2016 2nd International Workshop on BIG Data Software Engineering

# Toward Big Data Value Engineering for Innovation

Hong-Mei Chen
*Shidler College of Business*
*Univ. of Hawaii at Manoa*
*Honolulu, USA*
*hmchen@hawaii.edu*

Rick Kazman
*Univ. of Hawaii at Manoa*
*& SEI/CMU*
*Pittsburgh, USA*
*kazman@hawaii.edu*

Juan Garbajosa
*Technical University of*
*Madrid (UPM)*
*Madrid, Spain*
jgs@eui.upm.es

Eloy Gonzalez
*Indra Software Labs, Spain*
*Madrid, Spain*
*egonzalezort@indra.es*

## ABSTRACT
This article articulates the requirements for an effective big data value engineering method. It then presents a value discovery method, called Eco-ARCH (*Eco-ARCH*itecture), tightly integrated with the BDD (Big Data Design) method for addressing these requirements, filling a methodological void. Eco-ARCH promotes a fundamental shift in design thinking for big data system design – from "bounded rationality" for problem solving to "expandable rationality" for design for innovation. The Eco-ARCH approach is most suitable for big data value engineering when system boundaries are fluid, requirements are ill-defined, many stakeholders are unknown and design goals are not provided, where no architecture pre-exists, where system behavior is non-deterministic and continuously evolving, and where co-creation with consumers and prosumers is essential to achieving innovation goals. The method was augmented and empirically validated in collaboration with an IT service company in the energy industry to generate a new business model that we call "eBay in the Grid".

## Keywords
Big Data; Value Discovery; Value Engineering; Architecture Landscape; Ecosystem; Innovation; Energy Industry.

## 1. INTRODUCTION
Big data represents unprecedented opportunities for enterprises to compete on analytics for achieving new levels of competitive advantage, including operation optimization, customer intelligence and product innovation. Much value is expected to be derived from high velocity, massive volume of data from everywhere. Big data has the potential to fundamentally transform organizational processes, business models and strategies, and even entire industries and markets [10][12][15].

To extract "value" from big data is daunting task. Organizations face challenges due to: (1) the technical complexity arising from the 4V (Volume, Variety, Velocity, Veracity) characteristics of big data [13]; (2) the organizational agility required for rapid delivery of value [10]; and (3) the rapid pace of technology proliferation and evolution [9]. Big data adoption is surrounded by high level of risks and uncertainty regarding costs, schedules, and benefits. Recently, a new process called "Value Discovery" has become common for big data system development in enterprises. Before deployment, many organizations would engage in a top-down innovation process—for some companies this process alone would take 3 or more years—to discover value from big data. Usually hundreds of use cases would be solicited from different lines of business and a smaller number of use cases would be selected for experimentation or prototyping to understand the costs, benefits as well as risks before deciding to further invest in big data [10][12].


BIGDSE'16, May 16 2016, Austin, TX, USA
© 2016 ACM. ISBN 978-1-4503-4152-3/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2896825.2896837

Such a value discovery process is critical for big data engineering. However, applying traditional "small" data system design thinking and system development approaches to big data value discovery and subsequent value realization, i.e., value engineering, is problematic and inadequate to meet the following requirements (REQs):

**REQ-1: Design thinking for Innovation:** A recent study of large enterprises [12] reveals that enterprises, first and foremost, desire innovation from big data: they are seeking transformational business opportunities and new business models. They want to be the "Uber" of their industry. These types of value expected from big data are a clear departure from the conventional "problem solving" paradigm in software and system engineering. In our empirical case studies, some organizations describe big data as "hammers looking for nails," coming from a problem-solving paradigm and thus they were unsuccessful in adopting big data. Those who did adopt big data did not see big data as a hammer. The focus in these successful cases was on design intention (mindfulness) for innovation and called for a design attitude and method for dealing with the constant and yet continually unexpected, possibly disruptive, innovations in the big data technology ecosystem.

Rooted in problem-solving paradigm, value engineering traditionally focuses on improving the "value" of goods or products and services by using an examination of function. Value is defined as the ratio of function to cost. For more than a decade, value-based software engineering [3] has called for attention to business value and not just software development costs and schedule issues. It advocates integrating value considerations (in contrast to being value-neutral, in which every requirement, use case, object, test case, and defect is equally important) into software engineering principles and practices. Value-based requirements analysis or value analysis involves an approach to improving the value of an item or process by understanding its constituent components and their associated costs. Value-based architecture analysis methods have also been proposed, such as the CBAM [21], which consider the cost, benefits and schedule implications/tradeoffs of different architectural strategies.

Similiarly, economics-driven software engineering is a stream of research that focuses on value, for instance, the ROI of techniques such as refactoring, and technical debt management. Service engineering [5][7] is another research area that has made the issue of business value salient and also integrates corporate governance measures to improve the cost-effectiveness of business-IT alignment. The focus of service engineering and economics-driven software engineering is on a conscious and explicit set of disciplined procedures designed to seek out optimum value for both initial and long-term investment. Big data value engineering shares the same goals. However, existing methods are inadequate for the big data context where data sources, system functions, requirements and hence "values" are continually moving targets. In addition, risks in the big data world are substantially greater, given the large amount of upfront investment and the rapidly changing big data technology landscape. This highlights the need for creativity in addressing the indeterminacy of wicked problems [28] and calls for design thinking that goes beyond the problem-solving paradigm.

**REQ-2: Design for the Open World**: The large variety of data from everywhere is an enormous opportunity for a big data system. It offers opportunities for forming alliances and collaborating with different partners in the supply network and for co-creating value with customers and prosumers [23]. What data sources to include is, however, not a static requirements problem. Traditional value engineering methods were largely based on closed world assumptions, which analyzed the requirements of a project for the purpose of achieving the essential functions at the lowest total costs over the life of the project. The closed-world perspective also assumes that an enterprise has control over the systems designed and that design outcomes are largely deterministic [22]. The fluid system boundary in big data world challenges the old paradigm and requires a new design approach.

**REQ-3: Integrating Value Discovery with Value Realization**: Big data value discovery is inherently a creative effort. There exist many ideation techniques, such as brainstorming (perhaps with electronic support such as Group Decision Support System), technology roadmapping, "blue ocean" strategies [23], etc. that were created to help develop scenarios or business cases. However, these techniques alone fall short for big data value engineering as the scenarios created are not conducive for value realization, e.g., requirements negotiation, architecture design and subsequent system development activities. Existing ideation methods offer little in the way of systematic assistance in reasoning about the risks, costs, and benefits associated with scenarios.

Studies have shown that "separation of concerns" in traditional software engineering is not conducive for value creation [3][21]. For instance, architecture design and requirements negotiations are conceptually tightly related but often performed separately in real-world software development projects. As our prior case studies (e.g. [21]) have revealed, this separation can cause uncertainty in requirements negotiation that hinders progress, can limit the success of architecture design, and often leads to wasted effort and substantial re-work later in the development life-cycle. It is particularly important for big data value engineering that a method can assist stakeholders to create new scenarios, and to elicit, explore, evaluate, negotiate, and agree upon architecture alternatives based on their understanding the architectural implications of each scenario. Such an integration will create a 'generate-and-test' process in rapid cycles. The stakeholders can better understand the ramifications of their requirements (expressed as scenarios) in terms of their conflicts with other requirements, their costs, their schedule implications, and their benefits along multiple quality attribute dimensions. As such, the stakeholders can make better decisions about their requirements and prioritize scenarios based on better informed and more holistic value decisions.

**REQ-4: Support for Value Experimentation and Verification**: Value engineering elicits ideas on alternative ways of maintaining or enhancing results while reducing life cycle costs. Value engineering can be applied at any point in a project, even in construction. However, typically the earlier it is applied the higher the return on the time and effort invested. Due to the scale and scope of big data projects, estimating total cost of ownership is difficult and complex and the system qualities (performance, scalability, interoperability) can not be cost-effectively measured by traditional horizontal prototyping methods. The rapid rate of technology proliferation and evolution in the big data area also creates problem for value assessment. We have proposed an architecture-centric approach, combined with strategic prototyping, to address this concern [9].

To meet the above-mentioned requirements, this paper argues for an eco-architecture approach for big data value engineering. We present an empirical case study in the energy industry to demonstrate the validity of our method, called Eco-ARCH. We first describe how our case company (which we refer to as IND) failed in their initial attempts at envisioning and designing a big data system, and then applied the Eco-ARCH method to discover value from big data and generate a new business model. We called this new business model "eBay in the Grid" where energy companies are not just suppliers of power for traditional consumers but also platform providers for emerging, evolving energy markets.

## 2. An Eco-Architecture Approach

The Eco-ARCH method was originally developed for ultra-large systems where no architecture exists [8][24]. An exploratory action research was conducted on the Demand-Response component of the U.S. Smart Grid for design, development and validation of the Eco-ARCH method. The Smart Grid is a complex multi-layered ecosystem composed of an enormous number of constituent systems; each electric utility company typically contains many constituent systems. There is no central planning for the system architecture although there are some constraints, such as policies and legal requirements addressing issues of continuity of service and public safety. Electric utilities are being asked to plan for the Smart Grid of the future where technologies and consumer behavior are constantly changing.

The steps of the Eco-ARCH method are illustrated in Figure 1. Each step of ECO-ARCH contains two levels of analysis: Macroscopic and Microscopic. Note that there are iterations between levels and among the steps. The method provides paired macroscopic and microscopic analyses, supporting innovation in dealing with wicked problems while offering a rational design process based on proven engineering techniques for meeting quality attribute requirements and integrating with "triple bottom line" goals: profit, people, and planet. It advances the frontiers of design science to deal with indeterminacy in system requirements, system behaviors and design outcomes. It also advances architecture analysis by focusing on the architecture landscape in an open ecosystem, instead of the single, concrete architecture assumed in traditional methods.

We created the Eco-ARCH method as the basis for big data value engineering because it embodies design thinking rooted in expandable rationality (addressing design for innovation) [17][18] and employs rigorous engineering principles (addressing efficient problem-solving). This method encourages a "futuring" mindset via ecosystem-wide scenario brainstorming, guides the construction of an architecture landscape for risk analysis, and uses balance-scorecard techniques for cost-benefit analysis. Value-based requirements analysis is integrated with the first step of value realization, as we will show.

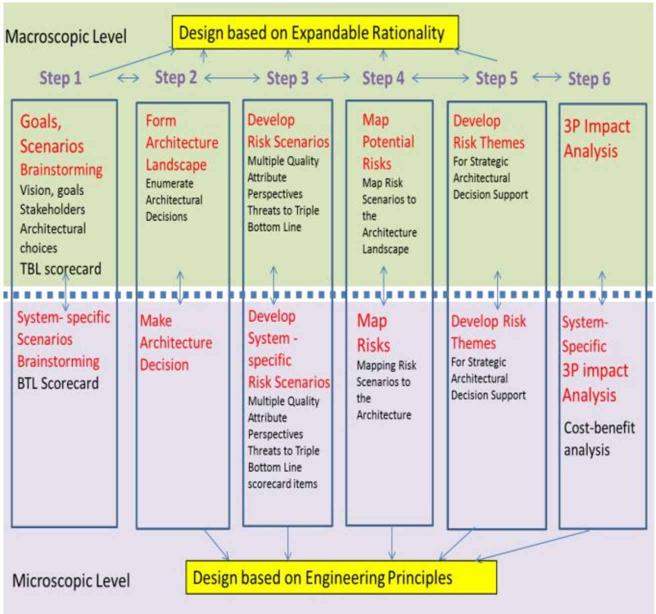


**Figure 1: The Original Eco-ARCH Method**

We must here distinguish between "design" (for example, making a graphic cover for an album, or composing a song) and "problem solving" (e.g.., playing chess, doing a puzzle). Design is essential to innovation and problem-solving is a moment in design. Engineering is fundamentally a problem solving paradigm grounded in "bounded rationality" [26]—essentially a search through a space of possible solutions, ending when a satisficing (i.e., good enough) solution is found. Since the bounded rationality paradigm treats design as a problem-solving activity it may limit creativity. Design thinking based on "expandable rationality" [17] sees problem-solving as only a moment within a design process.

Eco-ARCH breaks from the old paradigm. The basic premises of expandable rationality are that: 1) design problems are wicked problems, 2) design problems are not fully knowable and they evolve during the process, 3) a design attitude sees problems as opportunities for the invention of new alternatives, and 4) problem solving is a subset of innovative design.

# 3. Augmented Eco-ARCH Method

Value engineering includes two phases: Value Discovery, and Value Realization. We have augmented the original Eco-ARCH method for big data value discovery with: 1) "Priming" techniques [16] for Futuring scenario generation, 2) a Big Data Architecture Scenario (BDAS) template for big data modeling, 3) a Big Data-Data Flow Diagram (BD-DFD) for process modeling, and 4) strategic prototyping [9] to meet the requirements stated above. These augmentation techniques were each validated independently before being integrating into Eco-ARCH.

The augmented Eco-ARCH method is integrated into the Big Data Design (BDD) methodology [13] which is focuses the practitioner on value realization. The steps of the augmented method are depicted in Figure 2. Steps 0 to 7 are in the Big Data Value Discovery phase while Steps 8-10 are in the Big Data Value Realization phase. We will only briefly describe these steps as BDD has been validated in separate studies and described elsewhere in detail in [13]. Included in the augmented Eco-ARCH Steps 4-7 are steps for integrating value discovery and value realization as REQ-3 stipulates. REQ-4 is met specifically in Steps 6-7 that support value experimentation and validation. REQ-1 and REQ-2 are satisfied by Steps 1-4.

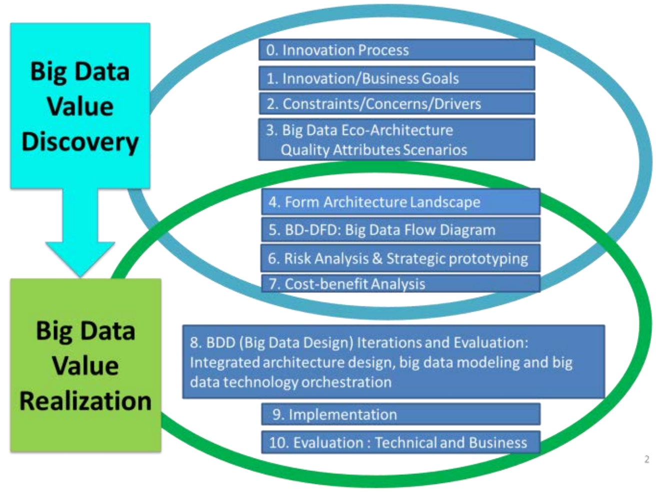


**Figure 2: The Augmented Eco-ARCH Method for Big Data Value Engineering**

**Step 0 - Innovation Process**: A top-down innovation process is essential to involve as many stakeholders as possible [12][10]. In many cases, stakeholders are not known at the beginning and will evolve and be included when appropriate.

**Step 1 - Innovation/business Goals**: Eco-ARCH starts with brainstorming for defining business and innovation goals. It's important to embrace "mindfulness" for innovation to discover value. An organization must explicitly attend to innovation for it to occur; this is known as innovation mindfulness [17]. The architecture or system design is an important tool to effect innovation based on this mindfulness perspective. It employs a futuring technique: stakeholders are guided to free themselves from existing conditions and existing business models and imagine what the future would be. At every brainstorming workshop, participants are "primed" to envision future systems for big data. Following a value-based approach, different goals will be voted on and prioritized. The top 5 goals (or any smaller number of goals agreed upon by the stakeholders) will be selected for expressing as quality attributes scenario in Step 3.

**Step 2 - Constraints, Concerns, and Drivers**: Unlike the traditional single-architecture system, the constraints, concerns and drivers for the big data ecosystem are brainstormed and then modeled, rather than being based on the existing business models. Risks and costs are often viewed as drivers in architecture decision-making and hence Eco-ARCH has incorporated approaches such as the Cost Benefit Analysis Method (CBAM) [21] that consider architectural decisions as investment decisions, as shown in Step 7.

**Step 3 - Big Data Eco-architecture Quality Attribute Scenarios**: In this step, new scenarios are brainstormed for each prioritized business or innovation goal. Ideas for innovation are elaborated and modeled using big data quality attribute scenarios [1]. Each of these scenarios may have an impact on the architectural design decisions that have been made (perhaps even as constraints), or not made. The big data scenarios focus on architecturally significant requirements—quality attributes such as performance, availability, etc. We then record big data modeling inputs using the BDAS template for each scenario. This template captures 14 data architecture elements, including data source quality, data variety, data volume, velocity, read/write frequency, time to live, queries, OLTP or OLAP, etc. Each input has a direct implication on subsequent architecture choices, data model selection, technology selection, and data access patterns. The template allows easy documentation of the data sources and requirements and facilitates data modeling during Step 8 in the Value Realization phase.

**Step 4 - Form Architecture Landscape**: An architecture landscape, encompassing all architecture choices derived from the scenarios, is a critical step for future systems where no existing system architecture is in place. Each scenario will exercise some architectural alternatives. By considering the full set of scenarios, an architectural landscape may be drawn, showing the possible alternatives for the creation and operation of the system. Each alternative represents a significant architectural decision that must be made. The alternatives may be based upon logical options (e.g. push versus pull communications; acknowledgement of messages or not), commercially available components (e.g. available types of networks or devices), or decisions within an architectural element (e.g. frequency of messaging; message re-send policy).

**Step 5 - BD-DFD (Big Data - Data Flow Diagram)**: Creating the BD-DFD is also a creative process that connects all the data flows and processes in each scenario and composes the context diagram for the architecture vision for the future. The BD-DFD can also be generated for a specific instance of architecture by selecting elements in the architecture landscape. Data flow diagrams are a familiar modeling tool, and we did not need to add any new constructs to them. The BD-DFD facilitates big data modeling, in Step 8.

**Step 6 - Risk Analysis and Strategic Prototyping**: In this step, risks are analyzed using a combination of architecture analysis and strategic prototyping [9] to achieve the value-based objectives. Using architecture analysis, risk scenarios are developed to describe challenges to the system from multiple quality attribute perspectives and threats to the triple bottom line. It is not enough that an architecture works well under normal conditions (those described in the scenarios), but it must work well when stressed, when faced with unexpected demands or failures, or when faced with evolutionary pressures. Risk scenarios are chosen to understand the implications of such challenges on architectural decisions. When risk scenarios are mapped onto an architectural landscape, the assumptions lurking behind each architectural decision become evident. Some of these assumptions, alone or in combination, may pose risks for the achievement of a system's quality attribute goals. This mapping, along with a model of each quality attribute, is the basis for the architecture analysis in the traditional ATAM [1]. In addition, the risks that we find as a result of the scenario mapping process can be consolidated into risk themes. In mapping substantial numbers of scenarios we often see the same kinds of risks emerging over and over. Such themes need to be explicitly identified as these pose the greatest risks to the success of the system. An architectural analysis exercise always locates many potential risks but not all risks are equally likely and not all of them have the same set of consequences. The commonalities in the risks found have led us to "roll up" many of the risks into themes so that these may be made the focus of future investigations.

Architecture analysis is a relatively low cost option for risk analysis. However, there are situations where architecture analysis alone can not provide an adequate understanding of the risks. In such situations, prototyping is often required. In cases where prototyping is involved, we have developed *strategic prototyping* guidelines [9] for either doing a throwaway or vertical prototype, rather than more expensive horizontal prototyping. Note that strategic prototyping in value discovery phase is different from "small" data system development where horizontal prototyping is the norm and it is often conducted in the value realization phase.

**Step 7 - Cost-benefit Analysis**: In this step, we employ techniques, such as the CBAM [21], to do a cost-benefit analysis of the architecture decisions and risks uncovered with respect to the business and innovation goals. Risks are *potential* problems, and we strive for early identification of risks as a means of assessing their impact and preventing them from being realized. Each risk might then be further analyzed in more detail, e.g., by building a performance model, or by creating a simulation, an experiment, or a prototype. And each of the architectural decisions that go into this risk should be keenly scrutinized by any architect, at both macroscopic and microscopic levels.

**Step 8 - Big Data Design (BDD)**: In this step, an architect must choose specific architecture elements—specific patterns and technologies—from architecture landscape to form an implementable instance of a system. The details of the BDD steps are described in [12] and depicted in Figure 3. BDD tailors and augments ADD (Attribute-Driven Design) [4] for combining architecture design, big data modeling and technology selection in an iterative design process to optimize each iteration goal.

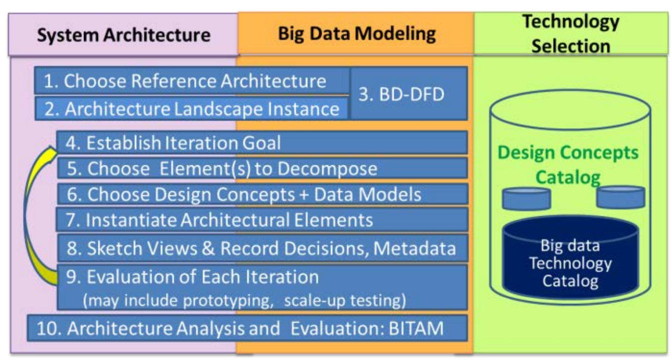


**Figure 3: Design steps of the BDD Method**

# 4. Empirical Case Study

Our case company, IND, is a company in the energy industry, providing outsourcing and integration services. It has 44,000 employees. Our case study was performed with the energy R&D Division from Fall 2014 to Summer 2015. This division focuses on two areas: 1) distribution operation and automation and 2) network monitoring and control operations: real time and quasi-real-time information for managing networked smart devices for the electrical distribution application domain. Smart grid management requires dealing with huge amount of data collected from smart meters and other devices connected to the power network. Currently, data is gathered in large volumes from meters and analyzed "off-line" in time-constrained periods (quarterly, hourly, daily, and monthly). However, electricity companies are demanding IT solutions to deal with the smart monitoring of power networks and perform big data analytics to drive insights: including: 1) energy efficiency by analyzing customer consumption summaries or real-time usage, 2) theft detection, 3) load forecasting to optimize utility companies' purchasing and generation decisions, 4) grid utilization, as well as outage prediction and detection, and 5) customer experience. Many use cases have been developed. This domain has many constraints in terms of government regulations and hundreds of standards for smart devices. In addition there are many suppliers of hardware, software, and services; the market is very granular, with stringent requirements including decision-making in nanoseconds with significant consequences. The energy industry is in a turbulent period, experiencing both enormous threats and enormous opportunities from new storage systems, new power generation systems, electrical vehicle infrastructure, and micro grids. IND was positioning itself to be not just a service provider and integrator, but also a "Grid Analytics Platform."

IND prides itself in innovation. "We use whatever technology is available at the time" stated the division head. They considered big data technology in 2009, but this technology was deemed too immature back then. They experimented with many different combinations of big data technologies and related architectures. The results were, however, unsatisfactory. They had issues with inconsistency of data (for instance, readings could be affected by the weather, by differences in smart devices, etc.) and so they needed to perform extensive consistency checks when they received the data. The data sources were also very diverse and IND was unable to achieve system scalability. They ceased experimentation with big data in 2011 and restarted in 2014. They have experimented with edge computing and fog computing for putting control and data processing power into smart devices to speed up data retrieval time.

Pursuing innovation, IND agreed to follow the Eco-ARCH method for big data value discovery. Two of the authors of this paper were creators of the Eco-ARCH method and were facilitating the steps for IND. Their high-level innovation goals, after voting and prioritization, were:

1. Reduce cost
2. Increase capabilities/quality (attributes)
3. Improve market position: product lines, time to market, differentiating features
4. Improve business processes: better, faster, smarter, cheaper, employee training/retention, DevOps
5. Improve confidence and image of the system: end users, customers (utilities), partners

All 5 of these goals are, in the end, related to Goal 2—without increasing their underlying capabilities, none of the other goals would be possible. Given these goals the IND stakeholders proceeded to generate big data quality attribute scenarios: 17 stakeholders generated over 50 scenarios. As a way to classify them and understand whether they had achieved adequate coverage, the scenarios were mapped to the current TC57 reference architecture, as shown in Figure 4.

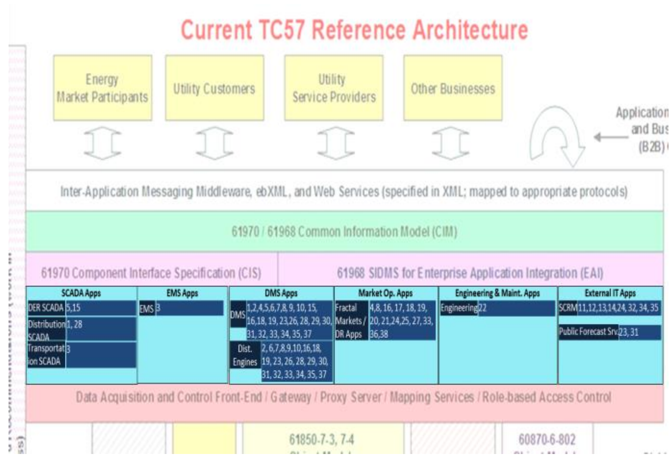


**Figure 4: The Current TC57 Reference Architecture Annotated with Brainstormed Scenarios**

After the big data scenarios were mapped and BD-DFD were sketched, a new business model emerged, which we called "eBay in the Grid." The idea was that, in the past, utilities and service providers saw themselves as primarily creators of systems that had relatively limited objectives: for generation, transmission, distribution, accounting, and so forth. By considering the architecture landscape, IND saw a new business opportunity in building a *platform*, providing a broad basis of generic services, data, and analytics upon which others could build value-added services.

The feedback that we received from IND in using the method was as follows:

1. The method allows them to think beyond their current state and guide to systematically explore various innovation options that they had not conceived before.
2. It was quite challenging for them to think beyond the current state at the beginning but after several working sessions, they were able to formulate quality attribute scenarios much more easily.
3. It was also difficult for them to understand what requirements were "architecturally significant" and what are not.
4. The priming exercises were received positively. They said that it cleared their minds and aided in the brainstorming process.
5. The discovery of a new business model was a "nice surprise".

## 5. Discussion

Although this was just a single case study, we have received generally positive feedback from IND regarding the use of Eco-ARCH for big data value discovery. However, this exercise illuminated 7 limitations of the method. These are each discussed below:

First and foremost, a complete and authoritative validation of Eco-ARCH is impossible. The concept of an architectural landscape, along with techniques for envisioning, analyzing, and scoring realizations of the landscape is, we believe, our most important contribution. A key element of Eco-ARCH is expansive design thinking for innovation, and the outcomes from this process—the architecture landscape and associated scenarios and risks. The discovery of new kinds of value propositions, such as a new business model in our case, are obviously useful but it must be noted that the exact conditions to induce such creativity are hard to pinpoint.

Second, Eco-ARCH guides the stakeholders to analyze potential risks lurking within the landscape, their consequences, their interactions, and their tradeoffs. Eco-ARCH supports risk-based reasoning for cost-benefit analysis of architecture decisions, which are investment decisions.

To validate the effectiveness of the method, we examined two aspects of the risks discovered: coverage and correctness.

1. Did we find a majority of the most important risks?
2. Were the risks that we found truly significant challenges to the achievement of some important system goal?

When performing an architectural evaluation using a technique such as the ATAM, correctness and coverage are reasonably easy to achieve, as long as the method is faithfully prosecuted. Coverage is achieved by ensuring that the appropriate stakeholders are involved, by capturing their concerns as scenarios, and by tracing the most highly ranked scenarios through the architecture. Correctness is achieved because the risk determination is done in real time, in front of all the stakeholders. If risks are misunderstood or misidentified, these mistakes are immediately apparent and corrected.

When evaluating an underspecified big data architecture with multiple unknown and unknowable stakeholders, it is not possible to get agreement on the risks. Hence we must rely on the stakeholders present as proxies. There is another, more fundamental issue with all risk evaluation and prevention methods. How do you measure the benefit of the risk avoided?

Third, a limitation of the present study is regarding the generalization of the case method. Thus far we have only worked with a single organization, although many parts of Eco-ARCH have been validated independently through other case studies.

Fourth, Eco-ARCH, like all methods cannot guarantee that the resulting system will be usable or that the policies that these systems enact will be attractive to a consumer. For successful big data engineering, the system must it easy to change such policies, with few ripple effects on other parts of the system.

Fifth, The Eco-ARCH method relies heavily on brainstorming for scenarios. Although the scenario-based approach has been heavily used in architecture design and analysis in the past, it cannot be overstated that the quality of the scenarios generated is critical. Expandable rationality design thinking focuses on a "breadth-first" strategy to co-evolve both problems and solutions, which may generate large numbers of scenarios. In our research, the issue of managing large numbers of scenarios and ensuring the quality of the scenarios selected in the ideation process [6] was a major challenge. All the researchers in the present study have deep experience in facilitating and consolidating scenarios, which may not be the case for other projects that adopt the Eco-ARCH method.

Sixth, the successful use of Eco-ARCH will rely on a top-down innovation process, requiring an alignment with an organizational culture that fosters innovation, is open to change, and possesses (or is willing to construct) an agile infrastructure. Eco-ARCH can assist big data value discovery and facilitate big data value realization with architecture agility, but the organizations must have innovation "mindfulness" to achieve innovations goals.

Seventh, the successful use of Eco-ARCH also depends on the ability of the architects. Traditional IS and CS curricula emphasize engineering approaches to problem-solving. As a result, it is difficult for software engineers and architects to make the transition to a more open, creative space. We will have to rethink IS curricula to address this challenge for big data value engineering.

## 6. Conclusions

Eco-ARCH addresses four big data value discovery requirements:

- REQ-1: Design thinking for Innovation
- REQ-2: Design for the Open World
- REQ-3: Integrating Value Discovery with Value Realization
- REQ-4: Support for Value Experimentation and Verification

resulting in a new approach to eliciting requirements for, envisioning, and designing, systems in to compete in a fast-moving, open marketplace.

Eco-ARCH embodies design thinking rooted in expandable rationality. It provides a dual macroscopic-microscopic analysis technique, allowing for innovation in dealing with wicked problems in the big data open world while offering solid engineering-based design with proven techniques for system-specific quality attribute evaluation, risk-based cost-benefits evaluation. We augment Eco-ARCH with four additional contributions: 1) "Priming" techniques for Futuring scenario generation, 2) a Big Data Architecture Scenario (BDAS) template for big data modeling, 3) a Big Data-Data Flow Diagram (BD-DFD) for process modeling, and 4) strategic prototyping. Our contribution of the architecture landscape addresses open world design problems while big data quality attributes scenarios facilitate data modeling in subsequent design steps. Strategic prototyping is an integral part of Eco-ARCH, utilizing the value-based engineering principle. Employing the Eco-ARCH method, our case company was able to conceive a radically new business model for not only tackling business transformation imperatives in the energy industry but also to achieve their innovation goals.

The Eco-ARCH is tightly integrated with BDD, as REQ-3 stipulates for full life cycle support of big data value engineering: from value discovery to value realization. Our approach calls for transformative design thinking to enable innovation. Finally, we foresee a critical challenge inherent in this approach, which is to revolutionize our IS education, to create business analysts, designers, and software engineers in the future who are equipped with the skills needed to create open-world innovations in a highly competitive marketplace.

## 7. ACKNOWLEDGMENTS

We would like to gratefully thank and acknowledge the vigorous support of the research group members at UPM as well as Indra's software architecture team.

## 8. REFERENCES

[1] Bass, L., Clements, P., and Kazman, R. 2012. *Software Architecture in Practice*, 3rd ed., Addison-Wesley.

[2] Buchanan, R. 1992. "Wicked problems in design thinking," *Design Issues* (8:2), pp. 5-21.

[3] Boehm, B. "Value-Based Software Engineering: Overview and Agenda," USC Technical Report, USC-CSE-2005-504.

[4] H. Cervantes, R. Kazman, *Designing Software Architectures: A Practical Approach,* Addison-Wesley, 2016.

[5] Chen, H-M. 2008. "Towards Service Engineering: Service Orientation and Business-IT Alignment," *Proceedings of the Hawaii International Conference on System Science (HICSS-41)*.

[6] Chen, H-M., and Vargo, S., 2008. "Toward an Alternate Logic for Electronic Customer Relationship Management", *International Journal of Business Environment* (2:2) pp. 116-132.

[7] Chen, H-M, Kazman, R., and Perry, O. "From Software Architecture Analysis to Service Engineering: An Empirical Study of Methodology Development for Enterprise SOA Implementation." *IEEE Transaction on Services Computing*, April-June 2010 (vol. 3 no. 2), pp. 145-160

[8] Chen, H-M, and Kazman, R. 2012. "Architecting for Ultra Large Scale Green IS," *Proceedings of GREENS 2012: First Workshop on Green and Sustainable Software (GREENS)* at ICSE 2012.

[9] Chen, H-M, Kazman, R., and Haziyev, S. "Strategic Prototyping for Developing Big Data Systems," *IEEE Software*, to appear, 2016.

[10] Chen, H-M, Schütz, R., Kazman, R., and Matthes, F. 2016. "Amazon in the Air: Innovating with Big Data at Lufthansa," *Proceedings of the Hawaii International Conference on System Science (HICSS-49)*.

[11] Chen, H-M, Kazman, R., and Haziyev, S. "Agile Big Data Analytics Development: An Architecture-centric Approach," IEEE *Proceedings of the Hawaii International Conference on System Science (HICSS-49)*.

[12] Chen, H-M, Kazman, R., and Matthes, F. "Demystifying Big Data Adoption: Beyond IT Fashion and Relative Advantage," *Proceedings of Pre-ICIS (International Conference on Information System) DIGIT workshop*.

[13] Chen, H-M, Kazman, R., Haziyev, S. and Hrytsay, O. 2015. "Big Data System Development: An Embedded Case Study with a Global Outsourcing Firm," *Proceedings of BIGDSE'15 (Big Data Software Engineering) Workshop* at ICSE 2015.

[14] Clements, P., Kazman, R., and Klein, M. 2001. *Evaluating Software Architectures: Methods and Case Studies*, Addison-Wesley.

[15] Davenport, T.H., Barth, P., and Bean, R. 2012. "How Big data is Different," *Harvard Business Review* (90: 10), pp. 78–83

[16] Dennis A. R., Minas R. K. and Bhagwatwarb A. P. 2013 "Sparking Creativity: Improving Electronic Brainstorming with Individual Cognitive Priming," *Journal of Management Information Systems*, Volume 29, Issue 4, 195-216.

[17] Fishman R. G. 2004. "Going Beyond the Dominant Paradigm for Information Technology Innovation Research: Emerging Concepts and Methods," *Journal of the Association for Information Systems* (5:8), pp. 314-355.

[18] Hatchuel A. 2002. "Towards Design Theory and Expandable Rationality: The unfinished program of Herbert Simon," *Journal of Management and Governance* (5:3-4), pp. 260-273.

[19] Hatchuel, A., and Weil, B. 2009. "C-K Design Theory: An Advanced Formulation," *Research in Engineering Design* (19), pp. 181-192.

[20] Hevner, A., March, S., and Park, J. 2004. "Design Science in Information Systems Research," *MIS Quarterly* (28:1), pp. 75-105.

[21] Kazman R., Asundi J., Klein M. "Quantifying the Costs and Benefits of Architectural Decisions", *Proceedings of the 23rd International Conference on Software Engineering (ICSE 23),* pp. 297- 306.

[22] Kazman, R., In, H. and Chen, H-M. 2005, "From Requirements Negotiation to Software Architecture Decisions," *Information & Software Technology*, 47(8), pp. 511-520.

[23] Kazman, R., and Chen, H-M. 2009. "The Metropolis Model: A New Logic for the Development of Crowdsourced Systems," *Communications of the ACM* (52:7), pp. 76-84.

[24] Kazman, R., Bass, L., Ivers, J., and Moreno, G. 2011. "Architecture Evaluation without an Architecture: Experience with the Smart Grid", *Proceedings of 33rd International Conference on Software Engineering (ICSE 33)*.

[25] Kim, S., In, H., Baik, J., Kazman, R., and Han, K. 2008. "Escaping from Red Ocean with Value-Innovative Requirements", *IEEE Software*, January/February 2008, pp. 80-87.

[26] Kimbell, L., and Street P.E. 2009. "Beyond design thinking: Design-as-practice and designs-in-practice," *CRESC Conference*, Manchester.

[27] Naedele, M., Chen, H-M, Kazman, R., Cai, Y., Xiao, L., and Silva, C.V.A. 2015. "Manufacturing Execution Systems: A Vision for Managing Software Development," *Journal of Systems and Software*, Volume 101, March 2015, pp. 59–68.

[28] Northrop, L., Feiler, P., Gabriel, R., Goodenough, J., Linger, R., Longstaff, T., Kazman, R., Klein, M., Schmidt, D., Sullivan, K., and Wallnau, K. 2006. *Ultra-Large-Scale Systems: The Software Challenge of the Future*. SEI/CMU.

[29] Porter, M., and Kramer, M. 2006. "Strategy and Society," *Harvard Business Review* (84:12), pp. 78-92.

[30] Rittel, H. J., and Webber, M. M. 1984. "Planning Problems Are Wicked Problems," in *Developments in Design Methodology*, N. Cross (ed.), John Wiley & Sons, New York.

[31] Sarasvathy S., Dew N., Read S. and Wiltbank R. 2008. "Designing Organizations that Design Environments: Lessons from Entrepreneurial Expertise," Organizational Studies, 29(03), pp. 331–350.

[32] Simon, H. 1996. *The Sciences of the Artificial*. MIT Press, Cambridge.

[33] Sterman, J. D. 2001. "System Dynamics Modeling: Tools for Learning in a Complex World," *California Management Review* (43:4), pp. 8-28.