Integrating Enterprise Architecture with CMDB/ITIL

João Carvalho [1], Miguel Silva [2], Pedro Fernandes [3]

1) Instituto Superior Técnico, Lisboa, Portugal

joaomfcarvalho@ist.utl.pt

2) Instituto Superior Técnico, Lisboa, Portugal

mms@ist.utl.pt

3) Inovaprime, Lisboa, Portugal

pedro.fernandes@inovaprime.com

Abstract

Today's organizations have ascertained a fierce reality: the changes in their business are so quick, that in order to keep itself competitive, they must base their business processes in a complex IT infrastructure. This dependency growth, had led to more complex systems and tools, which, sometimes, overlap the functionalities of each one. Configuration Management DataBase (CMDB) from ITIL and Enterprise Architecture (EA) tools are such an example. An integration between them is imperative in order for faster implementations, better aligned EAs and reducing the efforts of manage the same data in two different repositories. In this paper we present the research area in which this subject fits in, an extensive description of such problem and our proposal for this integration.

Keywords: CMDB, Enterprise Architecture, ITIL, Integration, CMDB tools, EA tools, data, Configuration Management

1. Introduction

In a world that is in constant change and evolution, organizations must rely on IT services to address their needs and those of their clients. Stakeholders need faster and improved methodologies to deliver its services or products in the most cost-efficient way possible. This can be achieved by automating different processes and services, reducing cost in human labor and malfunctions in the delivered products and services. Such automation will make the organization's IT infrastructure more complex and difficult to manage each day. Requests for Change (RFC) can be very frequent and need to be answered with the highest brevity, without disrupting the services that rely on this infrastructure.

To do so, it is necessary to monitor all IT assets, and align them with the business so that they deliver the maximum business value. This, in turn, can be accomplished with the implementation of a Configuration Management process as the one proposed by ITIL's best practices. "Configuration Management covers the identification, recording and reporting of IT components, including their versions, constituent components and relationships"[OGC, 2006]. Such a process is supported by a Configuration Management Database. "The CMDB helps IT managers focus on business services as well as IT technology by providing a comprehensive view of the IT environment" [Hinich 2007]. This, along side with other services from ITIL, will allow "organizations to efficiently and reliably manage services and to satisfy performance, availability, and cost objectives" [Johnson 2007].

However, having a well defined IT infrastructure and operating according to all the best practices does not mean that the infrastructure is configured to achieve all organizational goals and strategies in the most efficient fashion. An organization may begin to follow a specific strategy, aligning its IT infrastructure to accomplish its objectives, but when the market changes, and with it, the customers needs and preferences, a change of strategy is pressed upon the organization. Does this affect the IT that supports most of the organizational services? In most situations the necessary adjustments are made to ensure normal operation is possible and everything else is kept unchanged. These solutions are mostly based on the aquisition of new hardware and software to patch each specific new requirement, disregaring the current infrastructure and how it fits in the new scenario. To avoid these pitfalls, and achieve a better mapping or alignment between organizational strategy and IT, the design and implementation of an Enterprise Architecture becomes imperative. Enterprise Architecture is defined as "a coherent whole of principles, methods, and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure" [Lankhorst e al. 2005]. Because "virtually everyone's job requires data, and access to data is the most frequent request" [Spewak e al. 1993], one major role of this methodology is to gather all information about the organization's assets, business processes, applications and other additional information.

When these two concepts coexist, i.e., an organization that is planning to design is own EA already has a CMDB implemented, in order to support ITIL's processes, organizations face a data redundancy problem. "Developing and populating enterprise architecture models is often the most time consuming part of the enterprise architecture development activity" [FEAD 2007]. So, besides the waste of time that filling in two different, but partially overlapping, repositories with organizational data can represent, there is also the difficulty of tracking the evolution of Configuration Items, and the changes they go through while the EA is being developed. This might result in a current model (As-Is), that does not match with the actual state of the organizational resources.

To address these issues, we present our proposal for an integration between the tools that support these two approaches. In our solution, and because "basing solutions on standards helps achieve goals sooner and with less risk" and "facilitate the interoperability needed to connect internal and external applications and data" [Johnson 2007], we will use the approach envisioned in the federated CMDB [Clark e al. 2007] to establish a bridge between these tools.

The next section will present the research area in which this paper falls in, followed by a deep description of the problem, our proposal and a final evaluation of the presented solution.

2. Research Area

CMDB

ITIL is "the most widely known and used process framework for managing IT services and infrastructure" [Johnson 2007] "has become the world-wide de facto standard in Service Management" [OGC 2006]. The need for such framework comes from "the recognition that organizations are increasingly dependent upon IT to satisfy their corporate aims and meet their business needs". One of the key processes of ITIL is the Configuration Management, which in V3 is presented as SACM (Service Asset and Configuration Management). Here is where all resources of the organization, baptized as Configuration Items (CI), are stored and kept track of. The tool responsible for this storage is the CMDB (Configuration Management DataBase).

CMDB is a fundamental component of SACM and therefore a great enabler of IT Service Management, because most of the other processes defined in ITIL rely on this repository to retrieve and store the needed information to accomplish its own responsibilities. A CMDB is seen as the central repository (or set of repositories in the case of a federated CMDB [Clark e al. 2007]) for organization's CIs and the relationships between them. These CIs in the eye of ITIL were primarily thought to be only IT Assets, but with the evolution of ITIL and the lack of standards and well defined best practices for CMDB implementation and data model structure definition, this concept was widened and used by CMDB vendors to also represent other kinds of assets like physical assets and organizational units.

CMDB Tools

Much of the CMDBs found in the market extend the definition of such system far beyond ITIL's definition. Even with the already spoken absence of definitions in a CMDB implementation, reading the ITIL's definition of CMDB, this tool is no more than a relational database with a special structure that allows the definition of elements with attributes and relations between each other. However, business demanded for a more robust system with interfaces for insertion of CIs, integration with other tools, modules for drawing the data stored according to some rules and the ability to define those rules and enable different views of the organization. Gartner enumerates five functions that are needed to make the difference between a CMDB and a Configuration Database: Reconciliation, Federation, Mapping and Visualization, Synchronization and Access Controls [Colville e al. 2006]. These functions are all found in top market CMDB tools and will be described later.

The context in which this system was designed, the ITIL's configuration management process, makes this tool mainly an IT initiative used to define and maintain the state of the art of the IT assets of one organization. With the matureness of this kind of tools and the people who really use it, vendors started to support more than just IT Assets, including also users responsible for activities or business processes, which ahead, will be supported by some applications, which, for instance, runs over a machine. This kind of overview, enables the power of an IT manager to be able to know the exact impact of a machine's change in the business processes that it supports. To answer this kind of advanced requisites, vendors chose to add specific modules to theirs CMDB, each responsible for a particular role, creating a whole tool capable of create views, open branches for testing some configuration, versioning, generate audit trails and integrate with other systems/tools through modules for data import and export.

Generally a CMDB is a tool developed for IT personnel. In terms of organizational position, it should be owned by the group responsible for the corporate back end systems (like HR and Finance systems). The responsibility of maintaining the structure of the CMDB and deciding between possible changes to CIs and their structure should be an self organization IT Service Management (ITSM) department or external organization that will act as this department.

CMDB's Functionalities

The major responsibilities of this tool can be described as: keeping track of changes in all organization's CIs and relations between them. The real business of this tool as a standalone implemented with only the requirements of ITIL's specification is quite small, but vendors tried to fight against this by adding some more functionalities that transformed a CMDB in something more powerful with advantages like being an unique entry point to all the CI's the organization, relations between them and the state of each one. From all of the functionalities added, only some of them were described by studies from Gartner [Colville e al. 2006] as fundamental to a CMDB success.

In Table 1, is a list of functionalities implemented by top market leaders in the CMDB area. Because CMDBs are most likely products directed to IT public, vendors have reported bottom level functionalities that are then mapped here, to top-level functionalities at the same level of the ones reported by EA vendors.

Some of these features are presented as automated because this would be the perfect case. Even though, actually much of them need human intervention like reconciliation, that besides present, isn't as general as defended by Gartner [Colville e al. 2006].

The CMDB's core with an interface for insertion, deletion and update of the data stored is responsible to support features 1, 2, 3 and 4. All the others are normally presented in separate modules for visualization, reporting and backoffice operations.

Enterprise Architecture

"An enterprise architecture (EA) is a coherent whole of principles, methods and models that are used in the design and realization of an enterprise's organizational structure, business processes, information systems, and infrastructure" [BMC 2005]. EAs are a way to align IT, human resources and organization's processes with the business strategies. According to Lars Nordström "Enterprise Architecture is a model-based approach to business-oriented IT management" [Lankhorst e tal. 2005], i.e. EA defines a model of the current state of business IT, personnel, and business processes aligned with the organization's objectives and strategies.

These objectives are achieved through the development of an architecture framework to define a series of reference architectures known as "as-is", "to-be" and "migration plan" models. The "as-is" model is related to the current state of the organization, the "to-be" model is to where the organization's current architecture should migrate to and the "migration plan" defines how to do this migration. These key architectural snapshots are also known by the terms "current", "intermediate" and "target" reference architectures.

These models and diagrams need to be stored and managed allowing stakeholders to have the better decision-making possible. This role is attributed to EA tools that besides the features for designing all diagrams and models have also a repository module responsible for these tasks related to data management.

EA Tools

According to Forrester, the EA tools are divided in three main categories: modeling tools, repositories tools and change management tools [Peyret 2007]. These categories are strictly related to one of three EA approaches: top-down, bottom-up and change management respectively. Vendors are focusing mainly in one of these categories but the need for adding more and more features is making the line between these categories becoming blurred. Also with the growth of interest of IT executives and business personnel in these tools, vendors increased the offer of collaborative and change management features which reflected in an increased interest in data models [Peyret 2007]. This run for features was a necessity that vendors found as response to the growing of EA stakeholders that use the tools. Each actor seeks in the EA tool for answers that help him in some decision-making: response to a catastrophic crash in some machines of the infrastructure; preview the future and the changes needed in the organization to support expected problems; performance of business process and workload allocated to some organizational unit or key user in the organization. Depending on the user, the system can present him with special pre defined models and diagrams that most adapt to his necessities and needs.

Bottom Level Functionality	Top Level Functionality Enabled	Description
1. Reconciliation	Data Quality and CI unicity	Should provide ways to prevent the creation of redundant data like the import of two CIs that are in fact the same element but that came from different sources. This feature is responsible for the data quality in the CMDB, which is commonly spoken as the Achilles' heel of this tool. Actually, there are situations where only people can decide whether an introduced CI already exists in the CDMB or not.
2. Federation	Allows specific data stores for example to each architecture in an EA	Having relationships to CIs that aren't stored in the own CMDB but in an external one. This enables a better management of all CIs in an organization as it allows the construction of domain specific CMDBs with less complexity than the general one.
3. Access controls	Data Security allowing only authorized personnel to access it	Ensure that only appropriate roles or actors have access to data both for read and/or write.
4. Versioning	Access to the evolution of the CIs through time	Allows the presentation of all states that a CI have passed since was created till now.
5. Mapping and Visualization	Visualization capabilities	Allows a spatial view of the relationships between CIs and its attributes. It's also necessary that this layer of the CMDB allows the definition of different views of the data. Being a dashboard of the organization in terms of its CIs is important to allow that different stakeholders can access to views that really mean anything to them.
6. Audit Trail Capabilities	Reporting Capabilities	Should be possible to in any period of time, to produce an audit trail report with who did what and when in the system, allows the awareness of exactly who was responsible for a certain change made in the system.
7. Synchronization	Integration with other tools for simulation, test changes and different configurations	Being able to update the CMDB with changes that were in study. This is basically a structure of branching for the data stored in the CMDB and, merging and reverting capabilities enabling the creation of alternate branches for test changes and then commit them to the baseline. Once an inappropriate change is detected, a notification to change management workflow should be triggered, in order to alert the responsible for the IT domain where the change will take place, to remediate the situation. This test changes' branches could be produced in and external tool, so the CMDB need to have a way for inserting data in the system.

Table 1 CMDB's functionalities table

The definition of the types of users that interact with this tool and their needs in it, is a job that depends from the organization's structure, and so, it's possible that are situations to which isn't yet defined a specific view for an actor requisites. Even so, there are some actors that are generally present in any organization such as managers, CIOs, CEOs, software developers and business process designers [Lankhorst e al. 2005]. For example, managers, CEO's and other business personnel are seeking for top-level diagrams with aspects about organizational architecture, process architecture and possibly information architecture and CIO's and IT personnel will be more interested in the technical and applications architectures.

The challenges for today's EA tools rely on deliver tools and methodologies to design diagrams that best suit all the stakeholders' needs. These methodologies and design techniques were already discussed in other researches [Soares 2007] and are complex enough for a stand-alone tool. The data repository feature is just some another subject to deviate EA tools vendors' attention and efforts. "Stakeholders define the needed diagrams not the architects" [Lankhorst e al. 2005] so these tools need to be the most general possible providing instruments to define the diagrams and not different diagrams already designed. What functionalities do these tools provide nowadays will be described later on.

Enterprise architecture is beginning to change from a methodology focused only on IT users to other business areas and business staff. This change was introduced by the presentation of predefined different organizational roles outside the box in EA products [Peyret 2007]. These different roles define the menus, views and models available for each user. There are three main roles that present the information from the system in different ways, such as personnel inside the EA team, IT personnel outside the EA team and business personnel outside the IT area.

These tools are responsible for enabling the best possible alignment between organizational objectives and strategies, with the IT that supports the organization. To accomplish this, it is necessary that they allow the automation and definition of rules that specify certain business restrictions and quality parameters to be maintained in the EA definition. It should also allow the redefinition of certain part of the EA due to change in the scenario where the organization acts and analyze the impact of different responses possible in order to decide for the best possible.

EA Tools' Functionalities

Using the three categories of EA tools defined above, we realized that these tools don't differ as much from one another as it was supposed to. So the functionalities table was unified to all tools. Table 2 presents a list of key functionalities that almost every customer seeks in EA tools [Peyret 2007].

These top level functionalities hide more operational features that these tools provide, allowing, as it was already described, vendors to answer to all new stakeholders' needs. It's important to map all features that a general EA tool must have if we want to know if our solution will be able to maintain all unchangeable. According to FEAD [FEAD 2007], there are more a few key features that distinguished these tools and are of extreme importance when deciding what tool to buy. These functionalities are:

1. Methodologies and Models;
2. Model Development Interface;
3. Tool Automation;
4. Extendibility and Customization;

5. Analysis and Manipulation;
6. Repository;
7. Deployment Architecture;
8. Costs and Vendor Support.

Functionality	Description
Modeling capabilities	Should be possible to define all kinds of views and models according to different standards for each part of the EA and the alignment between them. Should be possible to define specific models for different stakeholders.
Simulation and analysis	These tools must allow simulations to the different architectures defined and their impact in the organization. Should allow reporting and analysis of the EA actual state.
Life-cycle management	Should support life-cycle management as well as change-management processes in line of ITIL's change management specification.
Publishing	Should allow the publishing of information for viewers in their most valuable diagrams and models.
Templates	Should provide a vast set of models for general and industry-specific implementations and support the most available frameworks possible in order to save time in implementations where the general cases are used.
Product Architecture	The own product architecture should be robust enough and scalable to be installed in larger organizations where the requirements demand for large data management and model designing.

Table 2 - EA tools' functionalities table

The first point is related to the Templates functionality spoke above in the table and determines what kind of methodologies, models and approaches a tool support. This point is also related with different methodologies for data modeling and the integration between all the methodologies and models supported. The second point is the key functionality of an EA tool. The better the interface is, the better quality it can enable to the EA architect in his design work. Lots of researches already sought for the better interface for each architecture and so an EA tool is better if it uses the best practices for each domain.

Tool Automation is related to the ability to define scripts and macros to faster populate the repository and certain common actions in models definition. The fourth capability captures how well an EA tool can be modified to meet certain uncommon and organization specific architectural requisites. This capability might also be related with how the information is structured. A good EA tool should provide ways to structure information through specific metamodel definition for each implementation. The fifth point is responsible for analyzing and manipulating the designed models of the organizations' architecture. This analysis may only confirm the correctness of the models to also apply defined rules that represent scenario restrictions as it was already described in the above responsibilities section. There are some kinds of analysis present in such kind of tools like gap analysis, impact analysis, etc. Repository functions are related to the way that models and organization's data are maintained and managed. Some tools provide also versioning and revert (or rollback) capabilities, the ability to import information from relevant sources or to create that information within the tool, as well as to export information to facilitate stakeholder's use. Deployment architecture describes the real architecture of the tool (client/server, web, desktop version, etc) and the eighth point the cost of

the tool, which is out of our scope for this paper. The modeling frameworks supported by a tool is also a key factor of EA functionalities.

Similarities between CMDB and EA tools

Among the functionalities presented, some gave us enough background for our solution. First the Mapping and Visualization CMDB module enable an overview of the organization actual state, something close to what an EA as-is model should be. In addition, if this module has the ability for the definition of views, restricting the data shown to some particular area, one has the possibility to define architectural views like the ones found in EA specification. Second, the Synchronization module enables the creation of testing branches used to simulate different configuration's strategies and test its impact in the organization's structure. Both EA modelers and CMDBs can generate views of organized data in different layers depending on the stakeholder that is using the tool.

As for data and its quality, both tools have strong focus in repository definition and management. Both CMDBs and EAs have versioning capabilities and a different metamodel specification (static or dynamic). Both focus on providing the best support for change management. CMDB's role in this subject is strictly related to data storage being the versioning and the branching functionalities, key factors for analyzing the past of the organizations' CI's and simulate hypothetic configuration in response to some expected change, while EAs provide valuable functionalities to help in this process, like simulation features. Both attempt to define the organization's state of the art of the relations and interactions between its elements (IT assets, business processes, structural information entities, etc). Both try to have the greatest return possible from any IT investment.

Functionality	Description
Simulation and basic analysis	These tools must allow simulations to the different architectures defined and their impact in the organization. Should allow basic impact analysis of change.
Life-cycle management	Should support life-cycle management as well as change-management processes in line of ITIL's change management specification. Both tools have a focus in this functionality being the CMDB the one that more complexly implements this support.
Publishing and Visualization	Should allow the publishing of information for viewers and visualization capabilities in order to deliver views and models of the stored data. Both implement this feature despite in a different way. CMDBs only show basic views of the data when EA tools have advanced functionalities for draw the diagrams and present them.
Reporting Capabilities	Should be possible to in any period of time, to produce an audit trail report with who did what and when in the system. This allows to now exactly how was responsible for a certain change made in the system. Other kinds of report are equally normal to find in both tools.
Access controls	Ensure that only appropriate roles or actors have access to data both for read and write.
Product Architecture	The own product architecture should be robust enough and scalable to be installed in larger organizations where the requirements demand for large data management and model designing.

Table 3 - CMDBs and EA tools common functionalities

3. Problem

Organizational data is the main source for the Enterprise Architecture design process. This data gathered about the organization structure will influence the final design achieved and all the needed alignment between different architectures. Being a very influent part of the EA specification it is important that the data used in the different architectures reflect as much as possible the organization's reality. This cannot be done if the EA repository can't reflect the evolution of the organizational CIs, while the process of designing and implementing the architecture is in progress. Additionally, it is not manageable to have organizacional data in two different systems. In times where the convergence to standards is the mote, it is critical for organizations that seek to follow standards and existent, tested and accepted best practices, to minimize the cost of a high volume of data management. It is a strenuous task to reflect changes and the evolution of CI's into the EA's repository automatically, when it is already filled up, because a CI might have been changed in a way that influences the architecture design itself.

As we saw in the previous section, there are some functionalities that both EA and CMDB tools deliver, making it beneficial for organizations to make these tools come closer. We present our proposal for the distribution of competencies of both tools, in such a way that the role of each one, is defined the best possible way.

4. Our Proposal

As seen above, althought there is a lack of specifications about what exactly a CMDB is, it is clear that it is not an EA tool. We based our approach on a paradigm already implemented in most of the CMDB tools in the market, as well as in the federated CMDB vision [Clark e al. 2007], that considers the CMDB as the data management core which then provides the necessary information for external, analytic and processual modules, as shown in Figure 1. To allow the CMDB to be integrated with other tools, that support other ITIL processes, rather than the Configuration Management itself, the federated CMDB presents a core CMDB, with an external interface for other tools to pull information from. "The data consumer interfaces will expose data retrieval operations such as query and subscription" [Clark e al. 2007] which will allow us to retrieve information from there and be informed whenever a data change occur (Figure 2).

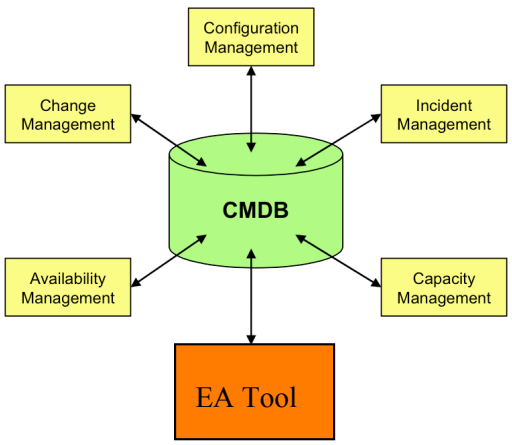

Figure 1 - CMDB as the central repository for other ITIL processes [Clark e al. 2007] and also to the EA tool

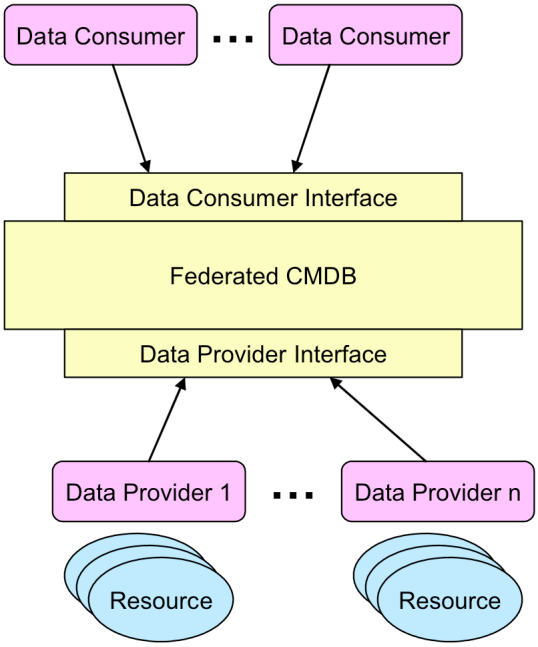

Figure 2 - Federated CMDB model proposal [Clark e al. 2007]

We propose that the EA tool will not be a simple data consumer of the CMDB, but a part of it, a CMDB module, because it will be responsible for some of today's CMDB functionalities, augmenting the strength of this relation. As a result, the As-Is model will be easier to achieve, as part of it is already present in the CMDB. The To-Be and Migration Plans will be kept in specific branches of the CMDB and iteratively refined until the moment they are made effective in the As-Is model, by merging the differences to the trunk.

CMDB's Data Model

The CMDB's domain model can either be static or dynamic. Assuming the usage of a static ontology and using the approach presented in "Integration of IT Service Management into EA" [Braun e al. 2007], we find a mapping between the ontology that supports the ITIL processes, and the different architectures, as presented in Figure 3.

Because compliance to standards is a key driver for today's solutions [Johnson 2007], we will base the initial domain for our CMDB and EA tool, on the previously described model.

Organizational impacts of this solution

Achieving the mapping between data from the ITIL and EA models, as the one presented in the last section, we enable an EA design better aligned to reality. This is true because the data is managed by a broadly implemented and tested process – the Configuration Management process – making it the most up-to-date and trustable data. The fact that it uses data from a CMDB, which is a cross-organizational unit tool, opens space for an improved organizational commitment to the EA, as each stakeholder will have responsibilities in the As-Is definition. This will increase the visibility of the EA tool within the organization. And since it will begin to

be associated with the CMDB, the organization's EA will extend its scope, and stop being restrained to higher-level business stakeholders.

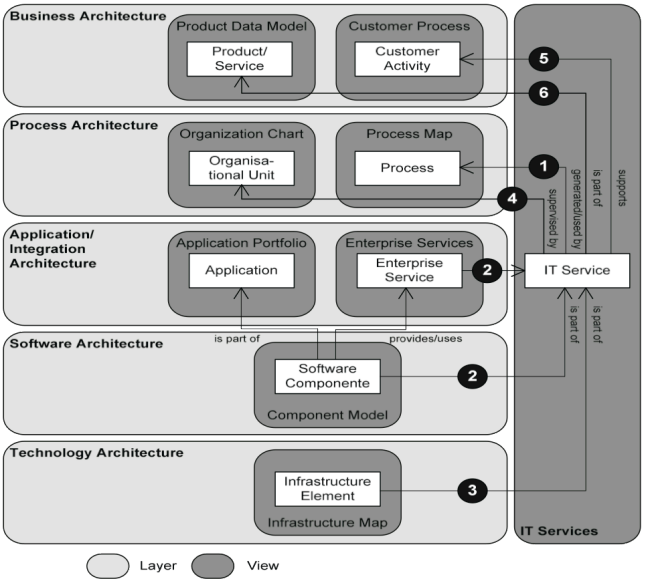

Figure **3** - Integration between IT Services and EA [Braun e al. 2007]

The role of the Enterprise Architect will also be set to change. In organizations where there is only one architect responsible for all the EA, using our approach will considerably simplify his life, and the resulting EA design, since now, with a large amount of data already available at his disposal, the As-Is model will be much easier to realize.

Functionalities of a CMDB tool in our solution

The CMDB will only be responsible for the features related to data and its management. Thus, the original list of functionalities is reduced to:

- Reconciliation
- Federation
- Access Control
- Versioning
- Synchronization
- Branching and merging capabilities

Functionalities of EA tool in our solution

We consider that the data repository component should be removed from EA tools, and be an activity that falls under the responsibility of a CMDB. Additionaly it gains responsability over the CMDB's visualization features, and keeps the remainder of its previous functionalities such as:

- Modeling capabilities
- Simulation and analysis
- Life-cycle management
- Publishing
- Templates
- Product Architecture

The last one now relates only to the robustness and scalability for the models that this tool stores.

5. Evaluation

Besides being more or less in agreement when speaking in the present functionalities, there is a point where CMDB vendors haven't got a so obvious consensus: the meta model definition. Some have opted by a static meta-model where the categories of the CIs are already defined in the first start of the system; and others chose a dynamic meta-model approach where the user can define the meta-model that most matches their needs. For the purpose of integration between a CMDB and an EA tool, the static meta-model is much easier to deal but is also very limitative for a solution that has to fit exactly to the organization's reality. In this domain, what is asked of these tools is exactly the power to react to change, predict and estimate the costs of changes whether to enable performance or correct some non-well evaluated decision. So, to this purpose, it is essential to maintain the dynamic ontology and to allow that organization's changes don't stay attached to today's infrastructure, systems configuration or even existent business processes which probably, after the EA definition will be different than the actual ones. So in order to implement this solution in a CMDB with a dynamic ontology is necessary to catch a change in the ontology and reflect it in the EA tool. The complexity to implement this automatically cannot worth it because once the ontology is defined is very unlikely that it'll change. The only exception are the changes that the Architect probably will do in the To-Be model, but, in this case, doesn't threats this approach, as the changes go from the EA to the CMDB.

Actually, even being mostly a technological solution, a CMDB can show models with a bottom up view of the system in their visualization module. In the other hand EA modelers aren't as focused on data as a CMDB and are more aligned with the production of top down views of the system (depending on the tool and point-of-view of its vendor). This solution tries to merge the best of two worlds integrating a top-down view of the organization that reaches the lower ground: the IT infrastructure, machines, routers and all the IT assets that support business. Associating the EA with the CMDB will also enable a broader share of organizations strategies and objectives, to all stakeholders making them part of whole, and more committed to these strategies.

6. Conclusion

The businesses need IT that can evolve as quickly as the real world. In that matter it is necessary to converge practices, tools and standards to help the interoperability of the entire IT infrastructure as whole. Data is very important for organizations. In that matter, data centralization is a good practice that this solution helps to achieve for this kind of support systems for ITSM. It was shown, CMDB and EA tools are really close one another, and so, it

makes all the sense to unify its competencies and integrate them with each other, optimizing the processes and services that they support.

This domain is still in its infancy, but once vendors of CMDB and EA tools start to think this way, organizations that need this kind of tools will save time and money in their installation process.

The next step will be an implementation of this solution. This implementation might face a problem when it'll try to be compliant with the federated CMDB API, because this interface was not yet presented to public. However, we think it's possible to implement a solution with an interface very close to the final federated CMDB API, and later make the necessary adjustment to match to the federated one.

7. References

Anaya,Victor; Ortiz, Angel; "How Enterprise Architectures Can Support Integration" , ACM, 2005

BMC Software, "What do you need from a Configuration Management Database (CMDB)?", BMC Software, 2005

Braun, Christian; Winter, Robert; "Integration of IT Service Management into Enterprise Architecture", ACM, 2007

Clark, Dave; Dublish, Pratul; Johnson, Mark; Kowalski, Vincent; Labrou, Yannis; Negritoiu, Stefan; Vambenepe, William; Waschke, Marv; Wiles, Van; Wurster, Klaus; "The Federated CMDB Vision", BMC, CA, Fujitsu, HP, IBM, and Microsoft, 25 Jan 2007

Colville, Ronni J.; "CMDB or Configuration Database: Know the difference", Gartner, 13 March 2006

Colville, Ronni J.; James, Greta A; "EA Tools and CMDBs: Similar but Different for Some Time to Come", Gratner, 23 August 2007

FEAD, "Enterprise Architecture Tool Selection Guide v.4.2", FEAD, July 2007

Hinich, Nancy; "Service Management Process Maps", Van Haren Publishing, 2007

James, Greta A.; "Emerging Enterprise Architecture Tool Capabilities", Gartner Research, 29 August 2007

Johnson, M. W.; Hately, A.; Miller, B. A.; Orr, R.; "Evolving standards for IT service management", ACM, 2007

Lankhorst, Marc, "Enterprise Architecture at Work: Modelling, Communication and Analysis", Springer, December 2005

Office of Government and Commerce, "Best Practice for Service Support", The Stationery Office thirteenth impression 2006

Peyret, Henry; "The Forrester Wave: Enterprise Architecture Tool, Q2 2007", Forrester, April 25, 2007

Soares, Luís; "Enterprise Architecture: Information Systems Architecture Visualization", IST, 2007

Spewak, Steven H.; "Enterprise Architecture Planning: Developing a Blueprint for Data, Applications, and Technology", Wiley, September 1993