RC25130 (W1103-130) March 22, 2011
Computer Science

IBM Research Report

Managing Enterprise IT Systems Using Online Communities

Maja Vukovic, Vijay K. Naik
IBM Research Division
Thomas J. Watson Research Center
P.O. Box 704
Yorktown Heights, NY 10598
USA

Research Division
Almaden - Austin - Beijing - Cambridge - Haifa - India - T. J. Watson - Tokyo - Zurich

Managing Enterprise IT Systems using Online Communities

Maja Vukovic and Vijay K. Naik
IBM T.J. Watson Research
Hawthorne, NY 10532, USA
{maja, vkn}@us.ibm.com

***Abstract*—Effective design and implementation of any IT Optimization process relies on critical technical and business insights about IT environment. The essential information, which relates to hardware and software assets, is often in collective possession of the infrastructure and application specialists. In this paper, we describe the Knowledge Harvesting & Information Synthesis System (KHISS), which engages enterprise online communities to capture the information required for IT optimization activities.**

We describe our experience in deploying KHISS principles to manage enterprise IT infrastructure and application portfolio, as a first step in on-Cloud migration – a large business transformation activity. We discuss the challenges in attracting and maintaining enterprise online communities, as well as in assuring the quality of knowledge collected through this collaborative mechanism to guide the strategic decisions about enterprise IT environment.

Keywords: datacenter management, enterprise IT, transformation to cloud, online communities;

I. INTRODUCTION

Enterprise IT systems are continuously challenged to support business processes in cost-effective and efficient manner leveraging of the latest evolution of technologies without affecting business continuity and integrity. This involves supporting not only the lifecycle operations such as change management, but, from time to time, also performing migrations, transformations, consolidations, and optimizations. Whether it is asset management, maturity assessment, transformation analysis, migration planning, or business process transformations, in each case, a detailed inventory of IT systems, dependency analysis, performance and operational requirements, and workload characteristics are a prerequisite. Because IT systems are a heterogeneous collection of different types of hardware and software components, each with its own lifecycle and timescale, the state of IT systems is continuously in a dynamic flux. Even very well managed enterprise IT systems find it hard to get a snapshot of the state of their IT systems at a short notice. Some of the key reasons for this are: (i) scale and distributed nature of the information, (ii) dynamic and complex interactions, and (iii) knowledge intensive processes.

Nature of Information: Data elements that capture the structure and operational aspects of enterprise IT systems encapsulate descriptions and configurations of physical and virtual servers, middleware, business applications and their associated security and business control parameters; as well as the teams of business and technical specialist responsible for hosting and managing them. In a large global enterprise, these elements are typically handled by specialized task teams, which focus on capturing this data in a number of data repositories, e.g. server repository, application repository, etc. Because of increasing degree of process automation and integration, information associated with end-to-end IT and business operations tend be knowledge intensive and domain specific.

Nature of complex IT operations: In order to perform IT Transformation activities and to continuously optimize the operations, it is important to be able to quickly discover information such as "Which servers are hosting application X?", "Which applications are collocated?", "Which applications are subject to US ITAR?", etc. Given the disparate data repositories, today it's not easy to efficiently discover such inferred knowledge. As a result, collecting, managing and consolidating this information in order to provide business and technical insights and analyze effects of any change or transformation is becoming a challenging endeavor.

When faced with a decision on whether to undertake a transformation or not, current approaches typically involve transformation experts manually reaching out to known knowledge owners (business, financial and technical specialists), to collect and manage the necessary information in shared spreadsheets. This involves handling responses through e-mail and chat, and even phone conversations and tracking the chain of responses, as often-targeted experts refer the request to another team member. Such practices tend to be ad hoc and at best provide crude estimates.

In this paper, we describe a novel approach that intelligently collects available information and augments with expert knowledge diffused within the enterprise in a cost-effective and efficient manner. In the following, we first characterize the distributed nature of the data and knowledge associated with key service management functions and highlight the challenges in knowledge harvesting for efficient IT transformation. In Section 3, we propose Knowledge Harvesting & Information Synthesis System (KHISS) for capturing information and building knowledge bases.

Section 4 describes the application of this system for the first stage of on-Cloud migration process. We describe mechanisms for engaging online communities for IT information management and highlight challenges in incentivizing participation and data quality insurance. Section 5 presents the related work and Section 6 concludes the paper with our experiences and observations from practical deployment and lays out future work directions.

II. MANAGING ENTERPRISE IT

A. *Dynamics*

Shown in Figure 1 is a typical IT system in an enterprise environment. The figure also shows the complexities of managing enterprise IT and, in particular, those associated with managing IT transformation.

Figure 1 shows a set of physical servers connected to each other via enterprise network. In this scenario, a virtual environment is assumed, where one or more virtual machines (VMs) are deployed on each physical server. A VM deployed on physical servers depends on that server for providing its services. Firmware and hypervisor are installed and running on each physical server. An operating system (OS) is installed in each VM and on top of the OS, middleware and other applications may be installed in each VM. As shown in Figure 1, various applications and middleware systems such as application servers, web servers, databases, etc. may be installed and running in a VM. In some cases a single application or a single middleware system may be installed in a VM and some other cases multiple systems may share the same VM. Various constraints and tradeoffs determine sharing and placement of VMs on a physical server as well as applications and other services in a VM. The operation of an application server or a web server may depend not only on the OS and the VM where it is installed, but may depend on the services provided by other components such as the database server,

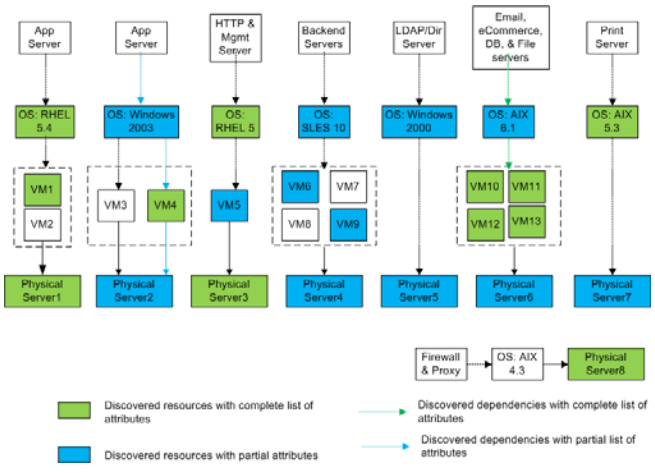

Figure 1 Sample view of information captured in a system managing the datacenter knowledge

LDAP server, a file server, or a backend application. Thus a large number of dependencies may exist among the IT components. Moreover, the IT components are not standalone functional components, but they support business workload generated by business processes. Performance, security, compliance, and availability requirements may determine how many instances of application or web or database servers are to be deployed at any given time. They also determine the type of isolation and secure connectivity is to be enforced around the IT components processing the workload.

When transformations are considered, not only the inventory of the IT assets needs to be captured, but also the dependencies and complex interactions among components enforced by the dependencies and the workload constraints. Typical information in data sources such as spreadsheets captures only physical inventory of the IT components or at most static dependencies, but they do not contain the workload imposed requirements and constraints.

B. *Distributed Knowledge in Enterprise Environment*

IT asset management is a set of business practices that join financial, contractual and inventory functions to support life cycle management and strategic decision making for the IT environment. The IT and business environment consists of software and hardware assets.

Most, if not all, of the information required in datacenter IT discovery and dependency analysis is collectively in possession of the infrastructure experts and users. Some of this information may be in structured and unstructured repositories managed by individuals, while other information may be entirely in the minds of these individuals. Individuals possessing the information are distributed through out the enterprise and are managing multiple roles.

Let us consider a core asset of an enterprise datacenter – a business application. In the most general sense business application refers to a high-level business functionality, such as support website, or a financial system, that is exposed to the internal and/or external end users. As such, this high level concept of an application is not programmatically discoverable. Furthermore, each application is characterized by a set of financial, contractual and technical attributes.

For example, let's take a business application "Support Website", shown in Figure 2. Business application hosting details capture the properties of the physical infrastructure, such as production, preproduction, test, and development servers (their hostnames, IP addresses, and security owners). In addition, each application requires certain middleware supporting it (e.g. LDAP, DB2, application and HTTP server), which can be programmatically discoverable. Despite the availability of discovery scripts and the availability of application and server repositories a

consolidated view of this information is missing.

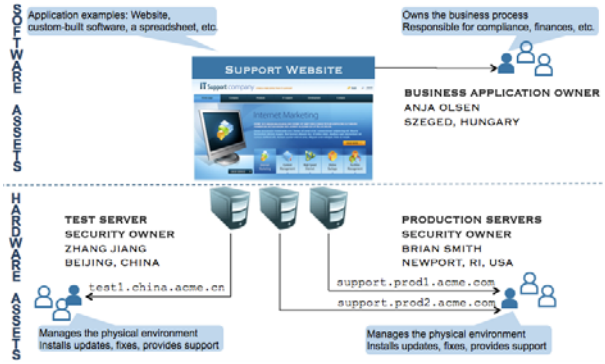

Figure 2 Application Portfolio Management - Core data elements and experts

On the other hand, from the business perspective, it is critical that the hosting charges are applied to the correct financial codes. Similarly, from governance and business controls point of view, it is important to ensure all the compliance regulations are in place for this application, e.g. that it is subject to US Traffic and Arms regulations. Security and compliance audit identifies the procedures and/or requirements for handling certain types of data. For example, security and audit may identify procedures for processing financial data. The processing of the financial data may need to be performed in a manner such that the financial data is ready for auditing. As another example, the storage and/or processing of personal and/or confidential data may require particular security procedures. These security procedures may be identified in security and audit. Lack of compliance may have huge financial consequences for the enterprise and its clients.

As visualized in the Figure 2**,** different characteristics of the application entity, and consequently of business and IT environment are distributed across the enterprise. This is even more prominent in the global enterprises, where role and knowledge are not only spread across business and technical teams, but also across continents and time zones. Experts responsible for this application, both as business or technical owners often may transition to a new role within or outside the enterprise, making it even more critical to capture this knowledge in a timely manner.

C. Pain Points of Current Practices

Because of the dynamic and evolving nature of datacenter, the discovered information also needs to develop continuously and needs to be up-to-date with sufficient confidence. Passive instrumentation based discovery and dependency analysis is time consuming, tedious, error prone, and may not scale. Some types of dependencies are implicit and cannot be easily detected via instrumentation.
Traditionally analysts would employ master spreadsheets to collect and file the data. However, in large-scale datacenters one may need to deal with tens of thousands of server systems hosting thousands of business applications. Applications are as a result typically categorized in tiers, based on their reliability and support requirements, criticality to the business and architectural complexity.

Financial codes and business control requirements change with the revisions of the governance regulations, and as a result this information needs to be continuously refreshed. Staff managing and maintaining such systems may further need to comply with certain criteria, in particular in the outsourcing arrangements. In addition, during the lifecycle of the application, eventually some of them become obsolete and sunset, whilst the infrastructure is still dedicated to them. Discovering such cases is critical to decommission and redeploy unusable server systems. Furthermore, by discovering the physical infrastructure of the business applications, aids in understanding the current topology, and requirements for migration to the cloud. Consequently to overcome these challenges there is a need for a streamlined process in gathering and refreshing data describing the enterprise datacenter, delivering increased time-to-value for various IT transformation activities.

III. KNOWLEDGE HARVESTING & INFORMATION SYNTHESIS SYSTEM

The Knowledge Harvesting & Information Synthesis System (KHISS) is designed to provide mechanisms needed to collect and correlate information from multiple sources including data repositories and by engaging online communities in an enterprise. It is a versatile system that can be adopted to different types of IT problems including IT migration and transformation. In each case, a model of the system for which the information is to be collected and synthesized is built and refined. KHISS tries to collect as much information about all aspects of the model as possible. In the process of analyzing the collected data, it synthesizes new information by creating correlations and linkages among different parts of the information and elicits additional information from subject matter experts based on the identified patterns. In the following, we first explain the architecture and design of the system and then provide an outline of how the system would operate on a real IT migration problem.

KHISS consists of following components:

- Information Discovery component
- Information Analysis and Correlation component
- Online Community component
- Quality Evaluator component
- Driver component

Information Discovery (ID) component: IT systems and datacenters evolve over time and various competencies affecting the datacenter operations – server, network, storage, power & cooling, facilities, application, database, development, compliance & security, business analyst etc. – manage and maintain information on different aspects of the

IT systems in their own unique ways. The ID component is a mashup service capable of integrating information available in structured data repositories such as spreadsheets, relational databases, directories, org charts and from unstructured information sources such as documents, presentations, e-mail and chat messages. In unstructured data sources, tabular data and lists are the primary targets, but associations may also be setup among key phrases and other types of information.

The ID component is also designed to interface with low level IT discovery tools such as IBM's Tivoli Application Dependency Discovery Manager (TADDM) that, given appropriate credentials, can automatically discover physical infrastructure components such as servers, network components, and application infrastructure components such as middleware and dependencies among these.

Information Analysis and Correlation (IAC) component: The complexity of an IT system is in the types and variations among components within the IT environment and the dynamic interactions among these components. To capture the complexity of the IT system, KHISS builds and refines a model of the entire IT system. This process starts with models of the basic building blocks such as physical servers, network and storage components, virtual machines, OS, middleware, application components, and so on. Each component has a set of attributes and a function to perform. For example, attributes of a physical server include architecture, processor, number of cores, amount and type of memory, storage devices, network adapters, vintage, firmware type and level, asset owner, administrator, location, and so on. The models and the list of attributes are extensible and may be modified as more information is collected and analyzed.

Using the models of the building block components, the IAC component analyzes and classifies the information obtained by the Information Discovery component from various sources. For example, information from server inventory may be used to indentify known servers and their attributes. Information from different sources may be overlapping, complementary, and sometime contradictory. The IAC component maintains redundant as well as contradictory information along with the source of the information. When possible, it tries to assess the age or recentness of the information. In most cases, recent information is more accurate than the older information. However, in some cases, information from a certain time period may be more relevant than most recent or older information. For example, configuration parameters used at the time a system was built and installed are more likely to determine the dependencies among components of the system than the information in a more recent inventory. In addition to identifying various IT components and their attributes, the IAC component also indentifies dependencies amount different IT components. Some of these dependencies are explicit in data sources and some times these have to be inferred. Dependencies can be dynamic and may change over time. For each dependency, a set of attributes is identified and values are assigned. A dependency could be between two software components such as a dependency between an payroll application and a database or between a software component and a hardware component such as a hypervisor and a physical server. Attributes of a dependency include the dependent and the supporting components, dependency type, location constraints, performance dependency, reliability and availability dependency, security and audit related dependency. In the case of latter case, the dependency may imply specific type of secure connectivity, encryption requirements, and logging for audit purposes.

It then creates associations and correlations among identified components. The associations may be created along multiple dimensions; e.g., all servers administered by an administrator, servers owned by the same manager, servers running specific version and release of OS, server on a subnet, and so on. Note that if all the relevant information about IT component such as a server is stored in relational database, complex queries can be run to obtain this type of information. The distinguishing feature of this component is that it forms these associations explicitly so that patterns and exceptions can be identified. In addition to associations, this component identifies correlations such as co-dependencies. For example, a web service may depend on a database and an LDAP server. Once the dependencies of the web service are identified, the IAC component then correlates specific database service instances with the LDAP service indicating that both services need to be up at the same time in order to support the web service.

The collected data is then aggregated, classified, and further analyzed for identifying patterns and outliers. Using the identified patterns, classifications, associations and correlations, the IAC component builds a model of the system being analyzed. The model is then further analyzed to determine contradictory information as well as missing information. For example, hostname or IP address of a server may appear in an inventory list of physical as well as virtual servers. This can happen because automated discovery mechanisms often cannot differentiate between physical and virtual servers. In another example, some attributes of a server may be missing. This can happen either because the inventory was incomplete or because of incorrect classification. An example of the former is the missing information about the rack location of a blade server. Missing values of any attribute of an IT component or of a dependency are all examples of this type. An example of the latter is the incorrect classification of print or a storage server as a regular server. Incorrect classification leads to association with incorrect attributes which may not

have valid values. In both cases, this is labeled as missing and incomplete information that needs to be resolved.

Online Community (OC) component: Validation and refining of the system model and resolution of the missing and incomplete information are accomplished with the help of the SMEs and other users who are knowledgeable about the system. This is accomplished by the OC component, which starts with the system model, the information used to build the model, and the identified missing information components. Using the current and past information on asset owners, administrators, and users of relevant IT components, the OC component, creates a graph describing the relations between IT components and various human actors. The graph also shows the organizational relationships among the human actors. Using the available information on role, skill level, relevancy to a particular IT component, it then assigns a knowledge weight for each actor that has a connection to an IT component. Higher the weight associated with an actor, the more relevant and more valuable is the information provided by the actor for that IT component. This constitutes the creation of the online communities for the purpose of soliciting targeted information from targeted community.

After establishing the relationships between IT components and human actors, the OC component tries to (a) validate and refine system model and (b) resolve the missing and incomplete information. Soliciting missing information and asking for validation of the existing information and the constructed models accomplish this. Email is sent to targeted users along with a customized URL that they can use to access and view the part of the model that is most relevant to their expertise. For example, the URL sent to a server system administrator may display rendering of the server along with its attributes showing known and missing values. The rendering may also show IT components supported by the server and internal and external dependencies among the IT components. The users are asked to view, validate, and update the information shown to them. They are also asked to identify any other human actors who may have other relevant information related to that or other IT components that are part of the system model. Using the input from the user, the available known information is updated and the IT model is refined. If information provided by the user is contradictory to what is known from other sources, the weight factors associated with the sources is used to create a confidence factor for the conflicting pieces of information. The information with the highest confidence value is used to build and update the system model. The OC component also follows up with the referrals provided by a user to other actors as potential source of information. It checks to see if those actors have already been contacted. It also evaluates the relevancy of the suggested actors. In some cases, it may ask another source in the community to assess the quality and benefit of interacting with the new contact. This process iterates until all the sources are exhausted or until there is sufficient confidence in the information collected and in the model developed.

Quality Evaluator (QE) component: The QE component works in coupled manner with OC component. As the input is received from users, the QE component evaluates the input and assigns a confidence score to the input. Among other factors, the confidence score is based on an assessment of the degree of familiarity of the user with a specific aspect of IT system. It also takes into account referrals and recommendation of the user by others for that topic. In addition, when confidence is low, the QE component may continue to seek further input from other users until sufficient confidence is achieved. In some cases, the confidence may not reach high and in such cases that information and other derived from that information are flagged as having lower confidence. In the next section, we describe the details of the concepts used for quality control.

System Driver (SD) component: The SD component allows the system to be configured and tuned for solving specific type of IT or other problems. The SD component can define models for basic building blocks of the system, define attributes, provide default values, and also may define rules to be used for searches, for confidence and validation purposes, and so on. For example, an SD component can use the system to build knowledgebase for transformation analysis, where as the same system can be used by another driver to perform maturity analysis. The SD component defines how the system can get started, how it can build on the acquired knowledge and can decide when it has sufficient information to make intelligent choices. In other words, the SD component makes the system “smart”.

IV. USE CASE: APPLICATION PORTFOLIO MANAGEMENT

A. Our Approach

We recognize that the first step before performing any complex operation such as IT transformation or optimization is to construct an accurate model of the current state of the system and that the information required for constructing the model is usually diffused within the organization. As discussed earlier, traditional approaches tend to be expensive, time consuming, tedious and error-prone. To address the pain points in cost-effective and efficient manner, we have developed a novel approach that combines and correlates information available from traditional sources with collective knowledge in possession of employees, users, and other individuals who interact directly or indirectly with the affected IT processes. The basic premise of our approach is that recorded information in structured and unstructured information repositories often complements the information known to the employees in the organization who use, administer, or operate the systems and processes. Our approach tries to bring the diffused information together by making intelligent information mining, analysis, and synthesis using contextual guidance to build and assess the quality of knowledge.

We assume that information available from individual sources can be incomplete, ambiguous, and sometimes

contradictory and that enterprise processes are never stationary. We also assume that IT processes do not operate in isolation, but they are closely tied to business processes and have owners, users, administrators, are associated with IT resources and services. At the core, our system consists of knowledge harvesting and assessing the confidence in the acquired knowledge. Knowledge is harvested from various sources and is correlated. Information in possession of employees and other individuals is actively sought, but only after asserting a certain degree of confidence in the associating the information with the individuals. For this the system builds and collaborates with online communities formed of enterprise actors associated with IT and business processes. The communities are formed based on high confidence links for specific purpose and are used for information collection, synthesis, and validation. Since information is not fully trusted and is assumed to be dynamically evolving, every piece of collected and synthesized information is associated with a confidence score.

At the high-level our approach is to collect information from data sources, identify structure, classify, gather and create metadata about types of information, correlate, identify structure, identify gaps, seek information from correlated actors – users, administrators, owners, service providers – synthesize information, and assign confidence. The system described here is designed to allow other tools to be built on top as driver for specific purposes. Such driver can provide specific context, define boundaries of the knowledge domain or specify rules if they exist. It can also serve as a mechanism to bootstrap, define processes, define participants, and define confidence functions. Coupled with the driver the system can be used to perform routine tasks such as subject specific queries as well as to perform complex analytics such as continuous optimizations.

B. Identification of Missing Information

IT optimization processes require an in-depth insight into the enterprise datacenter structure, operations, and resource utilization. For example, when considering a migration to Cloud environment, one needs details on which applications are most critical, of most value to the enterprise, and would benefit the most of the Cloud-environment? From the technical perspective this translates to understanding the physical infrastructure hosting these business applications.

In shared environments, this implies detection of other applications that are co-hosted on the same servers. Who are the security owners of these machines? Before application is moved, both business and technical teams need to be engaged. What are the compliance regulations and requirements for this application? Once the application is in the Cloud, certification and compliance needs to be continuously ensured. Such rare pieces of information may not always be stored in the data repository and are commonly owned by various experts.

To trigger various knowledge management processes, we form the target experts by obtaining the "seed" of the community from existing enterprise directories, such as application or server repository. In other words, from the current data repositories, which enlist role ownership, we obtain list of e.g. network administrators, application and security owners, etc., to be included in this process.

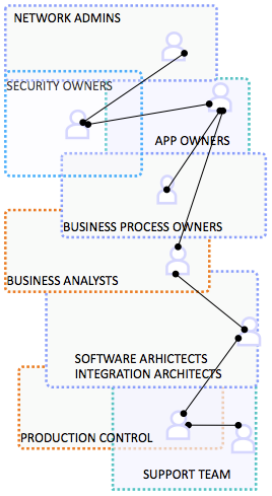

Figure 3 Experts in enterprise datacenter management

Figure 3 shows a variety of experts present in the enterprise datacenter. It is not uncommon that roles and responsibilities may overlap, e.g. Business Process Owners may perform business analysis on feasibility of application migration that is embedded in the process they own. The connections in the figure demonstrate the involvement of different roles and experts at different parts of the large-scale business transformation process, such as on-Cloud migration. For example, the transformation activity starts with understanding the physical infrastructure and this is where network administrators and system administrators get involved. Application owners provide information about the business capabilities of the discovered systems. Business process owners together with business analysts provide input on the feasibility and criticality of the move the selected application on Cloud. Finally software and integration architects work on designing and implementing the move, providing scripts and so on, whilst production control and support team ensure that cloud-enabled application is running smoothly.

C. Identification Information Sources

The identification of the components within the enterprise datacenter may be obtained from different sources, such data repositories, spreadsheets, and/or databases. These sources may contain complementary information pertaining to specific components. For example, server registration

database contains server purpose, current security and administrative owners. Additional information may be obtained through automated discovery processes, such as middleware discovery tools.

D. Incentives

Engaging experts from known enterprise communities bootstrapped the data collection process. When relying on online communities, the critical question is how to build and maintain such a community over time? Material and social incentives can make all the difference between success and failure. From the HR and business control guidance point of view, such an activity needs to be clearly specified as either as part of the regular work item or a voluntary activity. Another challenge in providing an incentive to in-house employees arises due to the different contractual arrangements. For example, would contractors be allowed to charge their contributions as additional hours?

In our prior work [1][2], we describe detailed results from engaging online community, through social incentives, to collect the details about physical infrastructure that is hosting business applications. For each contribution users earned virtual points, which were used to create a user ranking. No tangible reward was provided in exchange for these points. The points were only exposed to one group of users. We found that point system had no impact on user participation. User's reported that having access to collected data was a sufficient motivation for participation.

E. Quality Assessment

How can we rely on the data collected through collaborative efforts? Quality is a key factor when evaluating the value proposition of engaging communities, in particular for business objectives that are on the critical path. Existing approaches to quality range from automated selection algorithms to engaging the crowd for reviewing process. Most commonly employed mechanisms include majority voting and aggregation of contributions [3][4].

Figure 4 shows that experts have some common knowledge. This provides us with the opportunity to cross verify the data collected. For example, data provided by the application owner about servers that host its application, can be verified by asking the server owner to accept or reject the proposed knowledge contribution. Figure 4 shows how confidence level of contributors and verifiers is tracked.

F. Continuous Validation and Information Synthesis

This section describes the process of continuous data collection and verification. It begins by creating a community based on the hardware and/or software components in a IT system. For example, community includes people having roles associated with the hardware and/or software components in the IT environment. More specifically, it may include people with roles, such as, for example, the owners of software applications, security owners, network administrators, production controllers, and other types of roles. Further, a number of roles may be associated with a particular component.

Thereafter, the process assigns a confidence level for each person in the community. The confidence level for a person may be based on a number of factors including, the role, the experience, skills, duration of time for which the person has the role, and/or other factors. Further, the confidence level of a person determines the quality of information that may be provided by the person.

The process then selects a number of components in the IT environment. The process identifies individuals with roles associated with the selected IT components. When forming a community, querying a database of experts for the number of components in this illustrative example may make this identification. In other illustrative examples, the identification may be made by selecting a person with a highest confidence level in a group of people having roles associated with a particular component for each component in the number of components.

Application	ID	Compliance	Application Owner	Validated By	Status	Confidence Level
Support Website	SW001	Y	Jane Smith	Peter Wilson	ACCEPTED	100%
Internal Messaging	IM002	Y	Josh Thompson	Tim Johnson	REJECTED	50%
Department Planner	DP003	N	Anna Peterson	Norman Miles Susan Ross	ACCEPTED	75%

Figure 4 Cross expert knowledge verification

Thereafter, the process sends a number of requests for missing information to the identified individuals. The process then receives responses from those individuals identifying the missing information. The process then updates the confidence levels in the individuals sampled in the community, and then the process terminating.

Enforcing quality assurance levels, and continuous verification, this allows the system to continuously scan the data model and discover a) low confidence b) unverified and c) missing data. As a result system triggers tasks for these items to be completed and verified.

V. RELATED WORK

This section provides an overview of existing applications of social networking within the enterprise, crowdsourcing platforms, and their usage within the enterprise, thereby setting our system in the context of the state of the art. Online communities are increasingly leveraged within enterprises, primarily for knowledge sharing and discovery of experts for specifics tasks. In addition, social networking sites are often designed to support employees within an enterprise in connecting and learning about each other through personal and professional sharing [5][6][7][8].

Crowdsourcing [9], a successful mechanism to harvest information and expertise from the online communities has

evolved over the past few years from its humble beginnings as isolated purpose-built initiatives, such as Wikipedia[10] and Mechanical Turk[11] to a growth industry. Enterprises are extending their business processes to leverage, often low-cost, scalable workforce in online communities. Examples can be found across the product and service lifecycle, from early design stages, where communities are engaged to submit ideas on new features (e.g. Dell's IdeaStorm) or products, to development (e.g. TopCoder), provide product support (e.g. FixYa, CrowdEngineering [12]), to solving business and research challenges (e.g. InnoCentive). As a result, numerous enterprise crowdsourcing models emerged, based on the type of the requested task, type of crowd/community engaged (e.g. internal, external, or hybrid), and type of the incentive, to name a few [13]. Aside from employing the external online communities, enterprises are also engaging internal online communities to execute various business transformation activities, ranging from collection of specific knowledge points [1], to maintaining support knowledge bases, as well as introducing dynamic internal support marketplaces [2]. In contrast, we describe a system for collecting and managing the lifecycle of the datacenter by leveraging online communities, and discovering the changes in t he same.

VI. Discussion and Conclusions

In this paper we described KHISS system, designed to provide mechanisms needed to collect and correlate information from multiple sources including data repositories and by engaging online communities in an enterprise. We summarize lessons learnt and managerial implication from deploying the KHISS within a large enterprise for application portfolio management, as the first step in on-Cloud migration effort.

It is critical when engaging the online communities for large business transformative activities to understand what is the value for participants to contribute. What is the encompassing effort? Another key learning from the deployment of our system is that the required task has to be put into the larger business context, so that participants can understand how their contribution affects the operators, and secondly, providing time estimates for task completion (i.e. cost of participation). Furthermore, by making progress of other participants transparent, this further emphasizes the community aspect of the effort.

While in our pilot deployment we only considered social rewards, one has to evaluate the suitability of material rewards for a specific task. Furthermore incentives can be provided for individual and group activities. For example, in our scenario, we could have awarded everyone additional points when a set of applications is completed. In the long run there is the question of how to retain high performers and their contributions, as the knowledge acquisition and verifications are continuously triggered.

Finally, it is key that the e-mail notifications sent with the task requests are highly personalized, so as to reduce the possibility for being disregarded by the users, or mistaken for spam or bot-mails. Furthermore, having a set of controls embedded in the e-mail, to allow for user to directly act upon the request helps remove a level of indirection and potential loss of users' attention, when required to switch between the email and information gathering system.

Our next steps will involve deploying KHISS through entire phase of on-Cloud migration process, thereby evaluating system end-to-end. Our focus will be on identifying the minimum level of data quality required to execute the transformation, while evaluating both ad-hoc and automated quality assurance mechanisms.

References

[1] Polychronis Ypodimatopoulos, Maja Vukovic, Jim Laredo, Sriram Rajagopal: Server Hunt: Using Enterprise Social Networks for Knowledge Discovery in IT Inventory Management. SERVICES 2010: 195-196

[2] Mariana Lopez, Maja Vukovic, Jim Laredo: PeopleCloud Service for Enterprise Crowdsourcing. IEEE SCC 2010: 538-545

[3] Sorokin, A., Forsyth, D.: Utility data annotation with Amazon Mechanical Turk. In: Proceedings of the Conference on Computer Vision and Pattern Recognition Workshops. IEEE Computer Society, Washington, WA, USA. 2008.

[4] Kern, R., Thies, H., Bauer, C., and Satzger, G. Quality Assurance for Human-based Electronic Services: A Decision Matrix for Choosing the Right Approach. In Proceedings of First Enterprise Crowdsourcing Workshop in conjunction with ICWE 2010.

[5] Mark Brodie, Jennifer Lai, Jonathan Lenchner, William Luken, Kavitha Ranganathan, Jung-Mu Tang, Maja Vukovic: Support Services: Persuading Employees and Customers to Do what Is in the Community's Best Interest. PERSUASIVE 2007: 121-124

[6] Huysman, M. and Wulf, V. IT to support knowledge sharing in communities, towards a social capital analysis. *Journal of Information Technology*, *21* (1). 40-51

[7] D. G. Bobrow, J. Whalen. *Community Knowledge Sharing in Practice: The Eureka Story*. Journal of the Society of Organizational Learning and MIT Press, Volume 4 Issue 2, Winter 2002

[8] JM DiMicco, W Geyer, C Dugan, B Brownholtz, DR Millen. (2009) "People Sensemaking and Relationship Building on an Enterprise Social Networking Site." Proceedings of the 42nd Hawaii International Conference on System Sciences (HICSS '09), January 2009.

[9] Darren Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. Convergence: The International Journal of Research into New Media Technologies, 14(1):75–90, 2008 .

[10] Kittur, A. and Kraut, R. E.. Harnessing the wisdom of crowds in Wikipedia: quality through coordination. In Computer Supported Cooperative Work, 2008.

[11] A. Kittur, E. Chi, and B. Suh. Crowdsourcing user studies with mechanical turk. In Computer Human Interaction, pages 453–456. ACM, 2008.

[12] G. La Vecchia, and A. Cisternino. Collaborative workforce, business process crowdsourcing as an alternative of BPO. In Proceedings of First Enterprise Crowdsourcing Workshop at ICWE 2010.

[13] M. Vukovic, Claudio Bartolini: Towards a Research Agenda for Enterprise Crowdsourcing. ISoLA (1) 2010: 425-434