

An Empirical Method to Derive Principles, Categories, and Evaluation Criteria of Differentiated Services in an Enterprise

Vikas S Shah¹

¹Wipro Technologies, Connected Enterprise Services
East Brunswick, NJ 08816, USA
vikas.shah@wipro.com

Abstract

Enterprises are leveraging the flexibilities as well as consistencies offered by the traditional service oriented architecture (SOA). The primary reason to imply SOA is its ability to standardize way for formulating separation of concerns and combining them to meet the requirements of business processes (BPs). Many accredited research efforts have proven the advantages to separate the concerns in the aspects of one or more functional architectures such as application, data, platform, and infrastructure. However, there is not much attention to streamline the approach when differentiating composite services derived utilizing granular services identified for functional architectures. The purpose of this effort is to provide an empirical method to rationalize differentiated services (DSs) in an enterprise. The preliminary contribution is to provide abstract principles and categories of DS compositions. Furthermore, the paper represents an approach to evaluate velocity of an enterprise and corresponding index formulation to continuously monitor the maintainability of DSs.

Keywords: *Business Process (BP) Activities, Differentiated Services (DSs), Enterprise Entities, Maintainability, Requirements, and Velocity of an Enterprise.*

1. Introduction

Traditionally, services of SOA are composited to associate enterprise entities and corresponding operations to business process (BP) activities. The concept of DSs is fairly novel that introduces level of variations necessary to accommodate all the potential scenarios that are required to be included within the diversified business processes [4] and [7]. DSs are the services with similar functional characteristics, but with additional capabilities, different service quality, different interaction paths, or with different outcomes [5]. DSs provide the ability to capture precise interconnectivity and subsequently the integration between BPs and the operations of enterprise entities [12].

Typically, BP association with enterprise entities begins with assessments of the goals and objectives of the events required to accomplish the BP requirements. After modeling, BPs are implemented and consequently deployed to the platform of choice in an enterprise. The DSs have the built-in ability to considerably amend the

BPs' associations to their activities and reorganize based on either changes to the or new BP requirements [5] and [19]. It allows accommodating the desire level of alterations and respective association in the BPs across enterprise by means of combining capabilities of more granular services or nested operations.

DSs deliver the framework to place and update BPs as well as other important capabilities of monitoring and managing an enterprise. It enterprises accelerated time-to-market, increased productivity and quality, reduced risk and project costs, and improved visibility. Enterprises often underestimate the amount of change required to adapt the concept of DSs. [15], [16], and [17] indicates that DSs are usually architected, updated, and built based on ongoing changes into the enterprise. For example, newly introduced product of digital electric meter will be added to the product database and the service to “*capture the meter data remotely*” gets updated explicitly and in composition with data service to formalize the capabilities of the new product. The primary concerns such as update to the data service and the behavior of digital electric meter during the outage are not being addressed or realized during later stages when the specific event occurs pertaining to the specific BP.

Consequently, the entire purpose of DSs and their association with the enterprise entities are misled. It indicates that through feasibility analysis and navigation of complex cross functional changes of BPs associated with the enterprise entities are essential before updating DSs. The analysis presented in this paper identifies core characteristics of DSs and their association to the modeled BPs of an enterprise. The paper presents an approach to rationalize the relationship between the DSs and the desired variability in BP activities. The goal is to streamline and evaluate association between the BP requirements and baseline criteria to incorporate them into DSs. It sets the principles, categories, and evaluation criteria for DSs to retain the contexts and characteristics of DSs in an enterprise during various levels of updates.

In section 2, the primary concerns of the DSs and corresponding review of the past research efforts are presented. Section 3 provides methodology to institute DSs in an enterprise and derives preliminary principles. Identified meta-level categories of DSs are enumerated in Section 4. The classification of DSs is based on characteristics as well as anticipated behavior of the DSs. Section 5 represents the evaluation method for velocity of change in an enterprise considering 7 different BPs. Section 6 proposes and derives practical criteria to indicate maintainability of DSs depending on their classification. Section 7 presents conclusion and future work.

2. Literature Reviews and Primary Concerns of Introducing DSs

BPs assist businesses to make decisions in order to manage the enterprise. Using a combination of a BP activities, associated metrics, and benchmarks, organizations can identify enterprise entities that are most in need of improvement. There has been an increasing adaptation of BPs to derive granular level principles for an enterprise in recent years [2], [18], [22] and [29]. The Open Group Architectural Framework (TOGAF) [31] reserves business architecture as one of the initial phase to define BPs. The Supply Chain Council's Supply Chain Operations Reference-model (SCOR), the Tele-Management Forum's Enhanced Telecom Operations Map (eTOM), and the Value Chain Group's Value Reference Model (VRM) framework are the prominent examples of specifying BPs.

However, widely accepted enterprise architecture (EA) and other frameworks [27] and [2] aren't addressing the complexities of implementing desired variability in BPs and corresponding BP activities. They are highly deficient in specifying synergies of the DSs to BPs in an enterprise. BP management suite providers are also offering either inherent SOA and EA capabilities or third-party integration adapters [8]. As specified in [3], [6], and [11], it is primarily to eliminate the friction between BPM, anticipated variations in services, and enterprise architecture modeling. The most prevalent examples are Oracle SOA suite [24], Red Hat JBOSS BPM and Fuse products, OpenText BPM suite, IBM BPM suite [21], and Tibco Software as indicated in [8]. BP management suites are still struggling to achieve their enterprise potential best practices to implement and update DSs.

The BP requirements are usually grouped to formulate the future state of an enterprise. These requirements drives the vision and guides the decisions to introduce DSs. Various different research efforts [20], [23], and [33] indicates that

the decisions are based on some or other way related to the following criteria.

- **Existing product or service offerings** and their enhancements, support, and maintenance. For example, DSs associated with the online payment BP has to consider the product subscribed or in use by the customer.
- **New products or services** that will enhance revenue or gain new market share in the current or near term timeframe. The most prominent DS example is to replace electric meter with the smart meter for specific set of customers.
- **Innovation related to future trends and competition.** Product and service offerings that require immediate development, however, will not contribute to revenue until outlying years. DSs deployed to prospect search and survey to investigate interest in advanced smart grid products are the examples.
- **Exit strategies for existing product or service offerings.** Proactively determining end life of the products or services. In many cases, the previous products and services are either need to be discontinued or advanced significantly. The foremost example is videocassette recorder.

The result of the decision process is a set of principles and key value propositions that provides differentiation and competitive advantages. Various attempts have been made either in specific use case [34] or in abstract standardization [32] and [25]. Rationalized principles have a much longer life span. These principles are direct or indirect reflection to attend the uncertainties of an enterprise. The principles should consider all the levels as well as categories of uncertainties identified or evaluated during the BP activities. In [14], three types of uncertainties are illustrated with examples.

- **State uncertainty** relates to the unpredictability that represents whether or when a certain change may occur. The example of state uncertainty is the initiation of outage process (by the utility corporation providing the outage to restoration services).
- **Effect uncertainty** relates to the inability to predict the nature of the impact of a change. During the outage due to unforeseen weather condition, it is absolutely unpredictable to know the locations or areas of impact.
- **Response uncertainty** is defined as a lack of knowledge of response options and/or an inability to predict the consequences of a response choice. Generally, utility provider has guideline for restoration during the outages, however, it is unpredictable during the situations that are never been faced before, such as undermined breaks in the circuits.

DS needs to implement these uncertainties either proactively initiating a change or reactively responding to the change. The conclusion of various studies [9], [10], [18], and [22] indicates that first step to consistently implement and update DSs is to define principles. These principles govern maintaining DSs in the correlations with the enterprise entities and advancements of BP activities.

3. Deriving Principles of DSs

The analysis of primary concerns and literature reviews illustrated in Section 2 justifies that the method for deriving principles of DSs should fundamentally have a focus at the BP requirements, identified and placed BP activities, and interdependencies between events of BP activities. The BP requirements have to be reviewed to certify the legitimacy and candidature for diversification to form DSs' specification. Figure 1 presents a sequence of steps performed to identify principles of DSs in an enterprise and architect DSs in adherence to BP requirements.

BP Requirements and Initiation: The first step is to validate BP requirements alignment with business and goals of an enterprise. *Stage 0* (initiation) is defined to reiterate and evaluate BP requirements at each phase (or step). When there is an ambiguity identified in the BP requirement at any step due to responsibilities associated with the corresponding step then Stage 0 has been initiated. *Stage ACN* is defined to analyze business impact, conflict of interest (if any exists), and notification across enterprise.

Discovering and Assessing Architecture Artifacts: When an enterprise receives alterations or new BP requirements, it needs to assess the impact in terms of other architectures associated with an enterprise (BP architecture, integration architecture, and system architecture). The responsibility of this step is to identify the need of introducing or updating architecture artifacts based on the process map (that is, association of services to the BPs or their activities). Primarily, it is accountable to identify whether any sublevel BPs (within existing BPs) and any additional BP activities required to be introduced. The need of introducing additional sublevel BPs or BP activities may be either due to critical to major advancements in BP requirements or changes necessary to other associated architecture artifacts (including integration and system architectures).

The other major responsibility of this step is to check availability of services for diversification based on BP requirements. It is also liable for specifying the desired

level of updates and interdependencies with enterprise entities associated with the services (DSs or other types).

Defining and Evolving Service Architecture: It is the primary step to define, update, version, and deploy DSs. The DS gets evolved and advanced accommodating the desired level of diversification identified in previous step. The responsibilities of this step also include evaluating the potential uncertainties and alternate path that needs to be derived in adherence to identified uncertainties.

The decision whether to introduce additional DS, additional operation to existing DSs, or changes to the operations of existing DSs has to be achieved during this step. Modeling to map DSs with BP activities and streamlining their implementation are the part of this phase of DSs enabled enterprise.

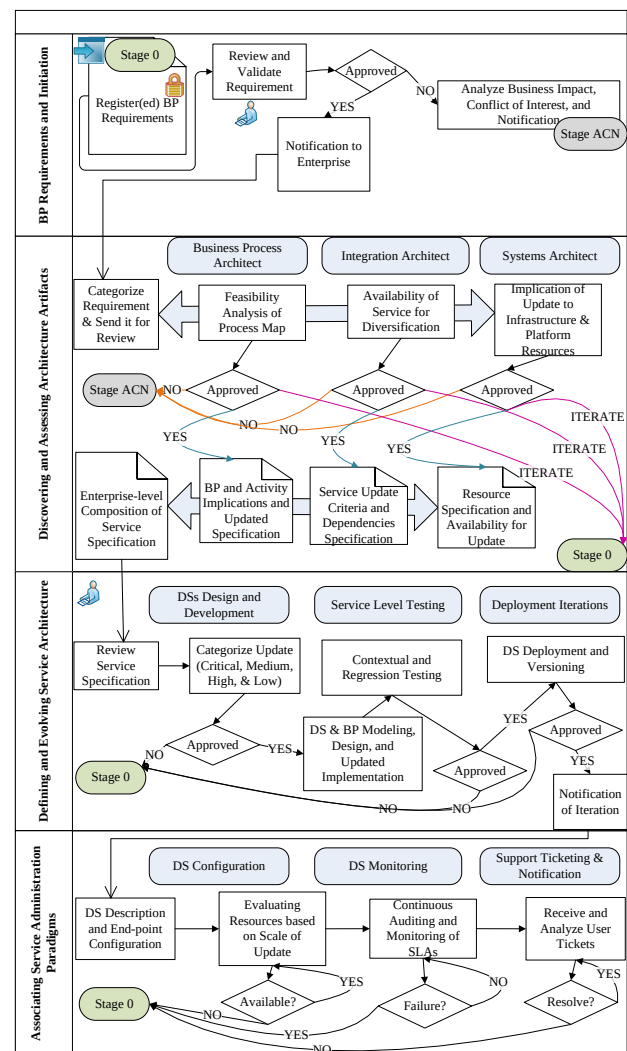


Fig. 1 Steps to identify principles of DSs and architect DSs in an enterprise.

Associating Service Administration Paradigms:

Specifying and resolving interdependencies of DSs with participant enterprise entities are the responsibilities of this step. It needs to ensure that DSs are in adherence to the availability of the enterprise resources and their defined Service Level Agreements (SLAs). Configuration, monitoring, and supporting DSs in association with enterprise entities (including any failure condition or resolution to uncertainties) are also the accountability of this step to derive principles of DSs in an enterprise and provide informed architecture decisions for DSs.

Following are the principles derived to identify, specify, develop, and deploy DSs in an enterprise based on the steps necessary to achieve BP requirements. Each step identified in Figure 2 reveals and constitutes the foundation for deriving the principles of DSs in relationship with BP requirements.

- Specification of DS's operation into information that can be utilized in BPs in the context of concrete activities. The most prominent example is BP activity "*generate invoice*" needs DS that retrieves and combines the information of *purchased products* and their *current pricing*.
- Deterministic specification of relationship between BP activities and enterprise entities in DS. In the example of BP activities *generate invoice*, if any discount has to be implied then it needs to be in correlations with the pricing of the product.
- Precisely define BP activity's events that can be emulated, monitored, and optimized through DS. The BP activity "*generate invoice*" request requires to be validated before retrieving the other related information.
- Impact of people, processes, and product (or service) offerings as metadata associated with the DS. The BP activity "*generate invoice*" can only be initiated by specific role associated with the employee (example: manager) or triggered by another activity such as "*completed order*".
- Specify and govern SLAs of DS in the context of associated BP activity. The invoice should be generated within 3 seconds of completing order is an example of SLA.
- Regularly place and evaluate governance paradigms for DS in association with BP activity to address uncertainties. The BP activity "*cancel order*" or "*returning an item (or product)*" can occur after invoice has been generated. If those activities are not defined and *updating, canceling or revising invoicing* capabilities are not defined then it needs to be introduced.

4. Identified Categories of DSs

Due to increasing availability and development of SOA and BPs [26] and [28] platforms, services are being characterized in numerous different aspects. The foremost utilized classification methodology is functional architecture types such as platform services, data services, application services, and infrastructure services. Another approach is to classify industry segment specific services such as healthcare services, utility services, and payment services. Certain enterprises are also inclined to introduce custom classification of the services due to unavailability of the standards as well as rationalization.

Identified principles of DSs indicate that DSs are required reacting to the set of events associated with BP activities. DSs are independently built or composited utilizing one or more types of services placed in an enterprise. DSs need to be categorized such that each type can be streamlined based on their characteristic and governed based on the type of SLAs associated with them. Following is the list of identified categories of DSs based on their characteristics.

Competency Services: DSs that participates to satisfy one or more competencies of the core business offerings are categorized as competency services. Certain features between different versions of the same product-line are generic and essential, however, some features need to be distinguished in the DS.

Relationship Services: DSs presenting external and internal relationships of the enterprise entities with the role associated with the entities such as customer, partner, and supplier. The example of such DS is the relationship of order with customer differs from the vendor and corresponding action needs to differ in the operations of DS.

Collaboration Services: Any DS offering collaboration among varied enterprise entities and BP activities are considered the participant of collaborative service category. Calendar request to schedule the product review meeting is the type of collaborative service where participants can be either reviewer, moderator, or optional.

Common Services: When an enterprise gain maturity, it needs to have standardized audit, log, and monitor capabilities. These standardized DSs falls in the category of common services. They are built to utilize consistently across multiple sets of BP activities with specific objective to monitor. *Generating invoice* and *amount paid* for an order are different BP activities, however, the number of item purchased are same and they are required to be monitored as well as verified between BP activities.

Framework Services: The framework services are to increase awareness of the enterprise's technology architecture capabilities. DS built to search metadata associated with application services, data services, platform services, or infrastructure services is an example of framework service. The DSs differs in terms of what type of metadata can be searched for which kind of service.

Governance Services: DSs deployed to ensure the policies and practices are the governance services. Most diversification to the security related services including role based entitlement are the participant of governance services.

Organizational Services: Organization culture has various impacts on the BP activities. DSs that offer common understanding of organization culture as well as corporate processes are the organizational services. Ordering and utilizing office supplies for different departments is an example of organizational service. In this example, DS differs in terms of accessibility of type of supplies to the particular department.

Strategic Services: DSs participates in making a decision that impacts strategic direction and corporate goals are categorized as strategic services. Financial analysis based selection of marketing segments and budgeting based on available statistics of annual spending are the types of strategic services.

Conditional Services: Certain BP activities require special attention and business logic dedicated to particular condition. The DSs built, updated, and maintained to accommodate such scenarios are subject to this classification. Credit card with special privilege for purchases over allocated limit is an example of such DSs.

Automation Services: They are the services defined and utilized to introduce desired level of automation, yielding additional business value for new or existing BP activities. Typically, automation related services require stronger bonding and maturity at the BP activities. Service to send email notification for the approval versus the service for online approval is the classical example of such DSs.

DSs can be associated with multiple categories. However, alias to the DS is utilized for the secondary category such that it can be independently monitored and audited. Optional DSs' common header elements (or metadata) are introduced to capture the runtime metrics for the DSs. Following are the additional information that DSs' provides at runtime for further evaluation.

- Instance identification of the DS.
- Category of the DS.
- BP name and activity utilizing the DS.
- Registered consumer group and associated role using the DS.
- Service's probability of failure (recursively identified from the audit logs).

5. Evaluating Velocity of an Enterprise

The experimental evaluation is based on set of 62 DSs out of 304 services (includes functional architecture type services as well as industry segment specific services besides dedicated DSs). The services are built in Oracle SOA suite [24] that has internal capabilities to map and generate relationship with BP activities. 4 iterations of the development, updates, and deployment have been conducted for the following 7 BPs. The BP activities and DSs are derived based on severity of the BP requirements.

BP# 1: Customer enrollment and registration

BP# 2: Manage customer information, inquiry, and history

BP# 3: Purchase order

BP# 4: Payment processing and account receivables

BP# 5: Invoicing

BP# 6: Notification and acceptance of terms

BP# 7: Account management

Velocity of the enterprise is representation of the rapid changes and updates necessary to achieve the BP requirements. The changes can be achieved through updating or introducing either DS operations, DSs, BP activities, or sublevel BPs. Correspondingly, the velocity is based on four types of ratios as specified bellow. The ratios are representation of the level of change necessary to achieve goals of BP requirement.

- DSs' Ratio (**DSR**) = (Additional composite service / Total number of services)
- DS Operations' Ratio (**OPR**) = (Additional accumulative number of DSs operations / Accumulative number of DSs operations)
- BP Activities' Ratio (**AR**) = (Additional BP activities / Total number of BP activities)
- Sublevel BPs' Ratio (**SBR**) = (Additional sublevel BPs / Total number of sublevel BPs)

The velocity evaluation presented in Eq (1) also introduces impact factor corresponding to each ratio, that is, *c* (critical), *h* (high), *m* (medium), and *l* (low). The assigned values for the impact factors are *c* = 10, *h* = 7, *m* = 4, and *l* = 2 to indicate finite value for the severity of update. There is absolutely no constraint to revisit the allocation of

severity to update impact factors during subsequent iterations of updates to BP requirements and corresponding deployment cycle. It should be based on findings as well as severity of BP requirements in consideration.

In Eq (1), #BPs represents total number of participant BPs to form DSs enabled enterprise (7 in this case). When there is a need to introduce or update sublevel BP due to BP requirement then it is considered *critical* (c) change to an enterprise. Whereas, update to or introduction of DS operation is considered lowest category of change, that is, *low* (l).

VELOCITY =

$$\frac{\sum_{BP=1}^{\#BPs} m \times DSR + l \times OPR + h \times AR + c \times SBR}{\#BPs} \quad (1)$$

Table 1 provides implementation based analysis and computed velocity of 4th deployment iteration of BP requirements corresponding to the 7 BPs (as described above). Following are the acronyms utilized in Table 1.

- **#DS:** total number of participant DSs for the BP.
- **#OPs:** accumulative number of DSs' operations involved.
- **#As:** total number of BP activities for the BP.
- **#SBPs:** total number of sublevel BPs of the BP.
- **#A-CS:** sum of new and updated DSs to the BP in iteration 4.
- **#A-OPs:** sum of new and updated number of DSs' operations introduced to the BP in iteration 4.
- **#A-A:** sum of new and updated BP activities introduced to the BP in iteration 4.
- **#A-SBPs:** sum of new and updated sublevel BPs introduced to the BP in iteration 4.

DSs' operations, DSs, BP activities, and sublevel BPs that are being reused across multiple BPs are counted at each and every instance for the purpose of accuracy to evaluate velocity.

Table 1: Velocity of the enterprise in iteration 4

BP#	#DSs (#A-CSs)	# OPs (#A-OPs)	#As (#A-A)	# SBPs (#A-SBPs)
1	7(0)	20(2)	8(1)	2(0)
2	12(3)	28(7)	15(0)	3(0)
3	18(4)	42(7)	22(2)	5(1)
4	8(2)	15(3)	15(2)	3(0)
5	5(1)	12(2)	10(1)	3(0)
6	3(0)	8(1)	7(0)	1(0)
7	9(2)	16(3)	14(1)	2(0)
VELOCITY (of Iteration 4) = 1.52				

As such there is no maximum limit set for the velocity, however, present deployment iteration's velocity score can be considered as the baseline for subsequent iterations. The progressive values of velocity are indicated in Figure 2 for each iteration (1 through 4) pertaining to the 7 BPs in consideration.

6. Formulating DSs Maintainability Index (DSMI)

There is no obvious solution to evaluate maintainability of DSs. The primary reason is due to the little to no effort for defining maturity model and standardization for DSs. SOA maturity models and governance are implied at more operational aspects of the functional architecture type services [30] and [13]. The other types of metrics presented in [1] and [32] to measure the agility irrespective of the maintainability concerns of DSs. The DSMI is an effort to compute and continuously monitor maintainability of DSs. Oracle SOA suite capabilities are utilized to monitor and log DSs. Service registry features are embraced to define, govern, and monitor SLAs as well as metadata associated with the DSs.

6.1 Paradigms to Derive Inverted DSMI

The paradigms to formulate DSMI are described below for each type of DSs.

Business continuity (BUC): It is to determine whether the introduced or updated DSs are able to continue the day-to-day business activities after the deployment (or iteration). The evaluation criterion for BUC paradigm is to monitor the number of unique support tickets created for type of DSs in context. For example, new customer registration is providing errors due to inaccuracies in validation of customer account number and/or customer identification.

The inverted ratio for BUC specific to the set of DSs associated with the DS type is derived below.

$iBUC_{<DS\ type>} = (\# \text{ of unique support tickets by the customer} / \#DSs \text{ deployed for } <DS \text{ type}>)$

Operational risk (ORI): Operational risks are basically to evaluate the DS level continuation of the enterprise operations. Typically, it is traced by the number of failures occurred for the DSs in the production cycle of present deployment iteration. The specific example of change purchase order request DS failed due to unambiguous condition occurred within the dedicated DSs. The inverted ratio for ORI specific to the set of DSs associated with the DS type is derived below.

$iORI_{<DS\ type>} = (\# \text{ of unique operational failures} / (\#DSs \text{ deployed for } <DS\ type>))$

The ratio of oRIS is being generated by comparing the failures with previous deployment iteration. The DSs header contains probability of failures and it is being automated at some extend to gain indicative operational risk at runtime (as stated in Section 4).

SLA Factorization (SPR): Scalability, reliability, and performance (SPR) are being bundled to evaluate SLA factorization. The SLAs defined in consideration of desired SPR for each type of DSs are configured and monitored. The SPR is identified based on the number of violations by the particular category of DSs in the present deployment iteration. The 4 seconds delay (when SLA is set for maximum 3 seconds) in sending order confirmation to vendor for specific product due to the heavy traffic is an example of SLA violation. The inverted ratio for SPR specific to the set of DSs associated with the DS type is derived below.

$iSPR_{<DS\ type>} = (\# \text{ of unique SPR specific SLA violations} / (\#DSs \text{ deployed for } <DS\ type>))$

Consistency (COS): Consistency can be evaluated at many different aspects. The primary objective of this criterion is to assess scope of the DS across multiple BP activities. Due to the BPs requirements, specification of the DS needs to incorporate high level interactions with enterprise entities and underneath events of BP activities. The consistency of DS is being derived based on the number of BP activities utilizing the specific type of DSs in considerations. The most prominent example is order delivery confirmation and status needs to be sent to customer, vendor, and account receivables. The inverted ratio for COS specific to the set of DSs associated with the DS type is derived below.

$iCOS_{<DS\ type>} = (\# \text{ of BP activities utilizing DSs of } <DS\ Type> / \#DSs \text{ deployed for } <DS\ type>)$

Extendibility and Continuous Improvements (ECI): Extensibility and continuous improvement of the DSs are evaluated based on customization required to accomplish BP requirements. It is computed considering the number of additional custom modeling as well as implementation needed in context of BP activity and enterprise entity. The primary objective is, whether respective DSs are able to accommodate these customizations within the dilemma of their dependencies with existing enterprise entities. If the payment is not received within 6 months then it needs to be sent for collection and vendor also needs to be notified, is

an example of extendibility of DSs associated with payment processing and account receivable BP. The inverted ratio for ECI specific to the set of DSs associated with the DS type is derived below.

$iECI_{<DS\ type>} = (\# \text{ of alternate BP flows accustomed in DSs of } <DS\ Type> / \#DSs \text{ deployed for } <DS\ type>)$

If “n” stands for the number of DS types identified in an enterprise (it is 10 in this case based on Section 4) then inverted DSMI can be computed based on Eq. (2). #Paradigms (number of paradigms) to impact the DSMI is 5 as described above.

Inverted DSMI = (1 / DSMI) =

$$\frac{[(\sum_1^n iBUC) / n] + [(\sum_1^n iORI) / n] + [(\sum_1^n iSPR) / n] + [(\sum_1^n iCOS) / n] + [(\sum_1^n iECI) / n]}{\#Paradigms} \quad (2)$$

Table 1 below presents the DSMI computed in the iteration 4 for the identified and deployed 7 BPs (as described in Section 5).

Table 2: DSMI in iteration 4

Paradigm DS Type (# of DSs)	iBUC	iORI	iSPR	iCOS	iECI
Competency (6)	0.33	0.83	0.5	0.5	0.67
Relationship (12)	0.25	0.67	0.5	1.5	0.5
Collaboration (4)	0.25	0	0.25	0.5	0.25
Common (7)	0.29	0.14	0.42	2	0.29
Framework (8)	0.5	0.25	0.75	0.5	0.38
Governance (6)	0.33	0.5	0.33	1.5	0.83
Organizational (7)	0.29	0.14	0.42	0.71	0.86
Strategic (7)	0.14	0	0.14	1	0.42
Conditional (5)	0.6	0.8	0.4	0.4	0.2
Automation (3)	0.33	0.67	1.67	0.67	2
Actual DSMI (of Iteration 4) = 1.76					

6.2 Analysis and Observations of Evaluation

Figure 2 provides the progress of velocity and DSMI through iteration 4 for the 7 BPs deployed, advanced, and monitored. The finite numbers indicate the significant reduction in velocity over the iterations. 58% reduction in velocity (of deployment iteration 4) compare to iteration 3.

The graph also indicates increase in DSMI over the iterations. The DSMI (of deployment iteration 4) is improved by 21% compare to iteration 3. The result directly illustrates that continuous monitoring and improvements in terms of reducing the number of issues reported by the business users, immediate resolutions to causes of services' failures, accurate modeling of DSs with respective to the BP requirements, and precisions in test scenarios decreases the velocity of enterprise and stabilizes the DSMI.

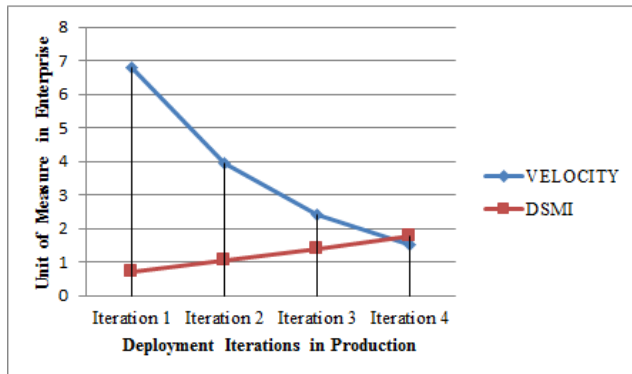


Fig. 2 Computed velocities and DSMI for all deployment iterations in production.

Essentially, it concludes that more number of BP activities utilizing single DS and more number of alternate path inclusion to single DS decreases the level of maintainability of DSs, however, it increases the consistency and extendibility of the DSs. Contrarily, introducing more number of DSs also increases additional level of SLAs' associations and uncertainties, however, introduces increased level of flexibility and agility in an enterprise. It is a trade-off that enterprise has to decide during the assessment of DSs architecture (2nd step described in Section 3 Figure 2).

7. Conclusions

The perception of SOA is receiving wide acceptance due to the ability of accustom and respond to BP related requirements and changes providing operational visibilities to an enterprise. DSs are the means to accommodate uncertainties of BPs such that an enterprise may able to gain acceptable level of agility and completeness. As such, there are limited to no standardization available to derive and maintain the qualities of DSs. In this paper, we presented necessity of rationalizing DSs and their principles. The research effort is to propose an empirical method to derive and evolve the principles of identifying

and placing DSs. The categorization and corresponding implementation for BP requirements into the DSs are identified and implied. Formulae to evaluate velocity of enterprise and assessment criteria to monitor maintainability of deployed DSs in terms of index are illustrated with an example implementation and validated in number of actual deployment iterations.

The rationalization achieved utilizing the methodology to derive and place principles of DSs increases consistency and predictability across multiple units as well as entities of an enterprise. The measurable implications due to changes in BP requirements and assessable maintainability are accomplished due to the classification and evaluation methodologies of DSs. The subsequent step is to determine more granular level of DSs types that can be leveraged in multifaceted BP scenarios. The underneath primary goal remains intact, that is, to evolve, retain, and stabilize maintainability of DSs.

Acknowledgments

Vikas Shah wishes to recognize Wipro Technologies' Connected Enterprise Services (CES) sales team to support the initiative. Special thanks to Wipro Technologies' Oracle practice for providing opportunity of implying conceptually identified differentiated principles, types, and measurements in 4 different iterations.

References

- [1] A. Qumer and B. Henderson-Sellers, "An Evaluation of the Degree of Agility in Six Agile Methods and its Applicability for Method Engineering," In: Information and Software Technology Volume 50 Issue 4, pp. 280 – 295, March 2008.
- [2] Alfred Zimmermann, Kurt Sandkuhl, Michael Pretz, Michael Falkenthal, Dierk Jugel, Matthias Wissotzki, "Towards an Integrated Service-Oriented Reference Enterprise Architecture," In: Proceedings of the 2013 International Workshop on Ecosystem Architectures, pp. 26-30, 2013.
- [3] Andrea Malsbender, Jens Poeppelbuss, Ralf Plattfaut, Björn Niehaves, and Jörg Becker, "How to Increase Service Productivity: A BPM Perspective," In: Proceedings of Pacific Asia Conference on Information Systems 2011, July 2011.
- [4] Anirban Ganguly, Roshanak Nilchiani, and John V. Farr, "Evaluating Agility in Corporate Enterprise," In: International Journal of Production Economics Volume 118 Issue 2, pp. 410 – 423, April 2009.
- [5] Aries Tao Tao and Jian Yang, "Context Aware Differentiated Services Development with Configurable Business Processes," In: 11th IEEE International Enterprise Distributed Object Computing Conference 2007, Oct 2007.
- [6] Anne Hiemstra, Pascal Ravesteyn, and Johan Versendaal, "An Alignment Model for Business Process Management and Service Oriented Architecture," In: th International Conference on Enterprise Systems, Accounting and Logistics (6th ICESAL '09), May 2009.

- [7] Bohdana Sherehiy, Waldemar Karwowski, and John K. Layer, "A Review of Enterprise Agility: Concepts, Frameworks, and Attributes," In: International Journal of Industrial Ergonomics 37, pp. 445 – 460, March 2007.
- [8] Clay Richardson and Derek Miers, "How The Top 10 Vendors Stack Up For Next-Generation BPM Suites," In: The Forrester Wave: BPM Suites, Q1 2013, For: Enterprise Architecture Professionals, March 2013.
- [9] Daniel Selman, "5 Principles of Agile Enterprise in 2012," In: IBM Operational Decision Manager Blog, Dec 2011.
- [10] Dean Leffingwell, Ryan Martens, and Mauricio Zamora, "Principles of Agile Architecture," Leffingwell, LLC . & Rally Software Development Corp., July 2008.
- [11] Douglas Paul Thiel, "Preserving IT Investments with BPM + SOA Coordination," Technology Investment Management Library, SenseAgility Group, November 2009.
- [12] Florian Wagner, Benjamin Klöpper, and Fuyuki Ishikawa, "Towards Robust Service Compositions in the Context of Functionally Diverse Services," In: Proceedings of the 21st international conference on World Wide Web, pp. 969-978, April 2012.
- [13] Fred A. Cummins, "Chapter 9: Agile Governance," In Book: Building the Agile Enterprise: With SOA, BPM and MBM, Morgan Kaufmann, July 28, 2010.
- [14] Haitham Abdel and Monem El-Ghareeb, "Aligning Service Oriented Architecture and Business Process Management Systems to Achieve Business Agility," Technical Paper, Department of Information System, Mansoura University, EGYPT, 2008.
- [15] Harry Sneed, Stephan Sneed, and Stefan Schedl, "Linking Legacy Services to the Business Process Model," In: 2012 IEEE 6th International Workshop on the Maintenance and Evolution of Service-Oriented and Cloud-Based Systems, August 2012.
- [16] Imran Sarwar Baiwa, Rafaqut Kazmi, Shahzad Mumtaz, M. Abbas Choudhary, and M. Shahid Naweed, "SOA and BPM Partnership: A paradigm for Dynamic and Flexible Process and I.T. Management," In: International Journal of Humanities and Social Sciences 3(3), pp.267-273, Jul 2009.
- [17] Imran Sarwar Bajwa, "SOA Embedded in BPM: High Level View of Object Oriented Paradigm," In: World Academy of Science, Engineering & Technology, Issue 54, pp.209-312, May 2011.
- [18] Jean-Pierre Vickoff, "Agile Enterprise Architecture PUMA," Teamlog, October 2007.
- [19] Leonardo Guerreiro Azevedo, Flávia Santoro, Fernanda Baião, Jairo Souza, Kate Revoredo, Vinícios Pereira, and Isolda Herlain, "A Method for Service Identification from Business Process Models in A SOA Approach," In: Enterprise, Business-Process and Information Systems Modeling, LNCS Volume 29, pp. 99-112, 2009.
- [20] Marinela Mircea, "Adapt Business Process to Service Oriented Environment to Achieve Business Agility," In: Journal of Applied Quantitative Methods . Winter 2010, Vol. 5 Issue 4, p679-691, 2010.
- [21] Martin Keen, Greg Ackerman, Islam Azaz, Manfred Haas, Richard Johnson, JeeWook Kim, Paul Robertson, "Patterns: SOA Foundation - Business Process Management Scenario," IBM WebSphere Software Redbook, Aug 2006.
- [22] Mendix Technology, "7 Principles of Agile Enterprises: Shaping today's Technology into Tomorrow's Innovation," Presentation, October 2013.
- [23] Michael Rosen, Boris Lublinsky, Kevin T. Smith, and Marc J. Balcer, "Overview of SOA Implementation Methodology," In Book: Applied SOA: SOA and Design Strategies, John Wiley & Sons, Web ISBN: 0-470223-65-0, June 2008.
- [24] Oracle Inc., Oracle Application Integration Architecture: Business Process Modeling and Analysis, Whitepaper, 2013.
- [25] Paul Harmon, "What is Business Architecture," In: Business Process Trends Vol. 8, Number. 19, November 2010.
- [26] Petcu, D. and Stankovski, V., "Towards Cloud-enabled Business Process Management Based on Patterns, Rules and Multiple Models," In: 2012 IEEE 10th International Symposium on Parallel and Distributed Processing with Applications (ISPA), July 2012.
- [27] Ralph Whittle, "Examining Capabilities as Architecture," In: Business Process Trends, September 2013.
- [28] Ravi Khadka, Amir Saeidi, Andrei Idu, Jurriaan Hage, Slinger Jansen, "Legacy to SOA Evolution: A Systematic Literature Review," Technical Report UU-CS-2012-006, Department of Information and Computing Sciences, Utrecht University, Utrecht, The Netherlands, March 2012.
- [29] Razmik Abnous, "Achieving Enterprise Process Agility through BPM and SOA," Whitepaper, Content Management EMC, June 2008.
- [30] Scott W Ambler and Mark Lines, "Disciplined Agile Delivery: A Practitioner's Guide to Agile Software Delivery in the Enterprise," IBM Press, ISBN: 0132810131, 2012.
- [31] The Open Group: TOGAF Version 9.1 Standards, December 2011.
<http://pubs.opengroup.org/architecture/togaf9-doc/arch/>
- [32] Tsz-Wai Lui and Gabriele Piccoli, "Degree of Agility: Implications for Information Systems Design and Firm Strategy," In Book: Agile Information Systems: Conceptualization, Construction, and Management, Routledge, Oct 19, 2006.
- [33] Vishal Dwivedi and Naveen Kulkarni, "A Model Driven Service Identification Approach for Process Centric Systems," In: 2008 IEEE Congress on Services Part II, pp.65-72, 2008.
- [34] Xie Zhengyu, Dong Baotian, and Wang Li, "Research of Service Granularity Base on SOA in Railway Information Sharing Platform," In: Proceedings of the 2009 International Symposium on Information Processing (ISIP'09), pp. 391-395, August 21-23, 2009.

Vikas S Shah received the Bachelor of Engineering degree in computer engineering from Conceicao Rodrigues College of Engineering, University of Mumbai, India in 1995, the M.Sc. degree in computer science from Worcester Polytechnic Institute, MA, USA in 1998. Currently he is Lead Architect in Connected Enterprise Services (CES) group at Wipro Technologies, NJ, USA. He has published several papers in integration architecture, real-time enterprises, architecture methodologies, and management approaches. He headed multiple enterprise architecture initiatives and research ranging from startups to consulting firms. Besides software architecture research and initiatives, he is extensively supporting pre-sales solutions, risk management methodologies, and service oriented architecture or cloud strategy assessment as well as planning for multinational customers.