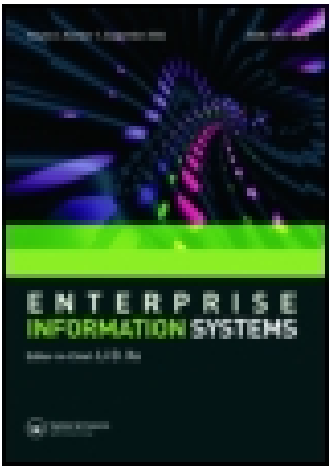

Enterprise Information Systems
Publication details, including instructions for authors and subscription information:
http://www.tandfonline.com/loi/teis20

Enterprise Pattern: integrating the business process into a unified enterprise model of modern service company

Ying Li[a], Zhiling Luo[a], Jianwei Yin[a], Lida Xu[b], Yuyu Yin[cd] & Zhaohui Wu[a]

[a] College of Computer Science, Zhejiang University, Hangzhou, China

[b] Department of Information Technology, Old Dominion University, Norfolk, VA, USA

[c] School of Computer, Hangzhou Dianzi University, Hangzhou, China

[d] Electric Engineering School, Zhejiang University, Hangzhou, China

Published online: 12 Jun 2015.

To cite this article: Ying Li, Zhiling Luo, Jianwei Yin, Lida Xu, Yuyu Yin & Zhaohui Wu (2015): Enterprise Pattern: integrating the business process into a unified enterprise model of modern service company, Enterprise Information Systems, DOI: 10.1080/17517575.2015.1053415

To link to this article: http://dx.doi.org/10.1080/17517575.2015.1053415

Enterprise Information Systems, 2015
http://dx.doi.org/10.1080/17517575.2015.1053415

Enterprise Pattern: integrating the business process into a unified enterprise model of modern service company

Ying Li[a], Zhiling Luo[a], Jianwei Yin[a]*, Lida Xu[b], Yuyu Yin[c,d] and Zhaohui Wu[a]

[a]*College of Computer Science, Zhejiang University, Hangzhou, China;* [b]*Department of Information Technology, Old Dominion University, Norfolk, VA, USA;* [c]*School of Computer, Hangzhou Dianzi University, Hangzhou, China;* [d]*Electric Engineering School, Zhejiang University, Hangzhou, China*

(*Received 20 October 2014; accepted 4 May 2015*)

Modern service company (MSC), the enterprise involving special domains, such as the financial industry, information service industry and technology development industry, depends heavily on information technology. Modelling of such enterprise has attracted much research attention because it promises to help enterprise managers to analyse basic business strategies (e.g. the pricing strategy) and even optimise the business process (BP) to gain benefits. While the existing models proposed by economists cover the economic elements, they fail to address the basic BP and its relationship with the economic characteristics. Those proposed in computer science regardless of achieving great success in BP modelling perform poorly in supporting the economic analysis. Therefore, the existing approaches fail to satisfy the requirement of enterprise modelling for MSC, which demands simultaneous consideration of both economic analysing and business processing. In this article, we provide a unified enterprise modelling approach named Enterprise Pattern (EP) which bridges the gap between the BP model and the enterprise economic model of MSC. Proposing a language named Enterprise Pattern Description Language (EPDL) covering all the basic language elements of EP, we formulate the language syntaxes and two basic extraction rules assisting economic analysis. Furthermore, we extend Business Process Model and Notation (BPMN) to support EPDL, named BPMN for Enterprise Pattern (BPMN4EP). The example of mobile application platform is studied in detail for a better understanding of EPDL.

Keywords: Enterprise Pattern; modern service company; business model; business process; value creation and economic analysis

1. Introduction

Since 1990s, the global industry structure has been gradually speeding up its process to focus on service. In parallel to the service orientation of the global economy, *modern service* has gradually detached itself from the traditional service and gained momentum to expand rapidly. We, based on management science literature, outline the characteristics of *modern service company* (MSC) (Wu, Wu, and Yao 2013) as follows:

- Internet-based: the services provided are mostly based on Internet. On the one hand, MSC can build the connection with customers via Internet. On the other hand, the services are provided through Internet (e.g. online shopping). Additionally, the feedback from customers is collected from the Internet.

*Corresponding author. Email: zjuyjw@cs.zju.edu.cn

- Value increment: the price of services is promoted in MSC comparing with traditional service companies.
- Agility: the services can be modified and optimised quickly to satisfy market changes and customer requirements. Hence, the innovation of the service and business model is agile.
- Open: the technique, platform and service are open to customers (especially the enterprise kind of customer). It is possible to extend the original service to develop a personalised or composite one.
- Structured: technology, information and resources (especially intangible resources, e.g. brand) can be organised and managed in traceable structures.
- Combination: most MSCs, rather than only involved in only one domain, combine two or more domains (e.g. IT and education). The requirements and techniques are combined to provide a unified solution, usually effectively and efficiently.

Taobao, a typical MSC in China, is established in 2003 with its primary business being the online mall (http://www.taobao.com). Without selling any goods, *Taobao* provides platform service to sellers and attracts buyers, a business model called C2C (customer to customer) in e-business. Not charging sellers for the basic platform service, the company has two major sources of income: the advertisement and the value-added services (e.g. web site decorating service). Like other MSCs, *Taobao* takes the collection of information and resources seriously. With the trading volume reaching 17.2 billion yuan (nearly 2.7 billion dollars) in a single day, it is currently the biggest online mall in China. Therefore, its trading data (logs) have unique value in customer behaviour analysis. The modelling of MSCs like *Taobao* is an interesting and challenging topic for researchers from both management science and information science.

Previous research on modelling of MSCs in these two different domains understandably holds a discipline-specific perspective. Figure 1 presents perspectives from the two aforementioned domains: the management science (the above dotted block) and the information science (the below one). As described in Van Der Aalst (2005), the *resource perspective*, common in management science, prioritises an organisational structure anchor to the workflow in the form of human and device participants responsible for executing tasks. In contrast, the information science highlights the *control perspective*, which describes tasks and their execution ordering through different constructors, which permit flow of execution control, and the *data perspective*, which deals with business and processing data. These three perspectives form a structure of three layers, descending in abstractness and ascending in concreteness from top to bottom, with the control flow perspective at the bottom layer, the data perspective in the middle and the resources perspective on the top. For example, the control flow perspective deals with how a

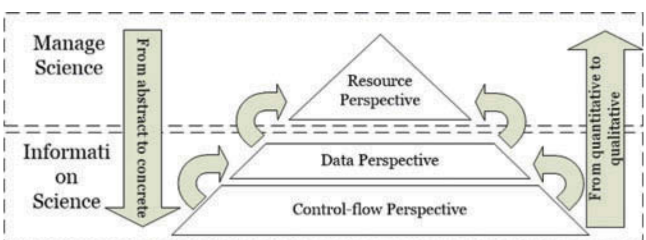

Figure 1. Three-layer perspectives and the relations.

company provides a service and which tasks it should accomplish when a customer gives a bad comment. Instead, in the data perspective, tasks are ignored or hidden since the data and its changes are centralised. From the bottom to top, the higher the position, the more qualitative the perspective becomes and vice versa.

Typically, the three perspectives are reified by three models:

- *Business process (BP) model* instantiates the BP from the *control flow perspective*. Both Business Process Execution Language (BPEL) (Barreto et al. 2007) and Business Process Model and Notation (BPMN) (Ruiz et al. 2012; White 2004) solve the problem of business modelling and process optimisation, with great success in the last few years. However, this model does not reflect the necessary economic elements supporting the high-level (the top) perspective analysis.
- *Artefact-centric (AC) model* (Bhattacharya et al. 2007; Cohn and Hull 2009), a special kind of BP model in which the BP is defined by data change in its life cycle, focus on the *data perspective*.
- *Enterprise business (EB) model* (Amit and Zott 2001; Morris, Schindehutte, and Allen 2005), a framework effective in assisting business strategy analysis while ineffective in process analysis and useless in relationship analysis between BP and business strategy, emphasises the *resource perspective*.

These three perspectives describe different parties of the MSC and none of them alone can fully model the enterprise and inclusively cover all the characteristics of MSC. Therefore, in this study, we aimed to build a unified model which we name *Enterprise Pattern* (EP) for MSC. Our model hinges upon four basic strategies to link the elements of EB to AC and BP:

- Resources allocating (RA): resource is the key element of the resource perspective. The strategy of resource allocation determines the amount of labour and money spent in each activity.
- Activities organising (AO): activity is the basic element of the control flow perspective. In this strategy, control patterns (e.g. gateways) are employed to connect activities in BP construction.
- Shareholders coordinating (SC): the shareholder is the participant of BPMN. Its essential extension is owing the resources. In classical BPMN, the participant is the one taking responsibility to a task. However, in the resource perspective in our model, when participating in the activity (task), the participant consumes some resources and creates others. Therefore, we redefine participant as those who owns and uses resources to participate in the activity.
- Productions designing (PD): the production of MSC refers to service. The strategy of PD contains the way of value realisation from the resource perspective, with the price strategy as its most useful part.

Ultimately, the unified model (*EP*) will be a combination of these four strategies. Currently, we have already finished the first step. In this article, we propose a formal modelling language, *Enterprise Pattern Description Language* (*EPDL*) – the first step which defines the four basic elements of the MSC from the four aforementioned strategies (*resource* from RA, *activity* from AO, *participant* from SC and *entity* from PD), and define an *extension of BPMN for Enterprise Pattern* (*BPMN4EP*) to support the language elements.

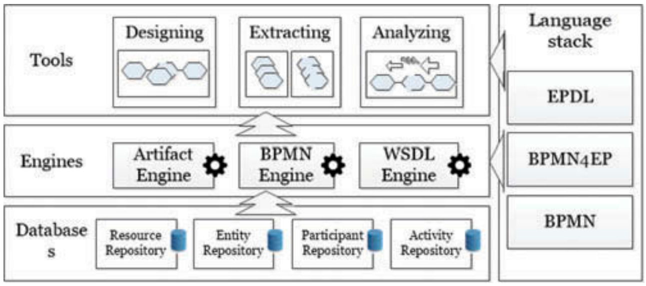

Figure 2. The architecture of Enterprise Pattern analyses.

Figure 2 presents the basic architecture of our EP. The left part, the implementation of the EP designing, extracting and analysing tools, includes three engines and four databases in this architecture. The right part is the language stack, with the BPMN occupying thee bottom layer, the BPMN4EP the middle layer and the EPDL the top layer.

This article makes the following technical contributions:

- The EP of MSC constructed by four strategies (RA, AO, SC and PD) is introduced. This pattern combines the three perspectives, namely the control flow perspective, the data perspective and the resource perspective, and bridges the gap between the BP model and the enterprise economic model of MSC.
- The first step, EPDL, which covers four basic elements (resource, activity, participant and entity) in strategies (RA, AO, SC and PD) is proposed in this article.
- BPMN4EP, which implements EPDL and extends the usage scope of BPMN, is introduced as an extension of BPMN and artefact BP.

The rest of this article is organised as follows. Section 2 presents a motivating example of our research; Section 3 formulates the EPDL; Section 4 discusses three flows corresponding to the three perspectives; Section 5 introduces BPMN4EP and its detailed comparison with BPMN; Section 6 elaborates the process of modelling MSC; Section 7 raises key issues relevant to the study; Section 8 reviews the related works. Finally, Section 9 concludes this article and provides an outlook on perspective continuations of our work.

2. A motivating example

The *Mobile Application Software Company*, the *developer* in the domain of mobile application development industry, is a typical MSC. The *mobile application* (*APP*, for short), each of which can (in most case) only be deployed on one kind of mobile phone operating system (e.g. IOS, WP or Android), is a kind of executable program deployed on *smart mobile phones* (such iPhone and Android Phone) to be purchased by the *customer*, the buyer of such programs. Regularly, the developer, instead of selling its APPs to the customer directly, puts its APPs on a market, the application store platform (*platform*). The platform collects huge amounts of APPs, presents them in a sequence of their rankings and provides the purchasing channel for the customer. Once a customer buys an APP from the platform, the fee will be allocated to the platform and developer by some ratio. A famous example of platform is the *APP Store* of Apple Inc. which claims over 40

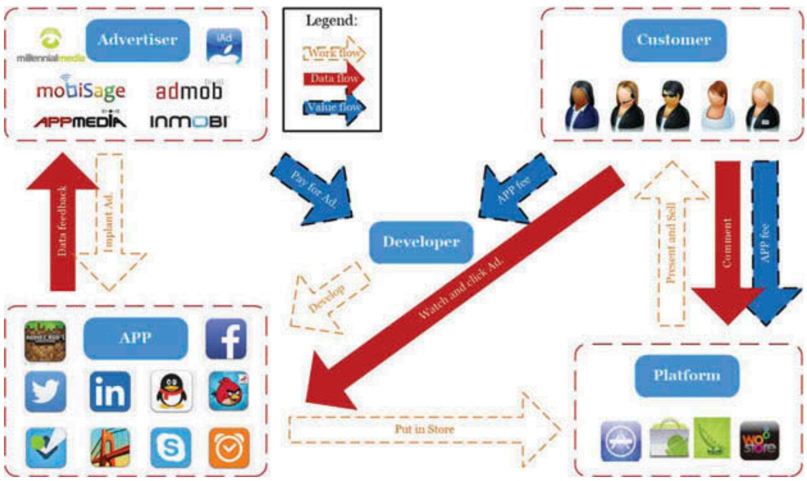

Figure 3. The data flow, work flow and value flow of the mobile application development industry.

billion downloads from the *APP Store* and over 500 million active users. Another interesting participator, which puts *advertisements* in the APP by purchasing to the developer, is the *advertiser*.

Figure 3 presents the major participators and three basic directed relations:

- Work flow: at first, the developer develops the APP which is sequentially embedded advertisement, is put in Store (the platform) and finally bought by the customer.
- Data flow: two kinds of important data are in operation – the advertisements clicked and the comments from the customer.
- Value flow: the advertiser pays for the advertisement presentation and the customer pays APP fee to the developer and platform.

These three relations correspond to three perspectives (control flow perspective, data perspective and resources perspective). From this example, which will be used in the following sections to study the formalised definition of each language element of EPDL, we can find out that these relations are complicated and interdependent. The existing modelling approaches can only cover one relation from one perspective, as will be discussed in Section 8.

3. Enterprise Pattern Description Language

This section formulates the key syntaxes and basic notions of EPDL. Figure 4 presents the relation of basic notions by Unified Modeling Language (UML) diagram. There are four basic elements: resource, activity, participant and entity, in EP strategies (RA, AO, SC and PD). A participant class with its many resources coordinates with the attributes pertaining to an entity class to redefine the concept of activity. The step denotes the execution order, from one activity to another.

Figure 4. The UML diagram of basic notions.

3.1. Basic set

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

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 $t \in \mathcal{C}_\varepsilon$ is an entity type, $\mathbf{DOM}(t) = \mathbf{ID}_t$.

3.2. Entity

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

Definition 2 (entity instance): *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 *a partial mapping that assigns each attribute* $AT \in \mathbf{AT}$ *an element in its domain* $\mathbf{DOM}(\tau(AT))$*, and* $q \in Q$ *the current state.*

The concept of entity embraces 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 1: In the motivation example, APP is an important entity. Its class and instance are presented in Table 1. The APP instance we used here is *Angry Birds* with its state on sale (available on *APP store*) its price 0 (this version being free) and its ID id391231.

3.3. Resource

The resource, which is a valuable object in process, falls into four kinds, as defined in management science literatures (as Figure 5).

Table 1. Entity example: APP entity and an APP instance.

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

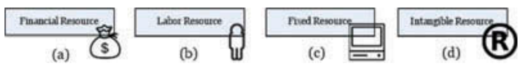

Figure 5. Four resource notations in EPDL diagram: (a) financial resource, (b) labour resource, (c) fixed resource and (d) intangible resource.

- The *financial* resource is the money in each form.
- The *fixed* resource includes houses, office equipment, etc.
- The *labour* resource refers to available manpower.
- The *intangible* resource includes the brand and information resources.

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

3.4. Participant

Definition 3 (participant class): *A participant class is a triple* $(C_P, \mathbf{RE}, \tau)$ *where* $C_P \in \mathcal{C}_\mathcal{P}$ *stands for the participant class name*, $RE \subseteq \mathcal{R}\varepsilon$ *a finite set of resources*, $\tau: RE \to \mathrm{T}$ *a total mapping*.

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

In EPDL, the key characteristic of the participant, under which the enterprise itself is subsumed as a participant to simplify discussion, is owning resources such as financial, fixed, labour and intangible resources.

Example 2: In the motivation example, there are four participants: developer, customer, platform and advertiser. Table 2 presents a developer participant class and an instance, whose ID is id191231.

3.5. Schema

Definition 5 (schema)**:** *The schema a finite set with* $\Gamma = \Gamma_E \cup \Gamma_P$ *consists of two key parts: an entity schema and the participant schema*.

Table 2. Participant example: developer and developer instance.

Developer RESOURCES	Developer instance
Labour: float Financial: float Fixed: float Intangible: float	ID:id191231 RESOURCE: Labour: 20.3 Financial: 32000 Fixed: 1230000 Intangible: 31233

The *entity schema* entails a finite set Γ_E of entity classes with distinct names in such a way that every class referenced in Γ_E also occurs in Γ_E. The *participant schema* followed Γ_E is a finite set Γ_P of participant classes with distinct names such that every class referenced in Γ_E.

3.6. Atom

Definition 6 (atom): *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 participant class) C in Γ;
(3) $\textbf{DEFINED}(t, D)$, where t is either an instance of entity class C and D an attribute in C or an instance of participant class C and D a resource in C;
(4) $\textbf{NEW}(t, D)$, where t is either an instance of entity class C and D an entity typed attribute in C or an instance of participant class C and D an entity typed resource in C;
(5) $s(t)$ (a (state) atom), where t is an 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: An example of a condition is as follows:

$$\textbf{DEFINED}(\text{id391231}, \text{APP.price}) \wedge \text{ on sale}(id391231)$$

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

3.7. Activity

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

M describes the mapping whose input attributes influenced each output. Considering a sequence of input attributes $xe_1, \ldots, xe_k$ and output attributes $ye_1, \ldots, ye_l$ $(k, l \geq 1)$. $\mathcal{M} \in \mathcal{R}^{k \times l}$ 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 4: In the motivation example, an example of purchasing APP is presented in Table 3. An entity instance (id391231) is used and a participant, namely id231441, has participated in this activity which reads the price and sales of APP as input attributes, and the financial resource of customer as input resources. While in the mapping relation, the amount of APP sales reflects itself. The precondition consists of three parts:

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

There are two effects of this activity:

(1) the sales of APP increases 1 and
(2) the money of customer decreases.

Figure 6 illustrates the activity of purchasing APP in EPDL diagram, with the arrows in the top edge of the activity representing the consuming and creating of resources. In Figure 6, the customer's financial resource is consumed.

3.8. Rule

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

Table 3. Activity example.

Purchasing APP	
Entity	id391231:APP
Participant	id231441:Customer
Read attributes	id391231.Price, id391231.sales
Write attributes	id391231.sales
Read resource	id231441.Financial
Write resources	id231441.Financial
Mapping relation	id391231.Sales → id391231.Sales
Precondition	**DEFINED** $(id391231, Price) \wedge$ $id391231.Price \leq id231441.Financial \wedge$ on sale $(id391231)$
Effect	$id391231.Sales = id391231.Sales + 1 \wedge$ $id231441.Financial = id231441.Financial - id391231.\ Price$

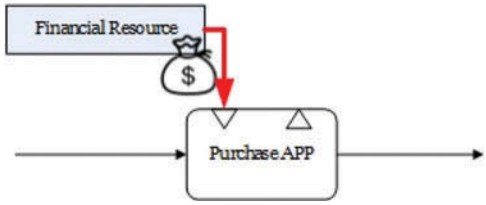

Figure 6. The activity notation in EPDL diagram.

- If φ invoke

$$\sigma(xe_1, \ldots, xe_k;\ ye_1, \ldots, ye_l;\ xr_1, \ldots, xr_m;\ yr_1, \ldots, yr_n), \text{or}$$

- If φ change state to ϕ.

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

3.9. Control step

Definition 9 (control step)**:** *A control step is a tuple* (f, A_f, A_t) *where f is the ID of this step*, A_f *the activity emanating outward, and* A_t *the activity pointing inward.*

The step connecting from A_f to A_t means invoking A_t once A_f finished.

4. Three flows and three perspectives

As we mentioned earlier, EPDL combines three basic perspectives: control flow perspective, data perspective and resource perspective and the corresponding three flows: work flow, data flow and value flow.

- Work flow: the operating order and control pattern of the activities from *control flow perspective* (Viriyasitavat, Da Xu, and Martin 2012; Viriyasitavat and Da Xu 2014). The work flow is present in the EPDL work flow diagram. A work flow diagram is designed and completed with needed information by the enterprise manager. The aspects which cannot be reflected in the work flow diagram, especially those about the attributes of entities, should be defined. The work flow diagram and the appended information make up the EPDL model of MSC. The detailed comparison between EPDL work flow diagram and BPMN is studied in Section 5.
- Data flow: the data dependency and life cycle from *data perspective*. It is present in both the EPDL data flow diagram and the life cycle diagram. Differing from the EPDL work flow diagram, these two diagrams are extracted from the EPDL model.
- Value flow: the value consuming and creating trace from the *resource perspective*. It is present in the EPDL value flow diagram and extracted from EPDL model.

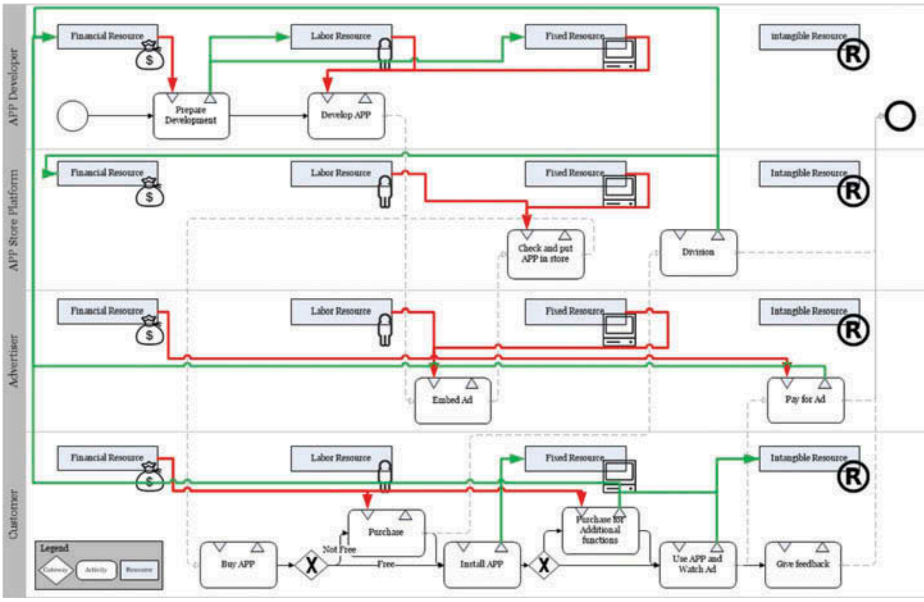

Figure 7. The swimlane diagram of mobile application platform service example.

4.1. *EPDL work flow diagram*

With the definition of basic elements of EPDL, the example of motivation can be expressed as a swimlane diagram.

In Figure 7, four participants are represented by four swimlanes.

4.2. *EPDL data flow diagram*

Definition 10 (referenced by)**:** *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 and modifies* AT_2 *and* $\mathcal{M}(AT_1, AT_2) = 1$.

Definition 11 (positive first-order attributes)**:** *Considering an attribute set* $\mathcal{AT}$*, we assume that the set* $\mathcal{AT}^{+1}$ *represents the attribute set which is referenced by each attribute in* $\mathcal{AT}$*,* $\mathcal{AT}^{+1}$ *being the positive first-order attributes of* $\mathcal{AT}$.

Theorem 1 (first-order attributes decision rule)**:** $\forall AT \in \mathcal{AT}^{+1}, \exists AT' \in \mathcal{AT}$ *such that* $AT \lhd AT'$.

Theorem 2: (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* as follows:

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

And we defined the *negative n-th order attributes* as follows:

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

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

Definition 12 (n-th order)**:** *An (n-th order) data 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,* $\{AT\}^n$ *the n-th order attributes of AT and* $\mathcal{A}_i \subseteq \mathcal{A}$ *is the activity set.*

Theorem 3 (n-th order attributes extraction rule):

$$\forall A \in \mathcal{A}_i, \exists AT \in \mathcal{AT}^0 \cup \mathcal{AT}^1 \cup \ldots \cup \mathcal{AT}^n, \, such\ that\ AT \in A.V_{Er} \cup A.V_{Ew}$$
$$\forall F \in \mathcal{F}_i, \exists A_1, A_2 \in \mathcal{A}_i, \, such\ that\ F.A_f = A_1\ and\ F.A_t = A_2.$$

With the help of the extraction rule, we can extract the n-th order attributes from origin process as shown in Figure 7. Figure 8 shows the second-order attributes analysis result from *feedback*. In it, part A reifies the information flow on second-order attributes of *feedback* while part B illustrates the attributes influence. Obviously, *feedback*, which has an impact on the sales of APP by the influence of its ranking, is influenced by the quality and price directly.

Differing from BPMN, BPMN4EP defines the life cycle of the entity and allows possible extraction of that of important entities from its description. Figure 9 presents the life cycle of APP, which also determines the semantics of some activities in the motivation example where the activity *check APP* is explained as transferring the states of APP from *unchecked* to *not on sale*.

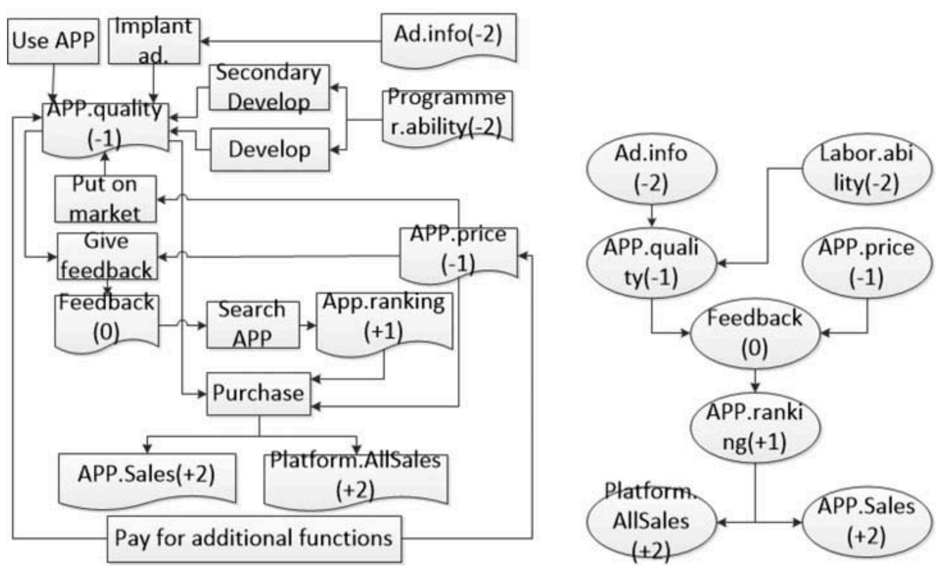

Figure 8. The two-order attributes information flow of feedback. (a) Data flow diagram. (b) Attribute inference.

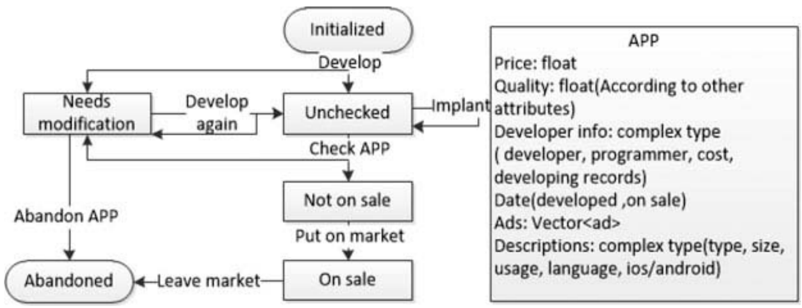

Figure 9. The life cycle of APP.

4.3. EPDL value flow diagram

Definition 13 (value flow)**:** *A value flow is a two-tuple* $(\mathcal{F}_v, \mathcal{F}_v)$ *where* $\mathcal{F}_v \subseteq \mathcal{F}$ *stands for the step set, and* $\mathcal{A}_v \subseteq \mathcal{A}$ *the activity set.*

Theorem 4 (value flow extraction rule):

$$\forall A \in \mathbf{A}_v,\ such\ that\ A.V_{pr} \neq \emptyset\ and\ A.V_{pw} \neq \emptyset.$$

$$\forall F \in \mathsf{F}_v,\ \exists\ A_1, A_2 \in \mathsf{A}_v,\ such\ that\ F.A_f \neq A_1\ and\ F.A_t = A_2.$$

The extraction rule provides a basic value flow with important resources and activities. With the defined resources and participants, the value flow can be extracted from EPDL model by extraction rule. Figure 10 presents the value flow of the motivation example. Interestingly, the four participants create (promote) its value in totally different ways:

- The advertiser embeds advertisement in APP to exchange the benefit, which can be added value of their assets or the promotion of its brand. This is a very common and basic business pattern of the advertiser.
- The developer, the originating force of the value flow, develops APP, the basic and most essential production in the whole process. Without it, no one can benefit from the process. The developer, who owns APP as its own resources,

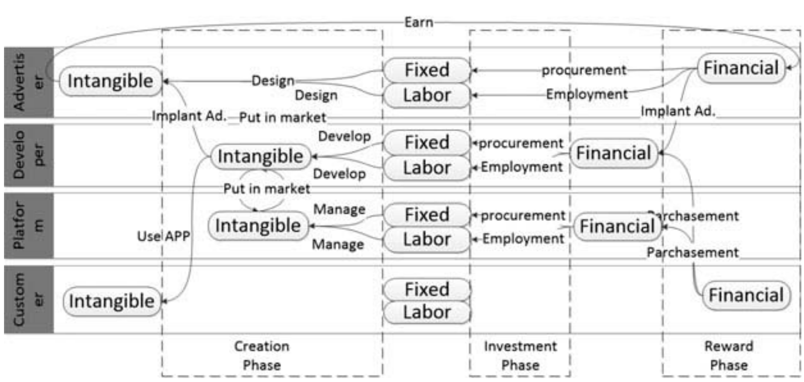

Figure 10. The value flow of four participants in mobile application platform.

cannot convert APP into income directly. It has to put the APP in the platform. Another interesting path can be found in value flow: the path from the developer to the advertiser. That is, the developer pays for the promotional cost of APP advertisement, while at the same time the advertiser pays for the advertisements implanted in APP. What if the two amounts are equal? If the amount paid from the developer to the advertiser is the same as the one flowing oppositely, the developer could implant the advertisement to other APPs without paying, a popular pattern of developer in practice. The developers get together to form a group and implant advertisements of other developers without paying.

- The platform which earns from each deal done in a APP market spends its operating cost to maintain the market, a typical EP called the *Third Platform* pattern.
- The customer pays for the APP and gets the right to use it to satisfy the need of customer.

To quantitatively analyse the value flow of each participant, we studied the value flow of the advertiser whose value creation process can be divided into three parts:

- The first step is the investment phase. 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.
- The second step is the creation phase. The advertiser designs the advertisement and implants it into an APP. In this step, 5 of 10 fixed resources and all 500 labour resources are used. The value promotion from the APP is 50. So the relative intangible resource becomes $500 + 5 + 50 = 555$ now.
- The last step is the reward phase, in which the advertiser converts the value of the advertisement (555) to income, with the financial resource of $-540 + 555 = 15$.

After the three steps, we can find that though the labour and intangible resources remain unchanged, the financial and fixed resources increase. The last result is depicted in Figure 11. With the help of the value flow, we can now understand how the advertiser makes profit by such a process and work out that the return on investment is $(15 + 5)/(10 + 500 + 30) = 3.7\%$. The discussion on how to calculate each resource with the influence of activities and how to reallocate the resources to get a higher return is beyond the scope of this article and will be presented in the future work.

4.4. *Three flows*

This section has presented the three flows and three perspectives from the EPDL model. These three flows have the following relations:

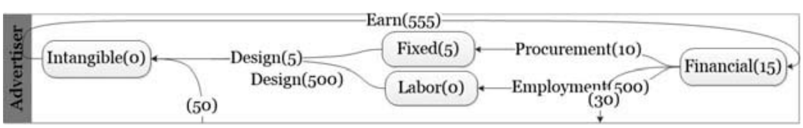

Figure 11. Advertiser value flow diagram.

- The EPDL work flow is the basic one which can cover most elements (activities, event, gateways and participant). With the appended information about the entity and resource, the EPDL model can be converted into the form of BPMN4EP, the extension of BPMN with the need element of EPDL, in the format of XML like BPMN. The EPDL data flow diagram, entity life cycle diagram and EPDL value flow diagram can all be extracted out from the EPDL model.
- These three flows, changing simultaneously when one of them is modified, reflect the different characteristics of the EPDL model from three perspectives: control flow perspective, data perspective and resource perspective.
- The AC BP model (Bhattacharya et al. 2007; Cohn and Hull 2009) integrates data into the control flow with great success. EPDL moves a step further to decide the characteristics of the AC BP and unifies all the three flows and three perspectives.

5. BPMN4EP

The BPMN4EP (BPMN for EP), the extension of BPMN 2.0 (Ruiz et al. 2012; White 2004) for better support of EPDL, offshoots into three major extensions, as mentioned in Table 4.

Table 4. BPMN4EP extends BPMN.

	BPMN2.0		BPMN4EP	
	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	Participant and its resources	Resources
Resource			Available source of wealth	
Task	Atomic activity	Name, is for compensation	Replaced by activity	
Activity	Work performed within process	is For Compensation, loopCharacteristic, resources,default, ioSpecification, properties, boundaryEventRefs, dataInputAssociations, dataOutputAssociations, startQuantity, compliteQuantity	Work performed within process	Name, readAttributes, writeAttributes, readResouces, writeResources, mapping, preCondition, conditionalEffect

(1) The *entity*, extending the concept of *Data Object* with states, can be created and modified from one state to another, and archived at last, the process being the life cycle of the entity. In the model of AC BP, (Cohn and Hull 2009), the data with its life cycle is called the *artefact* for which we use the *entity* instead. Its class is mostly be declared in the head of BPMN and instances are defined in the process.
(2) The *participant* extends itself by owning resource and taking its resources into the service process to create (promote) values.
(3) We extend the definition of *activity* to reading and writing on the attributes of entities and the resources of participants (resp.)

Table 4 provides the detailed extension of BPMN4EP on BPMN.

6. Enterprise modelling

As suggested in the motivation example, the software enterprise is a typical MSC whose complete process of modelling is to be discussed in this section.

- Analysing background: there are three important categories of shareholders, Advertiser, Platform and customer in addition to the software enterprise itself, which is perceived as a special participant (Developer). Thus, there exist four participants whose activities need to be prepared and whose life cycles of the corresponding APP need to be determined.
- Designing EP model: at first, the EPDL work flow diagram (Figure 7) can be drawn in a swimlane diagram. In addition to the basic BPMN elements (activities, events and gateways), the resources and the connection lines (consuming and creating) should be appended. After that, the definition and its related data about the entity need to be input in the EP model. At last, the EP model can be present as a BPMN4EP document.
- Extracting flows: in addition to the work flow, the data flow and value flow should be extracted from the EP model (BPMN4EP exactly) by the extracting rules. The EP data flow diagram is shown in Figure 8, the life cycle of the APP in Figure 9 and the EP value flow diagram in Figure 10.

The three aforementioned steps make up the basic process of modelling an enterprise while the management of the EP model is part of our future work which will not be addressed in this article.

7. Discussion

In this section, we discuss some aspects of EPDL.

- The EP work flow diagram accommodates all the activities, events, gateways and resources, some of which can not be irrelevant in developer analysis. Thus, we can simplify the original diagram by removing the factors not directly relevant to such analysis. The simplified diagram, present in Figure 12, show clearly that the income of the developer derives from the platform division, the advertiser payment and the customer the payment by purchasing for additional functions.

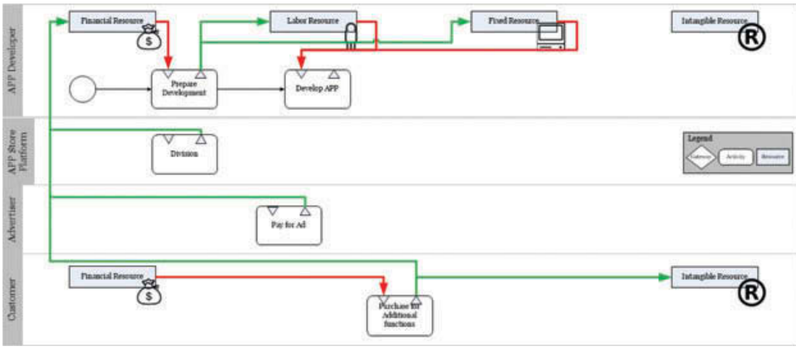

Figure 12. The simplified EP work flow diagram of the motivation example.

- The four participants are classes instead of instances. Under this framework, the platform is a class of the APP store platform, while the *APP Store* of Apple Inc. remains an instance of platform. Both practically, a participant class can have more than one instance. That is, the developer can contact two or more advertisers (e.g. *admob* and *mobisage*) and the APP can simultaneously appear in two or more platforms (e.g. *APP Store* and *PP mall*). Additionally, while the customer represents all the instances (the person) of the destination people, the developer can also represent two or more instances, if the competitors (other software enterprises) of the destination software enterprise are considered. However, competition is beyond the main scope of this article and will be studied in our future work.
- For diagramming purposes, please download a Visio tool of EPDL workflow diagram at https://sourceforge.net/p/spdlvisio/.
- We have studied 62 enterprises from 355 public MSC in the growth enterprise market, the second-board market which is very similar to the NASDAQ Stock Market, in China, and analysed their EPs (Wu, Wu, and Yao 2013). We have found six EPs: the *long tail EP*, *mulitple platforms EP*, *free EP*, *secondary innovation EP*, *un-bundling EP* and *systematic EP*.

8. Related work

Previous literature has outlined three models relevant to the enterprise modelling of MSC: classical service model, BP model and EB model.

8.1. Classical service model

The classical service model has been first raised by Casado, Tuya, and Younas (2011) to address transaction of web service and later elaborated by Vaculin, Heath, and Hull (2012) who share the same view with us in that data should be treated as both input and output and that it is not only possible but also necessary to define data behaviour. In Vaculin's work, a web data and AC system (W-DAS) is proposed, in which data are treated as business artefacts. However, we go further than him in this way, by not only introducing the business artefact but also defining the resources and participants to support quantitative economic analysis. A second model, the comprehensive service pattern model assisting the design and composition of services, is proposed by Chien-Hsiang, San-Yih, and I-Ling (2012). This model differs greatly from that of ours since it is primarily designed to simplify the composition process and enable flexible pattern-based reuse. In contrast, our EP has unique advantage in analysing the value creation and promotion potential of each participant. A third model, the Service Value Broker pattern, a special design pattern introducing the concept of value, was pinpointed by Duan, Kattepur, and Wencai (2013). It was partial since it did not take the value creation of different particulars into consideration. A fourth pattern, the concept of web service usage pattern, was presented by Liang et al. (2006). Although Liang tried to include a three-level hierarchy structure including user request level, template level and instance level, he failed to formulate the resource and participant in this model. In addition, SAP Inc. (Cardoso et al. 2010) proposes an influential service modelling language named Unified Service Description Language, which plays a major participant in the Internet of Services to describe tradable services. However, due to its lack of the resource perspectives, this model cannot support the necessary economic modelling and analyses of the MSC concurrently.

8.2. Business process

Research on the foundation of our work, BP, has been thriving. To name a few, Bhattacharya et al. (2007) formulates the AC BP model; Calvanese et al. (2014) uses UML to represent each element in an AC model and Hull (2008) provides a brief survey on AC model. Based on these studies, we adopt the AC model to present the life cycle of the entity in this study. However, the model proposed in previous literature cannot be directly used to analyse the value creation, since it does not separate resources from attributes. Therefore, we extend the model by adding a new construct about resources to distinguish them from attributes so as to extract the value flow.

Among the many researchers who have been engaged in studies of the business process, some are more conspicuous than others. Notably, Meyer et al. (2013) discuss the modelling and enacting complex data dependence; Appel et al. (2013) capture the stream of events; Reijers, Slaats, and Stahl (2013), by elaborating the declarative model of BP, propose a research agenda for the development of modelling approach; Wong and Gibbons (2008) provide the process semantics for BPMN; Russell and Van Der Aalst (2007) extend BPEL to the human task; Rodrguez et al. (2012) expand BPMN to include data quality consideration and Tziviskou et al. (2013) invent a language named Service Description Language for Contract specification for modelling business terms. Xu (2011), Xu et al. (2009, 2008, 2012) and Xu and Viriyasitavat (2014) discuss the new trends of BPM. Wang and Xu (2008) capture the integration and modelling of enterprise BPM.

Building on these works, we extend BPMN2.0 to support the elements of value creation and reserve most of its characteristics about the BP. YAWL (Van Der Aalst and Van Hee 2004; Van Der Aalst 2005; Kiepuszewski, Hofstede, and Aalst 2003) is a famous workflow modelling language based on Petri, proposed by Aalst who studied 20 work-flow patterns and laid a solid foundation for the work flow pattern analysis. However, our work goes one step further by unifying the three flows of the three perspectives. DYNO (Scholten et al. 2011) is another interesting notation to help platform providers in creating services. It is a specialised model for the platform instead of a universal one and it cannot satisfy the requirements of other participants.

8.3. Enterprise business model

Similar to the BP in information research, EB models and their economic analysis abound in the line of management research. Most noticeably, Amit and Zott (2001), by defining how firms manage their transactions with shareholders such as customers, partners, investors, and suppliers, laid foundation of business model research. Morris, Schindehutte, and Allen (2005) pinpoint that the business model is constructed in the field of business strategy. Niu, Xu, and Bi (2013), Niu et al. (2013) discuss the recent developments on enterprise information systems. In general, the similarities and difference between the business model and the EP can be outlined as follows:

- Both the business model and the EP are motivated to figure out how the stakeholder makes profit by acting as a specific participant and taking part in activities in the BP. The essential destination of research in this area is to assist modifying or redesigning the business model to get a better benefit.
- Both business models and EPs focused on the participant. These models give some suggestions on RA for a better benefit.
- While the business model has its root in economics, the EP is in essence a computer model. While the former anchors its interests in economic effect, the latter focuses on the relationship between the BP and value creation.

Innovatively, Gordijn (2002), Gordijn and Akkermans (2001) propose a famous e-business modelling tool, e3-value, the first attempt to integrate information with business in one model. E3-value, by focusing on the value flow of e-business, achieves great success and lays the foundation for our work. Building on Gordijn, our work strives to unify more perspectives than his e3-value by incorporating the control flow perspective and the data perspective. And we believe that the information about the data (the attributes of the entity) can play an extremely important role in explaining the value creation and promotion simultaneously.

9. Conclusion

Our work addresses the problems of modelling the MSC and we attempt to build a unified enterprise model of three different perspectives (control flow perspective, data perspective and the resource perspective). We define the EP as the combination of four kinds of strategies: *RA*, *AO*, *SC* and *PD*. In this article, we propose the initial results of our project, a declaration language named EPDL consisting of four basic language elements extracted from strategies (*resource* from RA, *activity* from AO, *participant* from SC and *entity* from PD). Furthermore, we extend BPMN to BPMN4EP with the four basic elements considered to support EPDL. The complete example of mobile application software enterprise is studied in detail. Our 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 EP.

Acknowledgment

We would like to thank Associate Professor Na Luo for her scrupulous polishing of the language and her insightful comments on the structure of this manuscript.

Disclosure statement

No potential conflict of interest was reported by the authors.

Funding

This work was supported by National Technology Support Program under grant no. 2013BAH10F02.

References

Amit, R., and C. Zott. 2001. "Value Creation in E-Business." *Strategic Management Journal* 22 (6–7): 493–520. doi:10.1002/(ISSN)1097-0266.

Appel, S., S. Frischbier, T. Freudenreich, and A. Buchmann. 2013. "Event Stream Processing Units in Business Processes." In *Business Process Management*, 187–202. Berlin: Springer.

Barreto, C., V. Bullard, T. Erl, J. Evdemon, D. Jordan, K. Kand, D. Konig, et al. 2007. "Web services business process execution language version 2.0 primer." In *OASIS Web Services Business Process Execution Language (WSBPEL) TC, OASIS Open*.

Bhattacharya, K., C. Gerede, R. Hull, R. Liu, and J. Su. 2007. "Towards Formal Analysis of Artifact-Centric Business Process Models." *Business Process Management*, 288–304. Berlin: Springer.

Calvanese, D., M. Montali, M. Estanol, and E. Teniente. 2014. "Verifiable UML Artifact-Centric Business Process Models." Proceedings of the 23rd ACM International Conference on Information and Knowledge Management, Shanghai, November 3–7.

Cardoso, J., A. Barros, N. May, and U. Kylau. 2010. "Towards a Unified Service Description Language for the Internet of Services: Requirements and First Developments." In *IEEE International Conference on Services Computing (SCC)*, Miami, FL, July 5–10, 602–609. IEEE.

Casado, R., J. Tuya, and M. Younas. 2011. "An Abstract Transaction Model for Testing the Web Services Transactions." In *International Conference on Web Service*, Washington, DC, July 4–9, 730–731. IEEE.

Cohn, D., and R. Hull. 2009. "Business Artifacts: A Data-Centric Approach to Modeling Business Operations and Processes." *Bulletin of the IEEE Computer Society Technical Committee on Data Engineering* 32 (3): 3–9.

Duan, Y., A. Kattepur, and D. Wencai. 2013. "Service Value Broker Patterns: Integrating Business Modeling and Economic Analysis with Knowledge Management." In *International Conference on Web Services (ICWS)*, Honolulu, HI, June 24–29, 615–616. IEEE.

Gordijn, J., 2002. "E 3-Valuein a Nutshell." Proceedings of International Workshop on E-business Modeling, Lausanne.

Gordijn, J., and H. Akkermans. 2001. "Designing and Evaluating E-Business Models." *IEEE Intelligent Systems* 16 (4): 11–17. doi:10.1109/5254.941353.

Hull, R., 2008. "Artifact-Centric Business Process Models: Brief Survey of Research Results and Challenges." In *On the Move to Meaningful Internet Systems: OTM 2008*, 1152–1163. Berlin: Springer.

Kiepuszewski, B., A. H. M. Hofstede, and W. M. P. Aalst. 2003. "Fundamentals of Control Flow in Workflows." *Acta Informatica* 39 (3): 143–209. doi:10.1007/s00236-002-0105-4.

Lee, C. H., S. Y. Hwang, and I. L. Yen. 2012. "A Service Pattern Model for Flexible Service Composition." In *International conference on Web Services (ICWS)*, Honolulu, HI, June 24–29, 626–627. IEEE.

Liang, Q. A., J.-Y. Chung, S. Miller, and Y. Ouyang, 2006. "Service Pattern Discovery of Web Service Mining in Web Service Registry-Repository." In *IEEE International Conference on, e-Business Engineering, 2006. ICEBE '06*. IEEE International Conference on Shanghai: IEEE, 286–293.

Meyer, A., L. Pufahl, D. Fahland, and M. Weske. 2013. "Modeling and Enacting Complex Data Dependencies in Business Processes." *Business Process Management*, 171–186. Springer.

Morris, M., M. Schindehutte, and J. Allen. 2005. "The Entrepreneurs Business Model: Toward a Unified Perspective." *Journal of Business Research* 58 (6): 726–735. doi:10.1016/j.jbusres.2003.11.001.

Niu, N., L. D. Xu, and Z. Bi. 2013. "Enterprise Information Systems Architecture—Analysis and Evaluation." *IEEE Transactions on Industrial Informatics* 9 (4): 2147–2154. doi:10.1109/TII.2013.2238948.
Niu, N., L. D. Xu, J. R. Cheng, and Z. Niu. 2013. "Analysis of Architecturally Significant Requirements for Enterprise Systems." *Systems Journal* 850–857. doi:10.1109/JSYST.2013.2249892.
Reijers, H. A., T. Slaats, and C. Stahl. 2013. "Declarative Modeling -An Academic Dream or the Future for BPM?" In *Business Process Management*, 307–322. Berlin: Springer.
Rodríguez, A., A. Caro, C. Cappiello, and I. Caballero. 2012. *A BPMN Extension for Including Data Quality Requirements in Business Process Modeling*, 116–125. Berlin: Springer.
Ruiz, F., F. Garcia, L. Calahorra, C. Llorente, L. Gonçalves, C. Daniel, and B. Blobel. 2012. "Business Process Modeling in Healthcare." *Studies in Health Technology and Informatics* 179: 75–87.
Russell, N., and W. M. Van Der Aalst. 2007. *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, 513.
Scholten, U., R. Fischer, C. Zirpins, C., and S. Scholten. 2011. "DYNO: A Notation to Leverage Dynamic Network Effects in Paas Ecosystems." In *IEEE international conference on Service-Oriented Computing and Applications (SOCA)*, Irvine, CA, December 1–6. IEEE.
Tziviskou, C., M. Palmonari, M. Comerio, and F. De Paoli. 2013. "Sedl-c: A Language for Modeling Business Terms in Service Descriptions." IEEE 20th International Conference on Web Services (ICWS), Santa Clara Marriott, CA, June 27–July 2, 547–554. IEEE.
Vaculín, R., T. Heath, and R. Hull. 2012. "Data-Centric Web Services Based on Business Artifacts". IEEE 19th International Conference on Web Services (ICWS), Honolulu, HI, June 24–29, 42–49. IEEE.
Van Der Aalst, W., and K. M. Van Hee. 2004. *Workflow Management: Models Methods, and Systems*. Cambriage, MA: MIT press.
Van Der Aalst, W. M. P. 2005. "YAWL: Yet Another Workflow Language." *Information Systems* 30 (4): 245–275. doi:10.1016/j.is.2004.02.002.
Viriyasitavat, W., and L. Da Xu. 2014. "Compliance Checking for Requirement-Oriented Service Workflow Interoperations." *IEEE Transactions on Industrial Informatics* 10 (2): 1469–1477. doi:10.1109/TII.2014.2301132.
Viriyasitavat, W., L. Da Xu, and A. Martin. 2012. "Swspec: The Requirements Specification Language in Service Workflow Environments." *IEEE Transactions on Industrial Informatics* 8 (3): 631–638. doi:10.1109/TII.2011.2182519.
Wang, C., and L. Xu. 2008. "Parameter Mapping and Data Transformation for Engineering Application Integration." *Information Systems Frontiers* 10 (5): 589–600. doi:10.1007/s10796-008-9112-5.
White, S. A. 2004. "Process Modeling Notations and Workflow Patterns." *Workflow Handbook* 2004: 265–294.
Wong, P. Y., and J. Gibbons. 2008. *A Process Semantics for BPMN. In Formal Methods and Software Engineering*, 355–374. Berlin: Springer.
Wu, Z. H., X. B. Wu, and M. M. Yao. 2013. *Business Model Innovation of Modern Service Company a Value Network Perspective*. Beijing: Science Press.
Xu, L. 2011. "Enterprise Systems: State-Of-The-Art and Future Trends." *IEEE Transactions on Industrial Informatics* 7 (4): 630–640. doi:10.1109/TII.2011.2167156.
Xu, L., H. Liu, S. Wang, and K. Wang. 2009. "Modelling and Analysis Techniques for Cross-Organizational Workflow Systems." *Systems Research and Behavioral Science* 26 (3): 367–389. doi:10.1002/sres.v26:3.
Xu, L., W. Tan, H. Zhen, and W. Shen. 2008. "An Approach to Enterprise Process Dynamic Modeling Supporting Enterprise Process Evolution." *Information Systems Frontiers* 10 (5): 611–624. doi:10.1007/s10796-008-9114-3.
Xu, L., and W. Viriyasitavat. 2014. "A Novel Architecture for Requirement-Oriented Participation Decision in Service Workflows." *IEEE Transactions on Industrial Informatics* 10 (2): 1478–1485. doi:10.1109/TII.2014.2301378.
Xu, L., W. Viriyasitavat, P. Ruchikachorn, and A. Martin. 2012. "Using Propositional Logic for Requirements Verification of Service Workflow." *IEEE Transactions on Industrial Informatics* 8 (3): 639–646. doi:10.1109/TII.2012.2187908.