Emerging Data Services Viscosity to Enrich Pivotal Capabilities of Interactive Enterprises: Towards a Lean Integration Architecture Approach

Vikas S. Shah

Abstract—**Primary focus of the lean architecture approaches is the enterprise value stream. They offer techniques to grip the consumers at the early stages of the integration architecture (IA) such that every subsequent artifact complements to IA. In recent years, interactive enterprise (IE) centric architecture integrates data and business processes by placing or updating reusable as well as orchestrated services of IA. The practices in data warehouse development have become ubiquitous, especially, as the need for system interoperability has grown and expectation has been increased to address susceptibility in association with enterprise-grade interactivity. More enterprises desire to architect application programming interface (API) landscape that consents to consolidating and associating segregated logical data sets to aspects of a responsive IE. In this paper, we provided an approach to disseminate and classify the emerging data services (DSs) that formulates an IE. The methodology to update these DSs enables IE to attend diversified characteristics of API context and traceability to the aspects of business processes (BPs). The lean IA principles and corresponding architecture trade-offs are derived to update and introduce. We present standardization to consistently evaluate these DSs within the characteristics of IE reportable across an enterprise.**

Index Terms—**Architecture trade-off analysis, business processes (BPs), data services (DSs), interactive enterprises (IEs), lean integration architecture (IA).**

I. INTRODUCTION

Advancements in interactive enterprises (IEs) involve real-time access to dynamic data sets due to the increasing necessity of rapid deployments of web enabled applications. In addition, businesses are inclined to gain the numerous advantages that are offered by exposing data and respective computations through services, often in the form of APIs. Presently, various SOA vendors are offering DSs platform [1] to enable an abstraction layer such that the anticipation in dynamicity of data pertaining to an interactive enterprise can be leveraged.

The capabilities of an IE is often affected by intangible, abstract goals that associate customer, business operations, organizational processes, and employees, and they are typically dependent on the actions of other parties [2] and [3]. DS modeling is a daunting task and usually error-prone process. DSs platform are open ended, however, deficient in standardization and in some cases they lacks to provide

generality in terms of implementation and diversification required for the interactive enterprises. Traditional approaches of modeling such data services that are being realized as integral capabilities of an interactive enterprise, where architects and designers are the providers and the business process owners, stakeholders, customers, vendors, suppliers, partners, and employees are either passive participant enterprise entities or consumers. They are least probable to result in accurate specification of DSs. BPs in an IE are complex considering numerous dependencies, unpredictability of the user scenarios and necessary updates required due to changing dynamics of the businesses. Another prime concern is the validity of the data in the context of utilization by the identified roles as well as diversity of its' temporal properties within an enterprise.

Our intense analysis and evaluation indicates that a set of basic lean IA principles provide a solid foundation for elaborated functionality built within the DSs. It enables a flexible architecture and avoids large monolithic systems. Implying lean IA approach leads focusing on core data dynamicity for an IE. It allows the rapid and rationalized method to advance the capabilities of an IE. A comprehensive DS modeling requires capturing three phenomena of an IE, that are, BP perspectives, interrelationships of the enterprise entities, and information flow, which should be integrated in the corresponding APIs.

Section II represents our gap analysis and approach in terms of value stream to derive, classify, and deploy DSs for an IE in the synergy of BPs and in alignment of API landscape. Section III provides the principles of updating these DSs to precisely formulate characteristics of data intensive IE. It illustrates the lean approach and architecture trade-offs during iteratively upgrading or introducing DSs. In Section IV, we have defined evaluation methodology in an effort to standardize the measuring of the characteristics of data intensive IE. The initial findings based on the viscosity of the deployed and upgraded DSs in the case of order management system (OMS) are also presented in Section IV. Section V represents the conclusion and future work to further strengthen the capabilities of data intensive IE.

II. POSITIONING DATA SERVICES IN AN INTERACTIVE ENTERPRISE

A consolidated set of value-added DSs can reduce operational costs and increase efficiency, all while improving the quality and utility of an IE. SOA is a business-centric agile architectural approach [4] that supports integrating business

data and processes by creating reusable components of functionality, specifically DSs. Traditionally, the identification of services has been done at a business function level. However, the business functions are highly dependent of either real-time or historical data to enable accuracy in informed decision. Due to inability of ubiquitous techniques of data warehouse and corresponding platforms [5], enterprises require reengineering their environments to allow sharing or even consolidation of business siloes in association with the data. There is a growing need for the underlying IA enabled DSs that provides ability to seamlessly access data from many different sources and delivers to various targets. DSs extend the value of BP services in SOA to elevate accuracy in support of informed decisions for businesses.

The observed advantages include the consistent implementation of data management across the enterprise, control of data access and security, and provisioning of data to the right application at the right time. It removes the complexities of accessing heterogeneous data by different BPs. These DSs are the basic building blocks of achieving customer intimacy, a single view of the customer that enables a high-quality, seamless customer experience. This benefits are not only to the business users but also the customers and partners with whom the business regularly interacts. In addition, SOA governance provides invaluable tools [6] and [7] with the mapping of relationships and dependencies of services, processes, policies, and applications.

A. Concerns of an Interactive Enterprise and Our Analysis to Associate Data Services

Despite DSs' offerings of high and intelligent information value stream, there are uncertainties that dictate an immediate requirement for improving the value and utility of data [8] in an IE. Corporate mergers and acquisitions, changes in regulatory compliance requirements, increased customer attrition, noticeable increase in call center activity, system migrations, improvements to customer-facing BPs, introduction of customer relationship management (CRM), or committing to master data management (MDM) are the initiatives that depend on organizational adjustments in data management practices in order to succeed.

Based on practical assessment, we identified preliminary step in either introducing or updating DSs is to understand the semantics and need of service granularity. It is extremely important to identify all application semantics that exist in the specific aspect. The data and corresponding logical relationship mapping can be dealt properly as well as accurately. Data abstraction, metadata mapping, and ultimately selecting the appropriate technology are the three key steps. We analyzed exactly which data required to be able to build all aspects of the IE and corresponding enterprise entities.

Consequently, we identified and traced the way each participant enterprise entity extract data and utilized them with respective their goals to achieve desired degree of interactivity in an enterprise. Enterprise entities are considered any system or subsystem (example: database, file system, etc.), software, interface, component, infrastructure element, network, device, and third-party or hosted solution participates in formation of an enterprise. Each enterprise

entity utilizes different set of data depending on the goals and nature associated with the particular enterprise entity and corresponding business scenarios. The most prominent example is the payroll system which is managed by third-party. It needs employee data and corresponding information of the pay structure including any deviation such as bonus and reward. Whereas, insurance agent working on commission and not as an employee has also needs to be accommodated in the payroll and either to the similar pay cycle or sporadic payment. The scenario indicates that, even though the third-party is being utilized, they still need diversification and monitory information specific to BPs and APIs.

B. Categories of Data Services

Due to the advancements of DSs technologies and limited standardization, organizations are bound to introduce customized types of DSs and/or approaches to data management. Typically, they are either categorized based on business verticals or horizontally based on departments or units. Primarily, we have identified following categories to diversify the DSs such that granular level of an IE can be articulated and progressed.

- **Enterprise entity management DSs**: DSs that can manage the participant entities of an IE
- **Provisioning DSs**: DSs that can encapsulate changes of data pertaining to application(s) and introduces modes of operations for BPs of an IE
- **Relationship DSs**: DSs that can carry information to specify and retain the relationship between the entities of an IE
- **Association DSs**: DSs that places and provides association between inter or intra enterprise entities to the external participants such as vendors, suppliers, and customers of an IE
- **Governance DSs**: DSs those are specific to the quality of data, policies, security, and other regulatory aspects of an IE

C. Value Stream of Exhibiting Data Services in an Interactive Enterprise

Fig. 1 illustrates value stream to define, orchestrate, and update data services in physical environment of an IE.
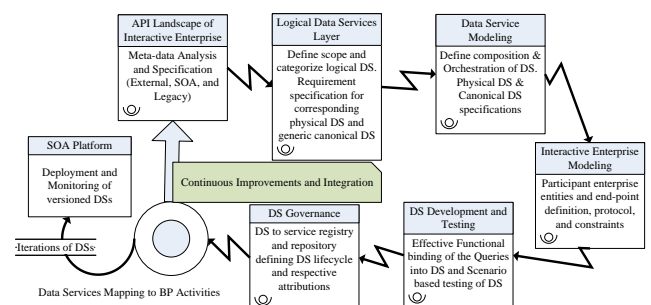

Fig. 1. Value stream to define, develop, deploy, upgrade, and orchestrate types of DSs in an IE.

As indicated in Fig. 1, we have categorized the value stream in eight different steps. It begins with the API landscape analysis; however, the entire value stream cycle is iterative. The iterations are driven by mapping the logical DSs to the BP activities. Any new API that needs to be introduced

has to be analyzed in the perception and input from the mapping specification and rationale between DSs and BP activities. Following are the brief understanding of each value stream and takeaways from each aspects of physical setup for an IE.

1) API landscape analysis of IE

The analysis is essential to understand the overall scope of the IE and perspectives of APIs. APIs can be either standalone or cross functional. In either case, value stream needs to capture meta-data pertaining to both types of APIs

2) Deriving logical DSs layer

Set of logical DSs are orchestrated to be exposed as either composite or single DS to participant enterprise entities of the IE. The affirmative relationship between the data sets and any further desired operations on them needs to be inherited to DS (example: agent and client relationship and computing agent's commission in insurance related business).

3) DS modeling

Generally, three levels of DSs are being introduced, logical, physical, and canonical to compose each category of the DSs. Physical DS is atomic and represents unified query, functions, or stored-procedure. Whereas, canonical DS is focused on defining the structure of the data that is required to be presented with the perception of generic ability to utilize as well as combine underneath physical DSs based on specific requirement. Logical DS either uses or orchestrates canonical and/or physical DSs.

4) IE modeling

Typically, mature enterprise uses enterprise architecture tools and frameworks to model IE. It is to provide appropriate level of dependencies, collaborations, and capabilities of the individual enterprise entity. At some extend, it also specifies the constraints and automation needs of IE.

5) Performing DS development and testing

During the past decade, DS platforms are available and adapted to build and deploy DSs. DS is considered distinct architecture artifact that needs to be developed, versioned, tested, integrated, and deployed. In this value stream, it is required to develop the DSs in the best possible specification and query plan that generates effective response considering each DS (logical, physical, or canonical) as individual artifact. The real-time and responsiveness [9] are the major value that can be brought to IE from it.

6) Implied DS governance

Typically, enterprise must define DS lifecycle corresponding to the lifecycle of the data in its generic sense of enterprise level implications. The DS registry and repository can be utilized to specify classification, enforce policies, and accessibility or security paradigms to DSs. This is the value stream where DSs are physically been classified and governed.

7) Mapping between DSs and BP activities

The DSs are so far being modelled, deployed, and governed in the perception of availability of data and synergies of API in previous value streams. However, if organization has well defined set of BPs then it becomes obligatory to bind the DSs with activities of BPs [10]. The

reusability and consistency across multiple business units are quite visible in this value stream. BPs are becoming the norm and being evolved to most accurately represent current business scenarios and changing dynamics of the marketplace. It is the primary reason for generating need to update and introducing new DSs.

8) SOA platform deployment and migration

SOA platform provides means to deploy, monitor, and audit not only the DSs but also the services generated utilizing the BPs (BP that utilizes DSs) [1] and [4]. It is the platform that enables maintaining and updating the binding between BPs and DSs irrespective of their individual version and upgrades. It is the relationship map of an IE. The quantitative parameters such as reusability and service level agreement (SLA) can be captured with respective to the individual BP, API, and DS.

Each category of DSs identified using the value stream needs to be updated and their cardinality gets increased or decreased based on the lean IA principles identified in Section III (B). The cardinality decides the viscosity of the DSs as specified in Section IV Equation (1).

III. Remitting Data Intensive Interactive Enterprises: Lean Approach

To achieve goals of data intensive IE, we identified the need for instituting foundation and traceable correlations between BPs, DSs, and APIs practices to improve the enterprise's experience while simultaneously maintaining economies of scale.

A. Lean Integration Architecture

Lean approaches focuses to avoid waste and captivating to make the best use of resources and to not produce anything unless it adds value to the customer [11]. The subsequent phase is to reduce inconsistencies by making sure that everything fits together and that every entity is aligned. The final step is to streamline flow in the process, reducing internal inventory and wait states in the processes. Lean objectives such as becoming a team-based learning organization with a sustainable culture of continuous improvement may require years to change entrenched gaps in the organizational processes. It is being observed and recommended that any enterprise utilizing on the Lean IA should establish baseline criteria to place and advance the DSs. In [12], advantages of lean IA are discussed and proven examples are provided in the characteristics of efficiency, agility, data quality, governance, innovation, and density of engagements.

Lean IA is about assembling the architectural parts into a cohesive, holistic system with an edge of streamlining differentiations of business functionalities. DSs are an essential competency in support of IA, BP integration, API correctness, and data governance strategies. IA methodologies involve a process that, like other processes, can be subjected to lean principles. Applying Lean practices to the process of IA can make IA more effective and efficient as indicated in [13]. It subsequently reflects to the DSs in an atlas of IE capabilities. While the industry trends, business needs, and business benefits paint a compelling perspective of

the need for and value of a mature lean IA program, there are a number of significant challenges associated when leveraging it to the DSs that have significant impact to the requirements of an IE. We identified primary challenges ([11], [12], and [14]) that are required to be addressed when defining and upgrading DSs pertaining to an IE.

- Undetermined level of BP involvement during data service definition
- Deficiencies in modeling DSs' complexity that are utilized by applications at runtime
- Quantifiable capabilities of API that leverages numerous types of DSs
- Introduction of additional BP or API without consensus of DS definition
- Investigating quality of the data before physical DSs identification
- Indeterminate scope of logical DSs that are already in usage

For a Lean IA practice to be effective, it must place strategy to continuously provide and evolve principles for each of these challenges based on advancements of correlation between DSs layer and the characteristics of IE.

B. Deriving Principles of an Interactive Enterprise: In the Perception of Data Services

We have rationalized an approach to identify and evolve lean IA principles. Fig. 2 provides step-by-step determination to place and advance lean IA principle for the data intensive IE. The focal point is the IE when deriving lean IA principles. The set of DSs and maturity of lean IA principles are always evolving and it also provides the qualitative measures of the data intensive IE.
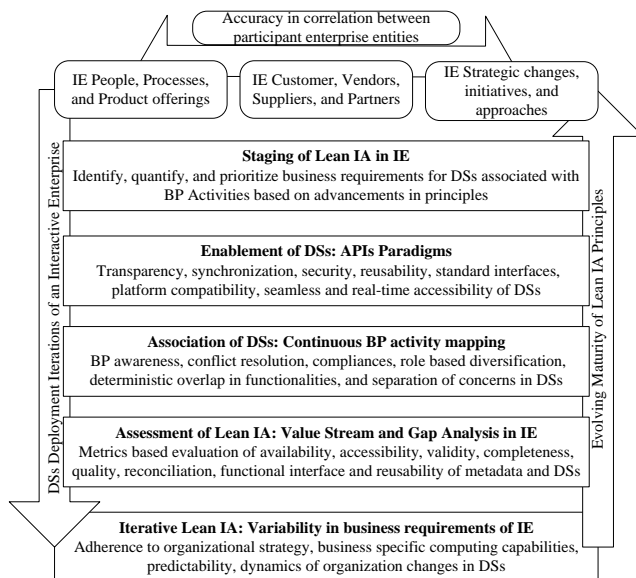

Fig. 2. Deriving Lean IA principles and their continuous evolution for data intensive IE.

The five steps of deriving principles are briefly discussed below in chronological order.

1) Staging of lean IA in IE

Business requirements must have to be modeled in BP and to the corresponding activities. Lean IA indicates eliminating overhead before it occurs. By prioritizing data intensive business requirements, we affirmatively able to derive DS specific technical requirements within the synergy of SOA

platform value stream. It eventually formed the basis for the DSs' orchestration.

2) Enablement of DSs in API paradigms

DSs enables enterprise-wide information model for APIs. API paradigms are assessed and associated with DSs specific attributions such as role based accessibility and temporal properties of data. API also generates meta-data and data. API specific data sets are also required to be considered for consequent introduction or updates of DSs.

3) Association of logical DSs with BP activities

The BP activities have high impact on the DSs. As indicated in staging of lean IA, the separation of concerns identified in BP activities is precisely incorporated to advance the value stream of DS models as well as corresponding orchestration. It also contributed to define granularity of the logical DSs.

4) Assessment of lean IA

Lean practices has been matured enough to indicate the performance of implied lean approaches. However, it is applied to many manufacturing related industries and lately to the software industry. It differs for each enterprise due to the characteristics of business and/or BPs. Metrics to continuously evaluate value stream and gap analysis are defined and periodically (or iteratively) measured. We iteratively perform value stream evaluation and gap analysis with the primary indicators correlating DSs, APIs, and BPs (Section IV).

5) Iterative lean IA

Due to increasing demand of digital business, the way of performing business and BPs are required to be updated and changed. The iterative lean IA is being introduced and carried over to associate anticipated changes of organizational dynamics and strategy within the DSs (example: offering 20% discount for 2 hours for online purchase of a particular product). They are temporal in nature, however, can occur in future with different DSs' attribution.

Following are the preliminary principles and purposes to update DSs based on steps in Figure 2 in iterations.

- Capture the complexities of underlying business scenarios to model DSs based on understanding of data definitions and their utilization across enterprise
- Evaluate, place, and rationalize DS's methods and/or patterns [15] that carters to APIs and associated BP activities
- Specify reusability factor for data services that participates in diversified BPs and streamline monitoring, auditing, and logging capabilities
- Accurate context of API and requirements of physical DSs to logical data services specification
- Scoping and profiling of the data definition during requirements of logical DSs to assess quality of data
- Enforce the DSs governance in the diversity of the enterprise's BP lifecycle from modeling to execution

C. Lean Integration Architecture Trade-off Analysis and Decisions

A lean IA practice can work with and take advantage of data intensive IE at all levels, but there are limitations on what

lean IA teams can accomplish depending on the levels of DSs and maturity of API. For example, if the API maturity is at a level where a holistic inventory of application is not maintained, it will be very challenging for the lean IA team to eliminate certain kinds of redundant data utilization in applications with overlapping DS's functionalities. The lean IA team must either define alternate flow directly to the BP or simply focus on other aspects of lean, such as standard interface definitions and rapid integration development to support capabilities of the IE.

The architecture trade-offs have to be performed and corresponding decisions have to be made available at the core of lean IA. The impact of the decisions becomes the baseline standards to update and introduce granular level of DSs for sustaining as well as improving maturity of data intensive IE. Below is the list of initial lean IA trade-offs for the identified principles.

1) Complexity of business scenarios versus reusability of DSs

It is mandatory for an IE to respond in real-time and efficiently in the dilemma of business scenarios. In certain cases (example: online auction of retail product), it is absolutely necessary to group data and corresponding functions for responding quickly and effectively. Orchestration of DSs may introduce unnecessary bottleneck. It needs to build dedicated DS for the business purpose depending on the limitations of the value stream. However, it must be evaluated before modeling the DS and their estimated performance depending on the sample data set or through emulation.

2) DS governance versus context of API

Maturity of the DS governance is at higher precedence over context of API. Data are the primary proprietary asset of any enterprise and they are growing due to the advancements in digital technologies. API must ensure the correct level of accessibility of these data, however, DS layer itself provides the means to protect privacy and enforce policies. Context of API has to be in alignment with DS governance. If not, context of API has to be reevaluated and appropriate business unit or department has to provide different approach or solution.

3) API requirements versus BP lifecycle

In the case of conflict between the API requirements and BP lifecycle, the BP lifecycle always remains at higher priority. The APIs can be generated and regenerated due to underneath reusability of the BP and desired consistencies between and among APIs. The API requirement becomes obsolete or alternate solution required to be identified unless if there is a need to update BP lifecycle.

4) Quality of the legacy data versus rationalizing DS pattern

Legacy data are primary success factor of any well performing organization. They are always being rich source of information and statistics, unless these data are obsolete and no longer in utilization. It is based on industry segment (example: insurance industry has to retain data over centuries whereas retail products that are no longer available or banned are rarely required to be accessed). Rationalizing of the DS

pattern [15] needs to consider quality of the legacy data based on industry segment and respective business activities. The DS pattern should be ready for any upgrades due to further findings and any new legacy data available (example: due to mergers or acquisition).

The exceptions to the decision due to architecture trade-off analysis can also be identified and specified during the advancements of IE based on maturity as well as measures of IE represented in Section IV.

IV. METHOD OF EVALUATION AND FACTORS OF STANDARDIZATION

We have implied categorization of DSs identified in Section II and lean IA principles derived in Section III to the order management system (OMS). The analysis and deployment iterations include 242 services combining all the DSs specific services, BPs specific services, and APIs specific services (out of more than 1500 enterprise-wide services). The service dependencies map and utilizations are being traced through service registry features as well as capabilities. We derived three metrics to understand the behavior, impact, and measure of the approach.

The viscosity of DSs is defined as the ratio of total number of physical and canonical DSs and corresponding number of logical DSs within the composite or unified DS that falls in one of the category described in Section II (B). If #PDS is the total number of physical DSs (PDS) utilized within the logical DS (LDS) in consideration, #CDS is the total number of canonical DSs (CDS) utilized within the LDS, #C_PDS is the number of PDS utilized in respective CDS, and #LDS is the total number of LDSs to compose the particular type of data service then the viscosity of the data service can be identified as represented in Equation (1).

$$\text{VISCOSITY}_{\text{DS}} = \left(\sum_{i=1}^{\#LDS} [\, \#PDS_i + \#CDS_i + \#C\_PDS_i \,] \right) / \#LDP \tag{1}$$

Degree of API coverage pertaining to DS type (Enterprise entity management, Provisioning, Relationship, Association, and Governance DSs) is derived in correlation with identified, deployed, and utilized BPs consumed by the API. For each API, the BPs that utilizes the type of DS in consideration has been leveraged to determine the coverage and respective impact to the changes of an IE.

If the #BA represents the number of total BP activities defined and placed for the BP in consideration, #DS represents number of DSs utilized to accomplish the BP activity in consideration, and #BP is the number of BPs associated with participant enterprise entities that API is utilizing then degree of API coverage for specific type of DS can be determined using Equation (2).

$$\text{DEGREE}_{\text{API Coverage of DS Type}} = \frac{\sum_{k=1}^{\#BP} [(\sum_{j=1}^{\#BA_k} [\#DS_j\,]) / \#BA_k]}{\#BP} \tag{2}$$

IE capabilities index (IECI) is an instrument to monitor the evolution of basic indicators and make capabilities comparison between and within participant enterprise entities. IECI measures how much variation API (of specific DS type) experiences relative to the limits of business requirements specification. It allows different application to be compared with respect to how well an IE controls corresponding type of DSs. It is based on Taguchi capability index [16]; however, it is in the coordination of business requirements, corresponding changes in the present application deployment iteration, and variations in BP activities in correlation with the type of DSs.

B_USL and B_LSL are the overall upper and lower number of alterations in business requirements specification based on the histogram of the BP activities utilized within the set of APIs (pertaining to the DS type in consideration).

(B_USL – B_LSL) provides finite number indicating allowable BP variations in the effect of APIs' changes dedicated to DS type.

IECI $_{\text{Data Service Type}}$ is derived as shown in Equation (3). "T" is the target state mean of the number of changes to APIs pertaining to specific DS type in the current deployment iteration.

The overall mean of the number of changes to APIs pertaining to the specific DS type is µ.

Actual variability of the overall APIs, expressed as a standard deviation, is σ. It is computed based on the present deployment iterations and relative to the variability information captured during previous deployment iteration.

$$\text{IECI}_{\text{DS TYPE}} = \frac{\text{B\_USL} - \text{B\_L}}{6\sigma\sqrt{1 + ((\mu - T)/\sigma)^2}} \qquad (3)$$

Table I provides our findings of the experimental evaluation in the case of OMS where each category of DS is being metrics in the equations. The data presented in the table are average of all the data services that falls into the particular category. BP activities, BPs, and APIs that utilizes multiple categories of DSs are considered in each category separately to provide more accuracy in comparative analysis.

TABLE I: DATA SERVICES TYPES' AVERAGE SCORE OF VISCOSITY, DEGREE OF API COVERAGE, AND IECI FOR PRESENT DEPLOYMENT ITERATION

DS Type (#DSs)	Average Viscosity	Average Degree of API Coverage	IECI σ = 1.3
Enterprise entity management (23)	6	7.5	1.8
Provisioning (21)	7	8	1.33
Relationship (18)	8.5	4.33	0.83
Association (12)	3	6.5	2.05
Governance (15)	4	5.5	1.66

We gained insight of the capabilities pertaining to the IE considering all the participant enterprise entities. As such, there are no upper or lower limits set for any of the metric; however, the present deployment iteration's results can be utilized as baseline for subsequent updates. Comparative analysis is performed for both, types of DSs and deployment iterations for the application and consequently for the entire enterprise. The outcome also provides indicative characteristics of particular type of DS. For example, relationship data services are high on viscosity whereas relatively low on capabilities compare to other DS types. The overall average degree of coverage (combining all five categories) is being increased by 46% when comparing with previous iteration. The results are very impressive as well as encouraging to approach further subcategorizing the DSs.

V. CONCLUSION AND FUTURE WORK

In this paper, we have identified the challenges and advantages of approaching lean IA principles when emerging data services to the capabilities of an IE. We have provided categorization to the DSs that can be metrics to capture the measurable coverage of API and capabilities of an IE in continuous basis either during the various business initiatives or organizational changes. The lean IA principles are being streamlined and the phases to evolve them are illustrated in the perception of DSs. We have presented an approach to formulate and perform value stream, gap analysis, and comparative analysis between different DSs types. A paper also presented an approach to mature an enterprise in terms of its interactive capabilities during the advancements and upgrades while satisfying the business requirements specification.

The lean IA approach thrive effectiveness and accuracy in modeling between DSs, BP activities, and functionalities of API. Moreover, increasing levels of data and SOA governance have been achieved. Data quality issues are being determined at early stages of the architecture definition. We are progressing to imply the lean IA principles to other aspects of integrations and also determining subcategories of data services that can be leveraged in multilayered APIs pertaining to the IE.

REFERENCES

[1] *Robert Eve (Composite Software), Data Services Platforms — Bringing Order to Chaos*, April 13, 2014.
[2] J. Horkoff and E. Yu, "Evaluating goal achievement in enterprise Modeling — An interactive procedure and experiences," *The Practice of Enterprise Modeling*, vol. 39, 2009, pp. 145-160.
[3] J. Barjis, "Collaborative, participative and interactive enterprise modeling," *Enterprise Information Systems*, Springer-Verlag, vol. 24, 2009, pp. 651-662.
[4] T. Espinha, A. Zaidman, and H.-G. Gross, "Understanding the runtime topology of SOA systems," in *Proc. IEEE 19th Working Conference on Reverse Engineering (WCRE)*, October 2012, pp. 87-196.
[5] S. Brandt, C. Maltzahn, N. Polyzotis, and W.-C. Tan, "Fusing data management services with file systems," *ACM Proceedings of the 4th Annual Workshop on Petascale Data Storage*, New York, NY, USA, November 2009.
[6] P. Bianco, R. Kotermanski, and P. Merson, Evaluating a service oriented architecture, Technical Report CMU/SEI-2007-TR-015, ESC-TR-2007-015, Software Architecture Technology Initiative, Carnegie Mellon University, September 2007.

[7] K. Laskey, P. Brown, J. A. Estefan, K. Laskey, F. G. McCabe, and D. Thornton, *Reference Architecture Foundation for Service Oriented Architecture Version 1.0*, Committee Specification 01, OASIS, December 2012.

[8] A. Harth, "Linked data principles for services and streams," presented at the 2011 STI Semantic Summit, July 2011, Riga, Latvia.

[9] K. O. Park and C. E. Koh, "Effect of change management capability in real-time environment: An information orientation perspective in supply chain management," *Behavior and Information Technology*, vol. 34, no. 1, 2015, pp. 94-104.

[10] B. Hopkins, "The patterns of big data — A data management playbook toolkit," Forrester Research Report, June 11, 2013.

[11] J. Coplien and G. Bjornvig, "What the system is, part 1: Lean architecture," *Lean Architecture for Agile Software Development*, ch. 5, John Wiley & Sons Ltd, 2010.

[12] J. G. Schmidt and D. Lyle, "Chapter 3: The integration factory," *Lean Integration: An Integration Factory Approach to Business Agility*, Informatica Corporation, Addison-Wesley, 2010.

[13] A. Zayati, F. Biennier, M. Moalla, and Y. Badr, "Towards lean service bus architecture for industrial integration infrastructure and pull manufacturing strategies," *Journal of Intelligent Manufacturing*, vol. 23, issue 1, pp. 125-139, February 2012.

[14] D. S. Linthicum, "Defining, designing, and implementing SOA-based data services," *Whitepaper*, David S. Linthicum, LLC, October 2009.

[15] J. Yuan and L.-R. Qiu, "Rule-based business data processing in service coordination systems," in *Proc. 9th IEEE International Conference on Broadband and Wireless Computing, Communication and Applications (BWCCA),* November 2014, pp. 317–320.

[16] M. Nataraj, V. P. Arunachalam, and G. Ranganathan, "Using risk analysis and Taguchi's method to find optimal conditions of design parameters: A case study," *The International Journal of Advanced Manufacturing Technology*, January 2006, Springer-Verlag vol. 27, issue 5-6, pp. 445-454.

Vikas S. Shah received the bachelor of engineering degree in computer engineering from Conceicao Rodrigues College of Engineering, University of Mumbai, India in 1995, and the master of science degree in computer science from Worcester Polytechnic Institute, MA, USA in 1998. Currently he is the 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.