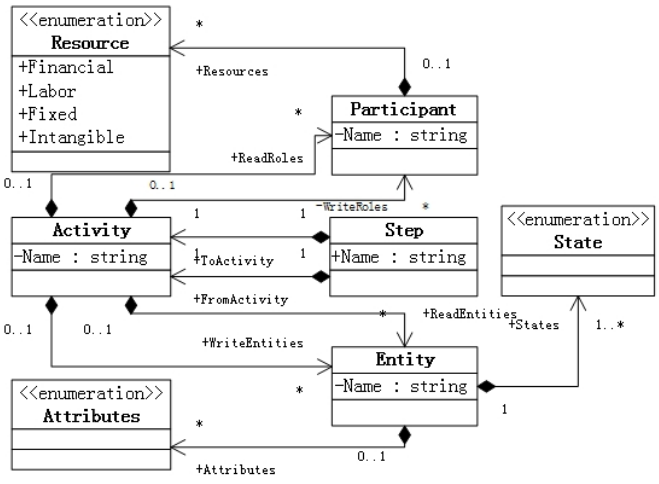

Fig. 2. The UML diagram of basic notions

A. Basic Set

We assume the existence of the following pairwise disjoint countable infinite sets: $\mathcal{T}_P$ of *primitive types*, $\mathcal{C}_\mathcal{E}$ of entity classes (names), $\mathcal{AT}$ of attributes (names), $\mathcal{S}$ of entity states, $\mathbf{ID}_{C_E}$ of (entity) identifiers for each class $C_E \in \mathcal{C}_\mathcal{E}$, $\mathcal{C}_\mathcal{R}$ of role classes (names), $\mathcal{RE}$ of resources (names), $\mathcal{A}$ of activities (names), $\mathcal{BR}$ of business rules. A $type$ is an element in the union $\mathcal{T} = \mathcal{T}_P \cup \mathcal{C}_\mathcal{E}$.

The *domain* of each type t in $\mathcal{T}$, denoted as $\mathbf{DOM}(t)$, is defined as follows:

- if $t \in \mathcal{T}_p$ is a primitive type, the domain $\mathbf{DOM}(t)$ is some known set of *values* (integers, strings, etc.);
- if $t \in \mathcal{C}_\mathcal{E}$ is an entity type, $\mathbf{DOM}(t) = \mathbf{ID}_t$.

B. Entity

Definition 3-1. An *entity class* is a tuple $(C_E, \mathbf{AT}, \tau, Q, s, F)$ where $C_E \in \mathcal{C}_\mathcal{E}$ is an entity class name, $\mathbf{AT} \subseteq \mathcal{AT}$ is a finite set of attributes, $\tau : \mathbf{AT} \to \mathcal{T}$ is a total mapping, $Q \subseteq \mathcal{S}$ is a finite set of states, and $s \in Q$, $F \subseteq Q$ are *initial, final* states (resp.).

Definition 3-2. An *entity instance* of entity class $(C_E, \mathbf{AT}, \tau, Q, s, F)$ is a triple (e, μ, q) where $e \in \mathbf{ID}_{C_E}$ is an identifier, μ is a partial mapping that assigns each attribute $AT \in \mathbf{AT}$ an element in its domain $\mathbf{DOM}(\tau(AT))$, and $q \in Q$ is the current state.

The concept of entity covers the business data objects. We may denote an entity class $(C_E, \mathbf{AT}, \tau, Q, s, F)$ simply as C_E. A class C_{E2} is *referenced by* another class C_{E1} if an attribute of C_{E1} has type C_{E2}. Similarly, an identifier e_2 is *referenced in* an entity instance e_1 if e_2 occurs as an attribute value of e_1.

Example 3-1. In MAP, APP is an important entity. The APP class and APP instance are presented in Table I. The APP instance we used here, is *Angry Birds*. Its state is on sale, meaning that it is available on *APP store*. Its price is 0, meaning that this version is free. Its ID is id391231.

TABLE I
Entity Example: APP Entity and an APP instance

APP Entity		An APP instance
ATTRIBUTES:	STATES:	ID:id391231
name:string	Initialized	STATE: On sale
quantity:int	Unchecked	ATTRIBUTES:
price:float	Not on sale	name: Angry Birds
version:int	On sale	quantity: 5
sales:int	...	price: 0
...		version: 12
		...

C. Resource

Before defining the role, we define the resource at first. The resources are those valuable in the process. There are four basic kinds of resources.

- The *financial* resource is the money in each form.
- The *fixed* resource includes the houses, office equipment, etc.

TABLE II
ROLE EXAMPLE: DEVELOPER AND DEVELOPER INSTANCE

Developer	Developer instance
RESOURCES:	ID:id191231
Labor:float	RESOURCE:
Financial:float	Labor: 20.3
Fixed:float	Financial: 32000
Invisible:float	Fixed: 1230000
	Invisible: 31233

- The *labor* resource is the resources of available manpower.
- The *intangible* resource includes the brand and information resources.

We define the type of each kind resource as the basic float type for simplification.

D. Role

Definition 3-3. A *role class* is a triple $(C_R, \mathbf{RE}, \tau)$ where $C_R \in \mathcal{C}_\mathcal{R}$ is the role class name, $RE \subseteq \mathcal{RE}$ is a finite set of resources, $\tau : RE \to \mathcal{T}$ is a total mapping.

Definition 3-4. The *role instance* of role class $(C_R, \mathbf{RE}, \tau)$ is a two-tuple (r, μ) where $r \in \mathbf{ID}_{C_R}$ is an identifier, and μ is partial mapping that assigns each resource $RE \in \mathbf{RE}$ an element in its domain $\mathbf{DOM}(\tau(RE))$.

In SPDL, The key characteristic of the role is owing resources (financial, fixed, labor and intangible resources). And it is quite different from the concept of participator in BP. An independent software company is a role but a software engineer is only a participator.

Example 3-2. In MAP, there are 4 roles: **Customer**, **Developer**, **Platform** and **Advertiser**. Table II presents a develop role and an instance, whose ID is id191231.

E. Schema

Definition 3-5. An *entity schema* is a finite set Γ_E of entity classes with distinct names such that every class referenced in Γ_E also occurs in Γ_E. The *role schema* followed Γ_E is a finite set Γ_R of role classes with distinct names such that every class referenced in Γ_E. The *schema* is a finite set $\Gamma = \Gamma_E \cup \Gamma_R$.

F. Atom

Definition 3-6. An *atom* over a schema Γ is one of the following:

1) boolean expression,
2) $t_1 = t_2$, where t_1,t_2 are instances of entity class (or role class) C in Γ,
3) $\mathbf{DEFINED}(t, D)$, where t is a instance of entity class C and D an attribute in C or t is a instance of role class C and D a resource in C ,
4) $\mathbf{NEW}(t, D)$, where t is a instance of entity class C and D an entity typed attribute in C or t is a instance of role class C and D an entity typed resource in C or,
5) $s(t)$ (a (state) atom), where t is a instance of entity class C and s a state of C,
6) $\neg c$, where c is an atom, and
7) $c_1 \wedge c_2$ and $c_1 \vee c_2$, where $c1$, $c2$ are atoms.

A condition is *stateless* if it contains no state atoms.

Example 3-3. An example of a condition is as follows:

$$\mathbf{DEFIND}(\text{id391231, APP.price}) \wedge \text{on sale}(id391231)$$

The condition is the combination of two atoms. The price of id391231 has been defined. id391231 is in the state of on sale.

G. Activity

Definition 3-7. A *activity* over schema Γ is tuple $(n, V_{Er}, V_{Ew}, V_{Rr}, V_{Rw}, M, P, E)$, where $n \in \mathcal{A}$ is an activity name, V_{Er}, V_{Ew} finite sets of variables of entity classes in Γ, V_{Rr}, V_{Rw} finite sets of variables of role classes in Γ, P a condition over V that does not contain $\mathbf{NEW}$, M a partial mapping from V_{Er} to V_{Ew}, and E a conditional effect.

M describes the mapping that which input attributes influenced each output. Considering a sequence of input attributes $xe_1, ..., xe_k$ and output attributes $ye_1, ..., ye_l$ $(k, l \geq 1)$. $\mathcal{M} \in \mathcal{R}^{k \times l}$ is a matrix.

$$\mathcal{M}_{i,j} = \begin{cases} 1 & \text{if } xe_i \text{ to } ye_j \text{ is a mapping in } M \\ 0 & \text{elsewise} \end{cases}$$

We denote $\mathcal{M}(i, j) = \mathcal{M}_{i,j}$.

Example 3-4. In MAP, an example of purchasing APP is presented in Table III. An entity instance (id391231) is used and two roles, namely id231441, id231357 have participated in this activity. This activity reads the price and sales of APP as input attributes. It reads the financial resource of customer and platform (resp.) as input resources. The Mapping relation is that the sales of APP influence itself. The precondition consists of 3 parts:

1) this APP has declared its price;
2) the price of APP should be smaller than the mount of customer's money;
3) and the APP is in the state of on sale.

There are 3 effects of this activity:

1) the sales of APP increases 1,
2) the money of platform increases,
3) the money of customer decreases.

TABLE III
ACTIVITY EXAMPLE

Purchasing APP	
Entity:	id391231:APP
Role:	id231441:Customer, id231357:Platform
Read Attributes:	id391231.Price,id391231.sales
Write Attributes:	id391231.sales
Read Resource:	id231441.Financial, id231357.Financial
Write Resources:	id231441.Financial, id231357.Financial
Mapping Relation:	id391231.Sales→id391231.Sales
Pre-Condition:	$\mathbf{DEFINED}(id391231, Price) \wedge$ $id391231.Price \leq id231441.Financial \wedge$ on sale($id391231$)
Effect:	$id391231.Sales = id391231.Sales + 1 \wedge$ $id231357.Financial = id231357.Financial + id391231.Price \wedge$ $id231441.Financial = id231441.Financial - id391231.Price$

H. Rule

Definition 3-8. Given a schema Γ and a set of activities $\mathcal{A}$. a *business rule* is an expression with one of the following two forms:

- "**If** φ **invoke** $\sigma(xe_1, ..., xe_k; ye_1, ..., ye_l; xr_1, ..., xr_m; yr_1, ..., yr_n)$", or
- "**If** φ **change state to** ϕ".

where φ is a condition over variables $xe_1, ..., xe_k; ye_1, ..., ye_l; xr_1, ..., xr_m; yr_1, ..., yr_n (k, l, m, n \geq 1)$, σ an activity in $\mathcal{A}$ such that $xe_1, ..., xe_k$ are all entity variables to be read, $ye_1, ..., ye_l$ are all entity variables to be written, $xr_1, ..., xr_m$ are all role variables to be read, $yr_1, ..., yr_n$ are all role variables to be written, and ϕ a condition consisting of only positive state atoms over $ye_1, ..., ye_l$.

IV. Analyzing Tools

Here are two basic analyzing tools: the information flow and value flow. The information flow concentrates on the data influence relation in the process which identifies how one attribute influences others. The value flow is the well-known approach in business administration to organize the work that a role conducts to achieve its business goals. Both information flow and value flow can be extracted from the basic process defined by SPDL. The information flow determines the transferring of important data while the value flow identifies the important activities inverting values. The quantity of each resource can be calculated by the data from the information flow.

A. Information Flow

Definition 4-1. An attribute AT_1 is *referenced by* another attribute AT_2, (denote $AT_1 \lhd AT_2$) if and only if there is at least an activity A which reads AT_1 as the input value, modifies AT_2 and $\mathcal{M}(AT_1, AT_2) = 1$.

Definition 4-2. Considering an attribute set $\mathcal{AT}$, we assume that the set $\mathcal{AT}^{+1}$ represents the set of attributes which is referenced by each attribute in $\mathcal{AT}$. $\mathcal{AT}^{+1}$ is the *positive first order attributes* of $\mathcal{AT}$.

First Order Attributes Decision Rule. $\forall AT \in \mathcal{AT}^{+1}, \exists AT' \in \mathcal{AT}$ such that $AT \lhd AT'$.

Cascading Decision Rule. $\forall \mathcal{AT}_1 = \mathcal{AT}_2^{+1}, \mathcal{AT}_2 = \mathcal{AT}_1^{-1}$.

We denote the *positive second order attributes* $\mathcal{AT}^{+2} = (\mathcal{AT}^{+1})^{+1}$. Similarly, we defined the *positive n-th order attributes*

$$\mathcal{AT}^{+n} = \begin{cases} \mathcal{AT}, & n = 0; \\ (\mathcal{AT}^{n-1})^{+1}, & n > 0. \end{cases}$$

And we defined the *negative n-th order attributes*

$$\mathcal{AT}^{-n} = \begin{cases} \mathcal{AT}, & n = 0; \\ (\mathcal{AT}^{-(n-1)})^{-1}, & n > 0; \end{cases}$$

An *n-th order attributes* is the set: $\mathcal{AT}^n = \mathcal{AT}^{+n} \cup \mathcal{AT}^{-n}$.

Definition 4-3. An *(n-th order) information flow (of AT)* is a triple $(\mathcal{F}_i, \{AT\}^n, \mathcal{A}_i)$ the steps with $\{AT\}^n$ where $\mathcal{F}_i \subseteq \mathcal{F}$ is the step set, $\{AT\}^n$ is the n-th order attributes of AT and $\mathcal{A}_i \subseteq \mathcal{A}$ is the activity set.

N-th Order Attributes Extraction Rule.

$$\begin{aligned} \forall \quad & A \in \mathcal{A}_i, \\ \exists \quad & AT \in \mathcal{AT}^0 \cup \mathcal{AT}^1 \cup ... \cup \mathcal{AT}^n, \\ \text{such that} \quad & AT \in A.V_{Er} \cup A.V_{Ew} \end{aligned}$$

$$\begin{aligned} \forall \quad & F \in \mathcal{F}_i, \\ \exists \quad & A_1, A_2 \in \mathcal{A}_i, \\ \text{such that} \quad & F.A_f = A_1 \text{and} F.A_t = A_2. \end{aligned}$$

B. Value Flow

Definition 4-4. A *value flow* is a two-tuple $(\mathcal{F}_v, \mathcal{A}_v)$ where $\mathcal{F}_v \subseteq \mathcal{F}$ is the step set, and $\mathcal{A}_v \subseteq \mathcal{A}$ is the activity set.

Value Flow Extraction Rule.

$$\begin{aligned} \forall \quad & A \in \mathcal{A}_v, \\ \text{such that} \quad & A.V_{Rr} \neq \{\} \text{and} A.V_{Rw} \neq \{\}. \end{aligned}$$

$$\begin{aligned} \forall \quad & F \in \mathcal{F}_v, \\ \exists \quad & A_1, A_2 \in \mathcal{A}_v, \\ \text{such that} \quad & F.A_f = A_1 \text{and} F.A_t = A_2. \end{aligned}$$

The extraction rule provides a basic value flow with important resources and activities. The detailed quantitative analysis on value flow can be found in section VI.

V. BPMN4SP

The BPMN4SP (BPMN for service pattern) is the extension of BPMN 2.0 [2] [3] for better support of SPDL. As mentioned in Table IV, there are three major extensions.

1) The *entity* extends the concept of *Data Object* with states. The entity can be created and modified from one state to another, and archived at last. The process is the life cycle of the entity. In the model of Artifact BP [5], the data with its life cycle is called *artifact* and we use the *entity* instead. The entity class most be declared in the head of BPMN and entity instances are defined in the process.
2) The *participator* is replaced by the *role* in our extension. The essential difference between participator and role is that role owns resources and takes its resources into the service process to create (promote) values. Instead, the concept of participator can not distinguish resources and normal attributes.
3) We extend the definition of activity with the reading and writing on the attributes of entities and the resources of roles (resp.)

Table IV provides the detailed extension of BPMN4SP on BPMN.

TABLE IV
BPMN4SP EXTENDS BPMN

	BPMN2.0		BPMN4SP	
	Description	Attributes	Description	Extended Attributes
Data Object	Basic Data object for Processes		Replaced by Entity	
Entity			Data attributes and states	Attributes,States
Participant	Partner Entity or Partner Role	Name, processRef, partnerRoleRef, partnerEntityRef, interfaceRef, participantMultiplicity, endPointRefs	Replaced by Role	
Role			Roles and its resources	Resources
Resource			Available source of wealth	
Task	Atomic activity	Name, isforcompensation	Replaced by Activity	
Activity	Work performed within process	isForCompensation, loopCharacteristic, resources, default, ioSpecification, properties, boundaryEventRefs, dataInputAssociations, dataOutputAssociations, startQuantity, compliteQuantity	Work performed within process	Name, readAttributes, writeAttributes, readResouces, writeResources, mapping, preCondition, conditionalEffect

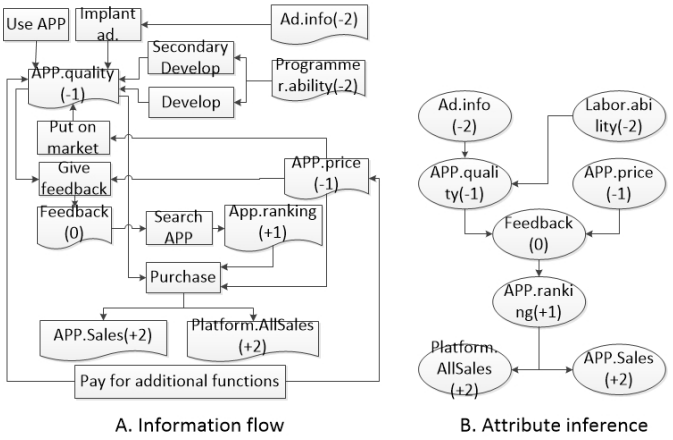

Fig. 3. The 2-order attributes information flow of feedback

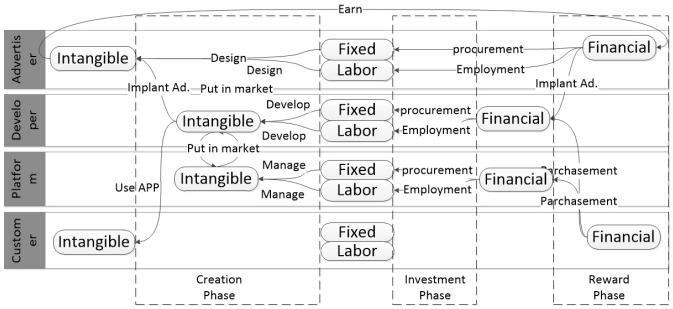

Fig. 4. The value flow of 4 roles in Mobile Application Platform

VI. CASE STUDY

This section discusses the example of MAP in detail. To analysis the value creation of each role in MAP, we format the process, which is presented in Fig. 1.

A. Information flow analysis

With the help of extraction rule defined in section IV, we can extract the n-th order attributes from origin process in Fig. 1. Fig. 3 is the second order attributes analysis result on *feedback*. In Fig. 3, part A is the information flow on second order attributes of *feedback* and part B is the attributes influence diagram. We can find out that *feedback* is influenced by the quality and price directly. And the *feedback* has an impact on the sales of APP by the influence of its ranking.

B. Value flow analysis

With the defined resources and roles, the value flow can be extracted from origin diagram by extraction rule defined in section IV. Fig. 4 presents the value flow of MAP.

To quantitative analysis the value flow of each role, we study the value flow of the **Advertiser**. The value creation process can be divided into three parts.

- Investment phase is the first step. The **Advertiser** uses 10 to buy fixed resources, 500 to employ programmers and 30 to pay the advertising fee to the **Developer**. The relative value of financial resource is $(-10)+(-500)+(-30)=-540$ now.
- Creation phase is the second step. The **Advertiser** designs the advertisement and implants into APP. In this step, 5 of 10 fixed resources and all 500 labor resources are used. The value promotion from the APP is 50. So the relative intangible resource is now $500+5+50=555$.
- Reward phase is the last step. The **Advertiser** converts the value of the advertisement (555) to income. The financial resource is now $-540+555=15$.

After the three steps, we can find that the labor and intangible resources are not changed but financial and fixed resources increased. The last result is depicted in Fig. 5. With the help of the value flow, we can now understand that the **Advertiser** earns by such a process and the return on investment is $(15+5)/(10+500+30)=3.7\%$. The discussion on how to calculate each resource changing with the influence of activities and how to re-allocate the resources to get a higher

return are beyond the scope of this paper. It will be presented in the future work.

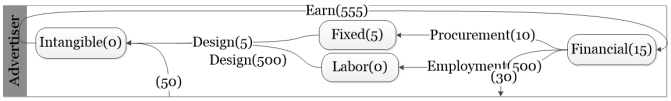

Fig. 5. Advertiser value flow

We have studied 62 enterprises from 355 public companies in the Growth Enterprise Market (GEM), China and analyzed their service pattern [6]. The GEM is the Second-board Market which is very similar to the NASDAQ Stock Market. We have extracted out six kinds service pattern: *The Long tail service pattern*, *Mulit platforms service pattern*, *Free service pattern*, *Secondary innovation service pattern*, *Un-bundling service pattern* and *Systematic service pattern*.

VII. Related Work

The technical results in the paper are closely related to three kinds of models: classical service model, business process model and enterprise business model.

A. Classical Service Model

Cassado [7] introduces a model for the business transaction of web services. Vaculin [8] hold the same opinion with us that data should not only be treated as the input and output. It is possible and necessary to define the behavior of data. A web data and artifact centric system (W-DAS) is proposed, in which data are treated as business artifacts in Vaculin's work. However we go further in this way, we not only introduce the business artifact but also defined the resources and roles to support quantitative economic analysis. Chien [9] proposes a comprehensive service pattern model assisting the design and composition of services. However the service pattern model (Chien's for short) is quite different from ours. Chien's is used to simplify the composition process and enable flexible pattern-based reuse. It is similar to workflow template. Our service pattern is used to analysis the value creation and promotion of each participator. Duan [10] proposes Service Value Broker (SVB) pattern. SVB is a special design pattern with the concept of value introduced. However it does not take the value creation of different particulars into consideration. Liang [11] presents and elaborates the concept of web service usage pattern. The web service usage pattern is a 3 level hierarchy structure including user request level, template level and instance level. However the resource and role are not formulated in this model. SAP [12] proposes a famous service modeling language named USDL (Unified Service Description Language) which plays a major participant in the Internet of Services to describe tradable services. However, USDL lacks the resource perspectives and it cannot support the necessary economic modeling and analyzing of the service pattern.

B. Business Process

The research on business process is the foundation of our work. Bhattacharya [4] formulates the artifact-centric business process model. Estanol [13] uses UML to represent each element in artifact-centric model. Richard [14] provides a brief survey on artifact-centric model. The artifact centric model is used in our work to present the life cycle of the entity. However, artifact cannot directly used to analysis the value creation, because it does not distinguish the resources with attributes. It is not possible to extract the value flow since all the data are represented as attributes.

There are many research results in business process including: Reijgers [16] discusses the declarative model of business process and proposes a research agenda for the development of modeling approach. Wong [17] provides the process semantics for BPMN. Russell [18] extends BPEL with the human task considered. Rodriguez [19] extends BPMN with data quality considered. Tzivikou [20] proposes a language for modeling business terms, which is named SeDL-C. The business process is the foundation of service pattern model. We extend BPMN2.0 to support the elements of value creation and reserve most of characteristics about business process. YAWL [21] [22] [23] is a famous workflow modeling language which is based on Petri. Aalst studies 20 workflow patterns and lays a solid foundation for the work flow pattern analysis. Furthermore, the different perspectives are also proposed in the literature. However, our work steps further and unifies three flows of three perspectives. DYNO [24] is another interesting notation to help platform providers in creating services. It is the specialized model for the platform instead of a universal one and it cannot satisfy the requirements of other service providers.

C. Enterprise Business Model

There are plenty of works about the economic analysis of enterprise business model in the research of management. Amit [25] defines how firms manage their transactions with stakeholders such as customers, partners, investors, and suppliers. It is the foundation of business model research. Magretta [26] believes that the business model is constructed on the field of business strategy. The similarities and difference between business model and service pattern is as follows:

- Both business model and service pattern are motivated to figure out how the stakeholder earns money by playing a specific role and taking some activities in business process. The essential destination of these researches is to assist modifying or re-designing the business model to get a better benefit.
- Both business model and service pattern are concentrated in the role. Based on these model, it is possible to provide some suggestions on resources allocating for a better benefit.
- Business model is an economic model while service pattern is a computer model. The former interests in economic effect and the latter focus on the relation between the business process and value creation.

Gordijn [27] [28] proposes a famous e-business modeling tools: e3-value. It is an important trial for building an information model of business model. E3-value focuses on the value flow of e-business. It achieves great success and lays the foundation for our work. However, our work makes a trial to unify more perspectives (the control flow perspective and the data perspective) which cannot be reflected in e3-value. And we believe that the information about the data (the attributes of the entity) are extremely important to explain the value creation and promotion.

VIII. Conclusion

In this paper, we address the problems of defining the business model in the service and providing basic quantitative analyzing tools (namely Q1 and Q2 in section I). We define the service pattern as the combination of 4 kinds of strategies (*Resources Allocating*, *Activities Organizing*, *Shareholders Coordinating* and *Productions Designing*). Finding that the existing research methods (service model, business process and enterprise business model) cannot cover these strategies and solve the Q2, we propose a new modeling language SPDL. SPDL consists of 4 basic language elements extracting from strategies (*resource* from Resource Allocating, *activity* from Activities Organizing, *role* from Shareholders Coordinating and *entity* from Productions Designing). Based on SPDL, an information flow analyzing tool and a quantitative value flow analyzing tool are proposed. Furthermore, we extend BPMN to BPMN4SP with the 4 basic elements considered to support SPDL. The BPMN4SP distinguishes resources with normal data and replaces the concept of the participator by the role. An example of Mobile Application Platform Service is studied in detail. The future work will mainly focus on three directions: studying the relation between the information flow and the value flow; discussing the allocation of resources in the value flow and developing the program to design service pattern.

Acknowledgment

This work was supported by National Science and Technology Supporting Program of China (No. 2013BAH10F02), National Natural Science Foundation of China under Grant (No.61272129), NewCentury Excellent Talents Program by Ministry of Education of China(No.NCET-12-0491) and Zhejiang Provincial Natural Science Foundation of China (LR13F020002, LY12F02029).

References

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

[2] S. A. White, "Business process modeling notation," *Specification, BPMI. org*, 2004.

[3] S. A. White, "Process modeling notations and workflow patterns," *Workflow Handbook*, vol. 2004, pp. 265–294, 2004.

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

[5] D. Cohn and R. Hull, "Business artifacts: A data-centric approach to modeling business operations and processes," *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering*, vol. 32, no. 3, pp. 3–9, 2009.

[6] Z. H. Wu, X. B. Wu, and M. M. Yao, *Business model innovation of modern service company a value network perspective*. Beijing: Science Press, 1 ed., 2013.

[7] R. Casado, J. Tuya, and M. Younas, "An abstract transaction model for testing the web services transactions," 2011 4-9 July 2011 2011.

[8] R. Vaculin, T. Heath, and R. Hull, "Data-centric web services based on business artifacts," 2012 24-29 June 2012 2012.

[9] L. Chien-Hsiang, H. San-Yih, and Y. I-Ling, "A service pattern model for flexible service composition," 2012 24-29 June 2012 2012.

[10] D. Yucong, A. Kattepur, and D. Wencai, "Service value broker patterns: Integrating business modeling and economic analysis with knowledge management," 2013 June 28 2013-July 3 2013 2013.

[11] Q. A. Liang, J. Y. Chung, S. Miller, and O. Yang, "Service pattern discovery of web service mining in web service registry-repository," in *e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on*, e-Business Engineering, 2006. ICEBE '06. IEEE International Conference on, (Shanghai), pp. 286–293, IEEE, 2006.

[12] J. Cardoso, A. Barros, N. May, and U. Kylau, "Towards a unified service description language for the internet of services: Requirements and first developments," pp. 602–609, 2010.

[13] M. Estanol, A. Queralt, M. R. Sancho, and E. Teniente, "Artifact-centric business process models in uml," in *Business Process Management Workshops*, Business Process Management Workshops, pp. 292–303, Springer, 2013.

[14] R. Hull, *Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges*. OTM Workshops, 2008.

[15] S. Appel, S. Frischbier, T. Freudenreich, and A. Buchmann, "Event stream processing units in business processes," in *Business Process Management*, Business Process Management, pp. 187–202, Springer, 2013.

[16] H. A. Reijers, T. Slaats, and C. Stahl, "Declarative modelingcan academic dream or the future for bpm?," in *Business Process Management*, Business Process Management, pp. 307–322, Springer, 2013.

[17] P. Y. Wong and J. Gibbons, "A process semantics for bpmn," pp. 355–374, Springer, 2008.

[18] 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, 2007.

[19] A. Rodrłguez, A. Caro, C. Cappiello, and I. Caballero, "A bpmn extension for including data quality requirements in business process modeling," in *Business Process Model and Notation*, Business Process Model and Notation, pp. 116–125, Springer, 2012.

[20] C. Tziviskou, M. Palmonari, M. Comerio, and F. De Paoli, "Sedl-c: A language for modeling business terms in service descriptions," 2013 June 28 2013-July 3 2013 2013.

[21] W. Van Der Aalst and K. M. Van Hee, *Workflow management: models, methods, and systems*. MIT press, 2004.

[22] W. M. Van Der Aalst and A. H. Ter Hofstede, "Yawl: yet another workflow language," *Information systems*, vol. 30, no. 4, pp. 245–275, 2005.

[23] B. Kiepuszewski, A. H. ter Hofstede, and W. M. van der Aalst, "Fundamentals of control flow in workflows," *Acta Informatica*, vol. 39, no. 3, pp. 143–209, 2003.

[24] U. Scholten, R. Fischer, C. Zirpins, and S. Scholten, "Dyno: A notation to leverage dynamic network effects in paas ecosystems," in *Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on*, Service-Oriented Computing and Applications (SOCA), 2011 IEEE International Conference on, pp. 1–6, IEEE, 2011.

[25] R. Amit and C. Zott, "Value creation in e-business," *Strategic management journal*, vol. 22, no. 6-7, pp. 493–520, 2001.

[26] M. Morris, M. Schindehutte, and J. Allen, "The entrepreneur's business model: toward a unified perspective," *Journal of business research*, vol. 58, no. 6, pp. 726–735, 2005.

[27] J. Gordijn, "E 3-valuein a nutshell," 2002.

[28] J. Gordijn and H. Akkermans, "Designing and evaluating e-business models," *IEEE intelligent Systems*, vol. 16, no. 4, pp. 11–17, 2001.