MCIS 2009 Proceedings

Mediterranean Conference on Information Systems (MCIS)

2009

The Information City: A Framework For Information Systems Governance

Sana Guetat
Lyon 3 University, sana.guetat-mabrouk@univ-lyon3.fr

Salem Ben Dhaou Dakhli
Paris-Dauphine University, sdakhli@computer.org

Recommended Citation

Guetat, Sana and Dakhli, Salem Ben Dhaou, "The Information City: A Framework For Information Systems Governance" (2009). *MCIS 2009 Proceedings*. 120.
http://aisel.aisnet.org/mcis2009/120

THE INFORMATION CITY: A FRAMEWORK FOR INFORMATION SYSTEMS GOVERNANCE

Guetat, Sana, Lyon 3 University, IAE de Lyon, 6 cours Albert-Thomas - BP 8242, 69355 Lyon cedex 08, France, sana.guetat-mabrouk@univ-lyon3.fr
Dakhli, Salem Ben Dhaou, Paris-Dauphine University, Place du Maréchal de Lattre de Tassigny, 75775, Paris, France, sdakhli@computer.org

Abstract

Modern organizations face many external constraints arising from an external environment characterized by various economic, political, and technology-related forces that reshape business dynamics. Many authors have emphasized that information technology plays a critical role in maintaining competitive advantage of organizations. Agile Information systems are among the instruments proposed by academics and practitioners to help organizations in managing continuous change and overcoming problems induced by external pressures. Nevertheless, building an agile information system - able to support the continuously changing organizational processes – is a difficult task notably because, in each organization, information system is a stack of applications developed using methods, languages, and tools which corresponds to different periods and technology eras. The heterogeneity of information systems makes difficult both the maintenance and evolution of existing applications or the development of new applications that must be integrated in the information system. Many solutions have been proposed by academics and practitioners in order to help organizations build agile information systems. Despite their richness, the proposed solutions don't consider the relationships between information systems agility and information systems governance. In this paper, we propose a framework - based on the city landscape metaphor - which links enterprise architecture and information systems governance.

Keywords: *Agile information system, information city, Enterprise Architecture, Target Information City Plan (TICP), TICP area, Information System Governance*

1 INTRODUCTION

Modern organizations face many external constraints arising from an external environment characterized by various economic, political, and technology-related forces that reshape business dynamics. The accelerated product lifecycles, reduced time-to-market, increased rate of change due to the globalization and the deregulation of economy, and the increasing volume of information to be processed are examples of such forces. The today's business environment of modern organization is continuously changing, competitive, demand driven and highly dynamic. Many authors have emphasized that information technology play a critical role in maintaining competitive advantage of organizations (Toffolon 1996) (Dewett et al. 2001) (Lucas et al. 1994) (Gurbaxani et al. 1991). These authors have analyzed how information technology (IT) impacts organizational characteristics and outcomes. In particular, (Dewett et al. 2001) stress that information efficiencies and information synergies are among the main performance enhancing benefits of IT and identify five main organizational outcomes of the application of IT that embody these benefits. Moreover, these authors discuss the role that IT plays in moderating the relationship between organizational components including structure, size, learning, culture, and inter-organizational relationships and the most strategic outcomes, organizational efficiency, and innovation. (Gurbaxani et al. 1991) use the economic agency theory (Alchian et al. 1972) (Jensen et al. 1976) and the transaction costs theory (Williamson 1989) to discuss the impacts of IT on the organizations size and shape. Finally, (Toffolon 1996) uses the Leavitt's organization model (Leavitt 1963) (Stohr et al. 1992) to demonstrate that IT help organizations in reducing the impacts of the external environment pressures. In other words, organizations survival depend on the efficiency of the support brought by IT to organizational processes. Therefore, organizations need sophisticated methods, approaches, and tools to solve the new challenges created by environmental constraints. Agile Information systems are among the instruments proposed by academics and practitioners to help organizations in managing continuous change and overcoming problems induced by external pressures. Nevertheless, building an agile information system - able to support the continuously changing organizational processes – is a difficult task notably because, in each organization, information system is a stack of applications developed using methods, languages, and tools which correspond to different periods and technology eras. For instance, an information system may be composed of applications running in mainframe environment, decision support systems based on data warehouses and datamarts, and web applications running in open environments. The heterogeneity of information systems makes difficult either the maintenance and evolution of existing applications or the development of new applications that must be integrated.

Many solutions have been proposed by academics and practitioners in order to help organizations in building agile information systems (Toffolon et al. 2002) (De Souza 2006) (Hovorka et al. 2006). Despite their richness, the proposed frameworks don't consider the critical role played by information systems governance in developing agile information systems. Information systems governance is an integral part of organization governance. It consists of the leadership and organizational structures and processes that ensure that the organization's information system sustains and extends the organization's strategies and objectives. Moreover, information system governance encompasses the full range of information system and IT management activities. In this paper, we propose a framework - based on the city landscape metaphor - which links enterprise architecture (Zachman 1987) (Zachman et al. 1992) (Kaisler et al. 2005) and information systems governance. In particular, our framework describes how agile information systems support organizations strategies through the development of new applications or the evolution of existing ones. Let us note that despite enterprise architecture is still a young academic research area, it is nowadays a well established discipline in IT industry and information systems field. This is due to the role played by enterprise architecture in helping organizations to manage change and overcome the problems resulting from building heterogeneous and non-integrated information systems that fail to support organizational processes. Our paper is organized as follows. Section 2 presents a layered model of enterprise architecture which links information systems architecture to organization's strategy. In section 3, we use the city landscape metaphor to model organization's information system as an information city. Section 4 demonstrates how the information city model contributes to information system governance. In section 5, we illustrate the use of the proposed framework as an instrument of information systems governance. Section 6 concludes this paper by listing lessons learned from the framework validation and future research directions.

2 A LAYERED MODEL OF ENTERPRISE ARCHITECTURE

This section provides a definition of the enterprise architecture concept prior to the presentation of the layered model of enterprise architecture.

2.1 The Enterprise Architecture Concept

Many definitions of enterprise architecture have been proposed in the literature. For instance, (Kaisler et al. 2005) lists two definitions of enterprise architecture while (Beznosov 1998) provides five definitions of this concept. In this paper, we define enterprise architecture as a holistic set of models of all the facets of an organization, intended to help it in change management in a flexible way. Therefore, enterprise architecture provides a set of mechanisms that helps organizations envisage the desired composition of the business and the necessary resources (People, Process, and Technology) to support the operations and objectives of the business. Enterprise architecture seeks to understand the linkages between the organizational processes that are executed and the software systems that support them. The discipline of enterprise architecture can be seen to include the analysis of he current (as is) organization’s information system and the development of desired future the organization’s information system (target or to be), as well as the transformation plans to bridge the gap. The purpose of enterprise architecture in the context of the IT environment is to develop and manage the organization’s information system in a way that supports the business strategy and aligns to business needs. Enterprise architecture is influenced by the business strategy and pressures driving an organization. The subsequent uses of such pressures in gap analysis lead to transformation plans and programs of work. Enterprise architecture is related partly to technology integration. Besides, organizations increasingly need to build an enterprise-wide capability to leverage technology that is distributed in different business units. Many organizations try to establish enterprise architecture standards to enable greater compatibility of information technology (IT) components and integration of applications and data across their business units.

2.2 The Layered Model of Enterprise Architecture

Many organizational actors - with widely varying backgrounds, interests, goals, points of view, and responsibilities - are involved in enterprise architecture definition and use. In order to avoid understanding problems related to information overload, enterprise architecture must present abstractions of information it contains. Such abstractions are referred to architectural views. According to (IEEE-1471 2002) and (Maier et al. 2000), enterprise architecture views are developed to address the concerns of stakeholders. The key is to identify and develop all relevant enterprise architecture views to sufficiently address all business needs. These views will enable the architect to demonstrate how the stakeholder’s concerns are being addressed in the enterprise architecture. The set of views chosen to represent a system is variable, a good set of view must be complete with respect either to the organizational actors concerned with this system, or to the multiplicity of information sources and their characteristics. That means that a good set of views of enterprise architecture must cover all concerns stakeholders and capture different information pieces related enterprise architecture. The multi-layered model of enterprise architecture proposed by (Dakhli 2008) is an example of a multi-viewed model which represents each view as a layer and describes the relationships between enterprise architecture views. This model relies on five interacting layers: the strategy layer, the process architecture layer, the functional architecture (information system architecture) layer, the applicative architecture layer, and the software architecture layer (Figure 1). The strategy layer defines the organizational problems to be solved and their organizational solutions. Such problems results from the organization’s external and internal constraints. External constraints may be economic, political, social, legal, or related to the evolution of the technology. Internal constraints reflect the impacts of external constraints on the organization’s components: structure, people, production technology, tasks and information technology (Leavitt 1963) (Stohr et al. 1992) (Toffolon 1996).

The process architecture layer describes the organizational processes architecture at the conceptual and the organizational levels. At the conceptual level, an organizational process are modeled as a nexus of activities processing and exchanging information. The processes organizational architecture is the projection of the processes conceptual architecture on the organization’s context, constraints, and priorities. Therefore, the organizational level models organizational processes as nexuses of operational tasks carried out by organizational actors in order to contribute to value creation. The organizational processes architecture is updated according to the organizational solutions defined by the strategic layer.

The functional architecture layer describes the information system architecture as a nexus of informational entities and functions. An informational entity is a set of information chunks which define a concept used by the organizational actors while carrying out an organizational process. A function is an action which uses and transforms at least one informational entity. An organizational process manipulates informational entities through the use of functions. A function may be considered as an aggregation on many sub-functions. Functions may be used by many organizational processes. Such functions are called reusable functions. Informational entities manipulated by many organizational processes are called shared information. Because of the invariant and stable nature of informational entities and functions, they are independent of the organizational structure and the roles played by actors within an organization. Architecture of an organization's information system is defined as a model describing the organization's functions and informational entities as well as the relationships between these concepts. The organizational processes architecture is updated by integrating the impacts of the organizational solutions defined by the strategic layer on the informational entities and functions.

The applicative architecture layer provides a map which describes the organization's applications and information flows they exchange. An application is a set of software systems which computerizes at least partly an organizational process. So, an application provides a software support to the value creation behavior of organizational actors. Such a behavior consists in carrying out organizational processes activities which manipulate informational entities through functions use. An application provides two categories of services: service-to-user and service-to-application. A service-to-user results from an interaction between an end-user and an application in order to help an organizational actor who carries out a set of operational activities. A service-to-application is an intermediate service provided by an application to another applications while processing information. An application may be considered as a dynamic conjunction of a set of organizational process activities with informational entities and functions in order to contribute to products and services production. The applicative architecture layer results from the interaction between the functional layer and the business process layer which supports the problem and operation spaces. The applicative architecture layer delivers a first level description of a software solution as a new or enhanced application which interacts with existing and future applications.

The software architecture layer describes each software solution as a set of software components and connectors distributed according to a software architecture model (e.g. MVC,…). A software solution is either the architecture of a new application which supports at least partly a new organizational process or the architecture of an existing application which is enhanced in order to take into account the modifications of an existing organizational process. Despite the richness of the existing definitions of the software component concept, we propose in this paper a definition of this concept which refers to functions. Our definition states that a software component is an autonomous and homogeneous logical unit which implements a function in order to provide a service either to end-users or to other logical units. A software connector is an autonomous and homogeneous logical unit which facilitates interactions between two software components. A software solution is composed of reusable and specific software components and connectors. A reusable software component implements a function used by many organizational processes.

Consequently, the software solution architecture has many facets associated with the four layers presented above (the process architecture layer, the functional architecture layer, the applicative architecture layer, and the software architecture layer). Each facet corresponds to an architecture metamodel which describes the basic concepts characterizing this facet and their relationships. Finally, we not that the layered Model of Enterprise Architecture presented in this section is compliant with the software global model framework proposed by (Toffolon et al. 2002). For instance, the process architecture layer corresponds to the problem space concept of the software global model while the solution space concept of this framework may be associated with the functional architecture, the applicative architecture, and the software architecture layers.

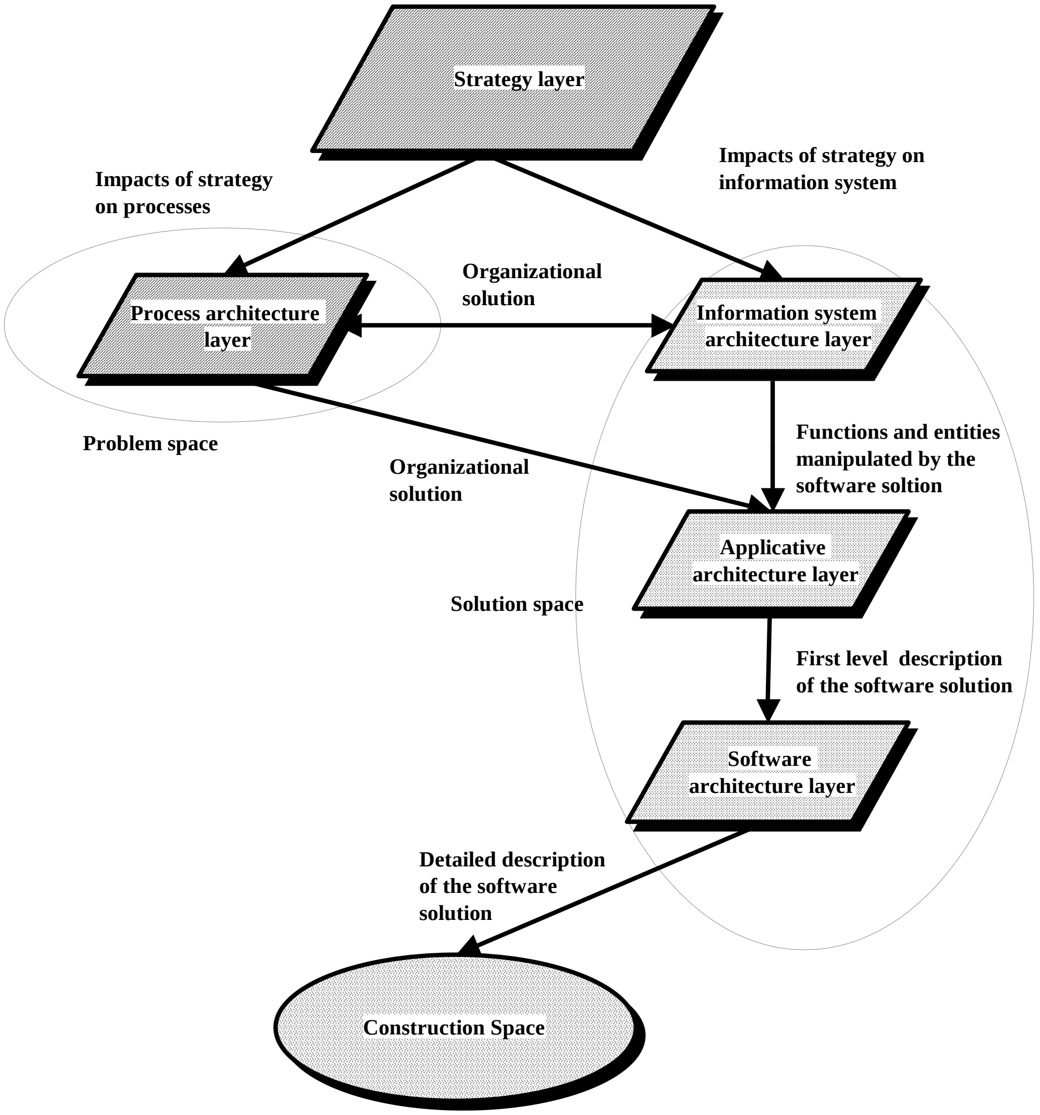

Strategy layer
Impacts of strategy on processes
Impacts of strategy on information system
Organizational solution
Process architecture layer
Information system architecture layer
Problem space
Organizational solution
Functions and entities manipulated by the software soltion
Applicative architecture layer
Solution space
First level description of the software solution
Software architecture layer
Detailed description of the software solution
Construction Space

Figure 1: The layered model of Enterprise Architecture [Adapted from (Dakhli 2008)]

3 THE INFORMATION CITY FRAMEWORK

The information city framework is based on the "city planning" or "city landscape" metaphor. This metaphor is used by architects to communicate more effectively the nature and value of architecture by relating unseen enterprise architecture to real-world concepts that are well understood (Sewell et al. 2001). (Noe 2000) stresses the important role of this metaphor in scientific discovery notably when seeking a new framework. According to this author, when a scientist faces an irregular case, the most effective way is to leverage a traditional concept or terminology as a metaphor to describe the case. A metaphor can be used in various ways. On the one hand, leveraging a metaphor is helpful not only in communication between architects and their clients but also to find some missing functions or knowledge in the new field. On the other hand, the "city planning" metaphor may be employed to build approaches which deal with the architecture of the whole information system of an organization. Such approaches are based on two analogies. Firstly, they assume that building software systems may be compared to building houses. Secondly, they compare organizations information systems to modern cities composed of common parts (roads, public gardens, roads,…) and private parts belonging to individual persons who live in the cities (houses, private gardens,…). In this section we present the "information city" concept prior to illustrating it with the city plan instrument.

3.1 The information city concept

The information city framework proposed in this paper generalizes the use of the "city planning" metaphor by stating that – within a modern organization – an information system may be considered as a city where the inhabitants are the applications belonging to this information system. In this city, called the information city, the common parts are information shared by all the information system applications while the private parts are composed of software artifacts owned by each application. An application belonging to the information city behaves as master of its proper data and artifacts and as a slave regarding shared information. That means that an application can use, update or suppress data and artifacts it owns but can only use a copy of shared information.

Comparing an information system to a city extends the use of the "city landscape" beyond the analogy between software and building construction by emphasizing the problem of information system governance. On the one hand, following the example of a city, the relationships between the applications which populate the information city must be managed. That means that a set of architecture principles and rules has to be specified in order to govern exchanges either between application belonging to an information system or between such applications and the external environment like other information systems or end-users. On the other hand, the vast number of application assets in combination with the natural expansion of the application portfolio as well as the increasing complexity of the overall information system, drive a need for the information system governance. Therefore, the "information city" framework permit defining architecture principles and rules which help organizations prioritize, manage, and measure their information systems.

3.2 The Information City Plan (ICP)

Using the "information city" framework makes organizations able to apply a structure for classifying information system applications, functions, or services in a coherent way. It defines responsibility plots from coarse to fine-grained into discrete areas, which together form the complete Information City Plan (ICP).

Developing the ICP of an organization's information city is a result of a deep understanding of both the business and IT strategy of this organization. One of the central concepts of the ICP is the desire to eliminate the intricacy of the IT environments through the separation of concerns from the applications. The following is a high level approach undertaken to derive the ICP shown below (Figure 3). The organization's business strategy is based on four principles described in the following table (Figure 2):

Principle	Description
Focus on customers	This principle determines the need to know and manage clients, to organize the company accordingly, to improve service quality, and the need for flexibility in how clients access our product offerings
Capitalize on the existing networks	This principle consists in energizing and improving the efficiency of the commercial activity, helping to keep the clients, by providing a global offer to the commercial network
Innovate	This principle consists in innovating and adapting the client offer with the need to enlarge the Products and Services offerings
Focus on operational excellence	This principle stresses on the continuous improvement of organization's processes and products

Figure 2: Organization's business strategy principles

The organization's IT strategy can be summarized as follows:

a) Be able to face organization and technological changes.

b) Be loosely coupled with the business strategy:

c) Shift from organizational processes supported by one application to multiple applications supporting different parts of the organizational processes.

d) Shift from dedicated data to shared data for businesses.

e) Be loosely coupled to the IT and project organization.

f) Be simpler with modular applications, with a maximum of internal cohesiveness.

Analysis of the principles behind the organization’s and IT strategies leads to the four following architecture principles which help guide the development of the organization’s information city TICP.

- Determine Front-office vs. Back-office responsibilities
- Specialize back-office regarding the organization’s processes
- Identify the components common to the back-office and the front-office**.**
- Separate in the front-office the functions related to management of the communication network from those related to management of the relationships with the organization’s customers and partners.

The first architecture principle - Determine front-office vs. back-office responsibilities – identifies the responsibilities of the organization’s front-office and back-office. The front-office is dedicated to management of the relationships with the organization’s external environment while the back-office is dedicated to the development of products and services. For instance, within an insurance company the back-office manages the insurance and services commitments whatever the distribution channels.

The second architecture principle - Specialize back-office regarding the organization’s processes – permits identifying a “Business Intelligence” area, a “Support area”, and at least one business area. A “Policy and Claims area” is an example of a business area within an insurance company.

The third architecture principle - Identify the components common to the front-office and the back office – refers to either the components that link the front-office and the back-office or the artifacts shared by the back-office and the front-office. Application of this principle results in identifying two areas: an “Integration area” and a “Shared information area”. The first area allows exchanges of informational flows and services between the back-office and the front-office applications. The second area contains information shared by all the applications of the organization’s information system as well as the applications which manage shared information data. The customers and products repositories are examples of information shared by all the applications of an organization’s information system.

The fourth architecture principle - Separate in the front-office the functions managing the communication network from those managing the relationships with the organization’s customers and partners – permits identifying two areas: an “Inbound and Outbound flows Management area” and a “Party Relationships area”. The “Inbound and Outbound flows Management area ” is dedicated to the management of the informational flows exchanged by an organization and its external environment. This area describes the various technology channels used by an organization while exchanging information with external environment. The “Party Relationship area” supports the relationships linking an organization with its customers and partners whatever the communication channel.

The following schema (Figure 3) presents an example of ICP which may be used to illustrate the information city in various service-intensive organizations like banks and insurance companies.

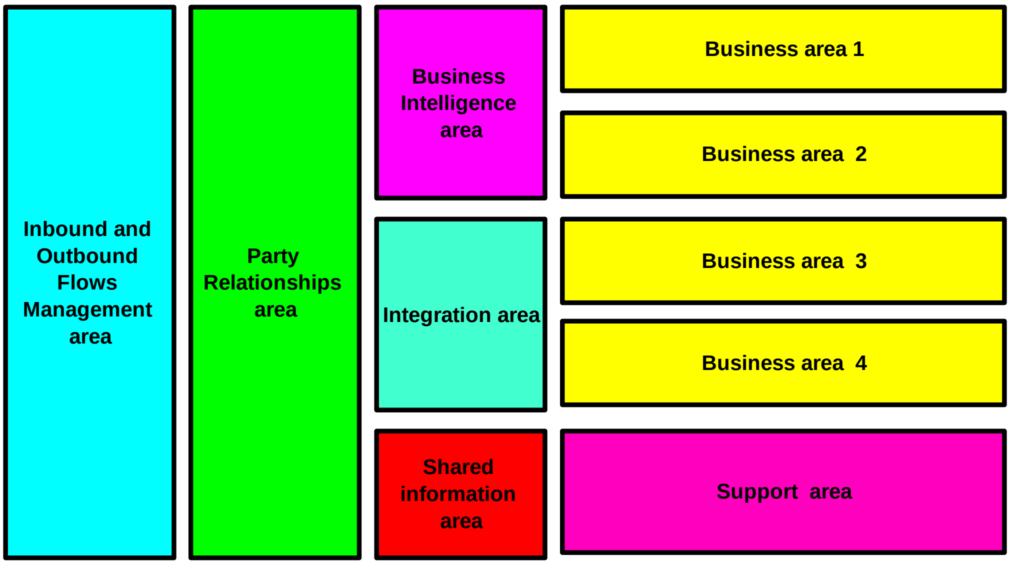
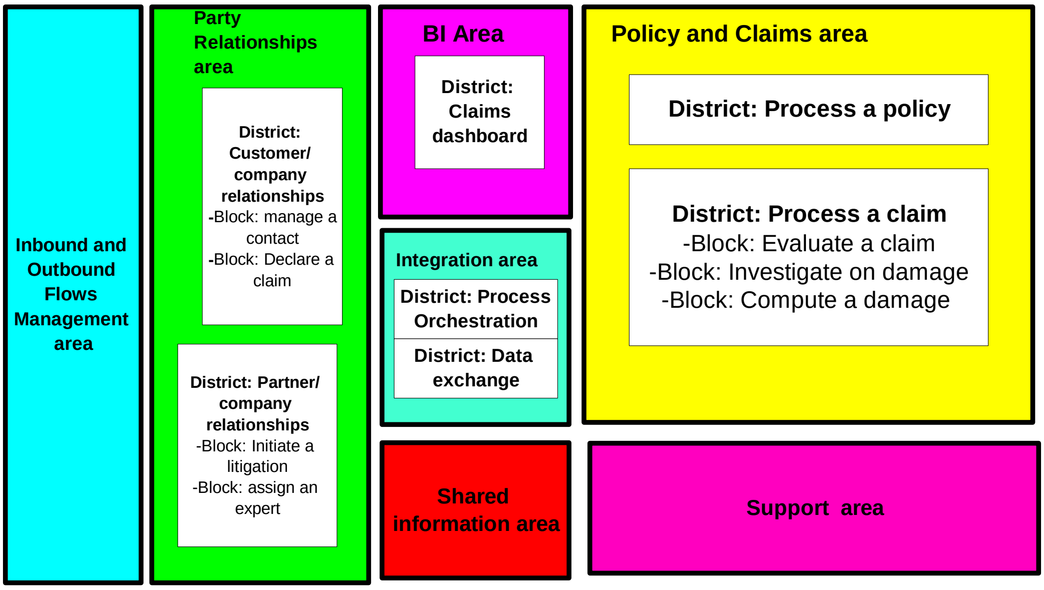


Figure 3: The Information City Plan (ICP)

Let us note that each area of the ICP can be broken down into more discrete areas of functionality. Generally, an area is composed of districts and a district is a set of blocks. The following example (Figure 4) illustrates such decomposition in the case of a Claims Management application within an insurance company. This example is not a complete decomposition of the information city ICP; rather it is an extract which is used to highlight the main required functions for this particular application.



Inbound and Outbound Flows Management area
Party Relationships area
District: Customer/ company relationships
-Block: manage a contact
-Block: Declare a claim
District: Partner/ company relationships
-Block: Initiate a litigation
-Block: assign an expert
BI Area
District: Claims dashboard
Integration area
District: Process Orchestration
District: Data exchange
Shared information area
Policy and Claims area
District: Process a policy
District: Process a claim
-Block: Evaluate a claim
-Block: Investigate on damage
-Block: Compute a damage
Support area

Figure 4: An example of ICP

4 INFORMATION SYSTEMS GOVERNANCE

In this section, we highlight how the information city framework can be used to support information systems governance. As stressed above, information systems governance consists in two related activities. Firstly, information systems governance defines a set of architecture rules and principles which help manage services and informational flows exchanges between applications belonging either to the same information system or to different information systems. Secondly, information systems governance provides a set of architecture rules and principles dedicated to the management of applications evolution. Taking into account these two aspects of governance is required to guarantee a high level of agility of information systems. The ICP plays an important role in the governance of information systems. At a high level, the ICP provides:

- Increased alignment between business, information system, and IT.
- A single medium based on a stable and lasting tool that facilitates clear communication on the evolution plan of the information system to senior management right down to the grass roots.
- A mechanism to guide and govern action plans and software projects.

At a more detailed level, the ICP helps companies:

- Coherently define and manage their application systems
- Apply business orientations in software projects.
- Establish a solid basis for decision-making (Position business needs, assess current applications, and finally draw target and roadmaps).
- Define clear responsibilities between applications belonging to the organization's information system.
- Evaluate the impacts of proposed solutions.
- Design software solutions by providing a detailed mapping of organizational processes and business functions to applications belonging to the organization's information system.

Moreover, the ICP can be used to provide a common and shared vision of the current and target structures of the organization's information system. The current structure of an information is system is obtained through the description of the functions and the informational entities supported by applications and the areas, districts and blocks containing them. The discrepancy between the current and the target organization's information system results from architecture principles and rules violation. An important aspect of information system governance within an organization consists in carrying out a gap analysis and developing a roadmap to move from the current structure to the target structure of this organization's information system. Two instruments are derived from the ICP to describe the current and the target information system architectures: the current Information City Plan (CICP) and the Target Information City Plan (TICP). The CICP is built prior to the gap analysis which identifies the main architecture problems of the current information system and proposes a roadmap to solve such problems and reach the TICP according to the organization's constraints and priorities.

Architecture principles and rules used in information systems governance are related to the organizational context and depend on the organization's constraints, priorities, and technical maturity. However, the information city framework provides a set of architecture rules and principles that may be shared by many organizations belonging to the same business domain. Such architecture principles and rules include the four principles identified above while defining the ICP. A deeper analysis of the ICP results in additional architecture rules and principles. For instance, in order to guarantee weak coupling between applications, the following principles may be applied. Firstly, exchanges of services and informational flows between applications belonging to different ICIP areas must take place through the Integration area. Secondly, each application must be belong to only one district and one block of an ICP area.

Furthermore, specific architecture rules may be defined to govern applications within a specific ICP area. The layered model of enterprise architecture presented above may be used jointly with the information city framework to define a set of architecture principles and rules that permit governing all the facets of enterprise architecture like processes, applicative a and software architectures. Generally, an architecture principle is associated to many architecture rules which provide details related to this principle and explain how to use it. Architecture principles and rules may be stored in a repository called Architecture repository. This repository is composed of at least two parts. The first part describes architecture principles and architecture rules common to all the ICP areas. While the second part describes architecture principles and rules specific to each ICP area.

Finally, the information city framework provides instruments to define software development, maintenance, and evolution approaches which are compliant with enterprise architecture principles. An example of such an approach is presented in (Dakhli 2009).

5 THE USE OF THE INFORMATION CITY FRAMEWORK: AN EXAMPLE

In this section, we illustrate how the information city framework can be used to govern the information system within an insurance company. We consider the Claim Management application recalled above. Moreover, we assume that this application already exists and implements the functions listed in Figure 4. The existing Claim Management application is composed of the following software systems: Claim processing, Policy processing, Claims dashboard, Process orchestration, Data exchange, Company/Customer relationships management, Company/Partner relationships management. We assume that the following architecture principles are applicable:

- **Principle 1:** Exchanges of services and informational flows between applications belonging to different ICIP areas must take place through the Integration area.
- **Principle 2:** Each application must be belong to only one district of an ICP area.
- **Principle 3:** Each application belong to the area which contains the functions and informational entities it implements

The software systems composing the Claim Management application are located in the CICP is according these architecture rules and principles (Figure 4).

Software system	CICP area
Claim processing	Policy and Claims
Policy processing	Policy and Claims
Claims dashboard	BI Area
Process orchestration	Integration area
Data exchange	Integration area
Company/Customer relationships management	Party Relationships area
Company/Partner relationships management	Party Relationships area

Figure 5: Location of the Claims management application in the CICP

The Claim Management application spreads out on several CICP zones and thus violates the architecture principles 1, 2, and 3. Exchanges between the different software systems composing this application take place according to point-to-point connections mode. Therefore, these software systems are strongly coupled and the Claim Management application is a tangled mess difficult to maintain or modify. To solve this problem, the Claim Management application must respect all the architecture rules and principles listed above in order to move from its current state associated with the CICP to its target state associated with the TICP. To respect the architecture principles listed above, the Claim Management application must split into five applications: Policy processing, Claim processing, Claims dashboard, Company/Customer relationships management, and Company/Partner relationships management. Data exchanges between applications belonging to different ICP areas and Process orchestration are carried out by two software packages: Enterprise Applications Integration (EAI) system and Business Process Modeling (BPM) system. (Figure 6) illustrates the target state of the Claim management application associated with the TICP. This state can be reached through the development of a roadmap based on a deep gap analysis.

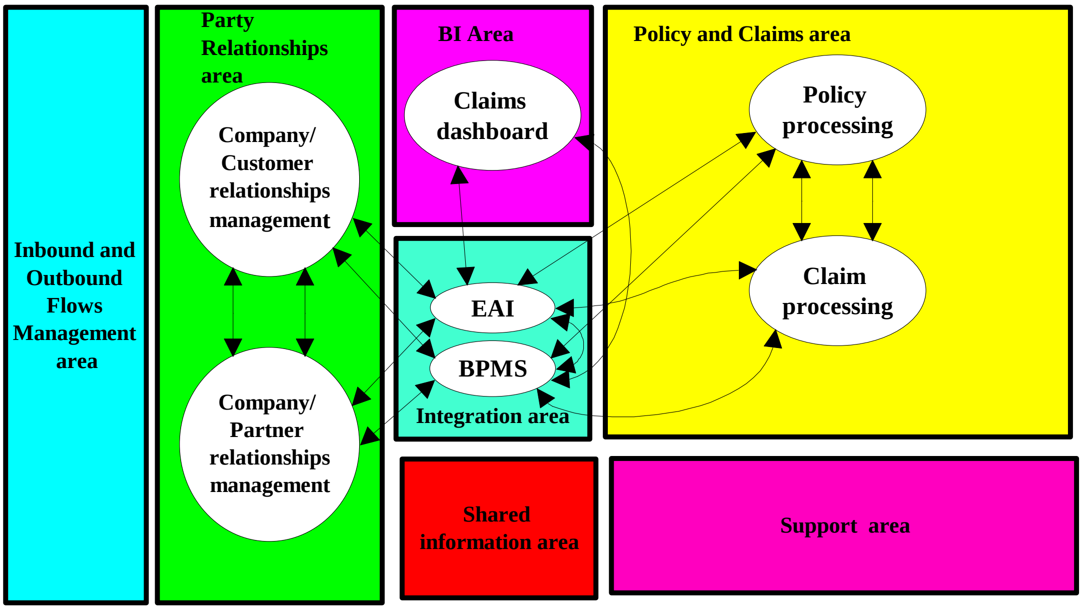


Figure 6: The target state of the Claim Management application

6 CONCLUSION AND FUTURE RESEARCH DIRECTIONS

This paper presents a framework – called information city – which provides a set of instruments that may help organizations in carrying out effective governance of their information systems. Our framework extends existing enterprise architecture frameworks at two levels. On the one hand, it provides information systems architects with a structure and instruments aimed at helping them in deriving architecture rules and principles. On the other hand, our work links enterprise architecture and information systems governance. In particular, it may be used to define software development, maintenance and evolution approaches which are compliant with enterprise architecture principles and rules. Finally, our framework may be used to define effective software artifacts reuse within an organization.

The validation of this framework within an important French insurance company results in identification of many research directions derived either from problems encountered or from architectural and governance related needs expressed by stakeholders. First of all, to evaluate the enterprise architecture of an organization, a maturity model of enterprise architecture has to be established. Such a model may allow either intra-comparison of the architectural maturity of applications which belong to the same information system or inter-comparison of the whole architectural maturity of many information systems. Furthermore, such a maturity model permits defining guidelines and practices to help organizations in enterprise architecture improvements and information systems governance. This problem is associated with a measure problem. Thus, the definition of a set of metrics to evaluate enterprise architecture within an organization is another important research direction. Finally, the use of this framework pointed out the information-oriented nature of enterprise architecture. So, linking enterprise architecture to knowledge may be an interesting research direction.

References

Alchian, A.A., and Demsetz H. (1972). Production, Information Costs and Economic Organization. American Economic Review, 62 (5), 777-795.

Beznosov, K. (1998). Architecture of Information Enterprises: Problems and Perspectives. Written for the "Advanced Topics in Software Engineering", seminar given by Dr. Michael Evangelist, Spring, 1998.

Dakhli, S.B.D. (2008). The Solution Space Organisation: Linking Information Systems Architecture and Reuse. In the Proceedings of the ISD'2008 Conference, Paphos, Cyprus, August 25-27, 2008. Springer-Verlag.

Dakhli, S.B.D. (2009). Software Solutions Construction According to Enterprise Architecture Principles. In the Proceedings of the AIM'2009 Conference, Marrakech, Morocco, June 10-12, 2008.

De Souza, K.C. (2006). Agile Information Systems: Conceptualization, Construction, and Management. Elsevier, 2006.

Dewett, T. and Jones, G.R. (2001). The Role of Information Technology in The Organization: A Review, Model, And Assessment. Journal of Management, 27 (3), 313-346.

Gurbaxani, V., and Whang, S. (1991). The Impact of Information Systems on Organizations and Markets. Communications of the ACM, 34 (1), 59-73.

Hovorka, D.S., and Larsen, K.R. (2006). Enabling agile adoption practices through network organizations. European Journal of Information Systems, 15, 159–168.

IEEE (2002). IEEE Recommended Practice for Architectural Description of Software Intensive Systems (IEEE Standard 1471).

Jensen, M.C., and Meckling, W.H. (1976). Theory of the Firm: Managerial Behavior, Agency Costs and Ownership Structure. Journal of Financial Economics, 3 (4), 305-360.

Kaisler, S.H., Armour, F., and Valivullah, M. (2005). Enterprise Architecting: Critical Problems. in the Proceedings of the 38th HICSS, Hawaï, IEEE Computer Society Press.

Leavitt, H.J., (ed.). (1963). The Social Science of Organizations, Four Perspectives. Prentice-Hall, Englewood Cliffs, New Jersey.

Lucas, H.C., and Baroudi, J.J. (1994). The Role of Information Technology in Organization Design. Journal of Management Information Systems, 10 (4), 9-24.

Maier, M.W., and Rechtin, E. (2000). The Art of Systems Architecting. Boca Raton, CRC Press.

Noe, K. (2000). Scientific philosophy for discovery. In Morishita, S. and Miyano, S. (Eds.), Discovery science and data mining, Kyoritsu Shuppan.

Sewell, M. T., and Sewell L.M. (2001). "The software architect's profession: An introduction, Prentice Hall, PTR.

Stohr, E.A., and Konsynski ,B.R.. (1992). Information Systems and Decision Processes, IEEE Computer Society Press.

Toffolon, C. (1996). L'Incidence du Prototypage dans une Démarche d'Informatisation. Thèse de doctorat, Université de Paris-IX Dauphine, Paris.

Toffolon, C., and Dakhli, S. (2002). The Software Engineering Global Model. in Proceedings of the COMPSAC'2002 Conference, August 26-28, 2002, Oxford, United Kingdom, IEEE Computer Society Press.

Williamson, O.E. (1989). Transaction Cost Economics. in "R. Schmalensee and R. Willig (Eds): Handbook of industrial organization", North-Holland.

Zachman, J.A. (1987). A Framework for Information Systems Architecture. IBM Systems Journal, 26, 276-292.

Zachman, J.A., and Sowa, J. (1992). Extending and Framework for IS Architecture. IBM Systems Journal, 31, 590-616.