Preprints of the 19th World Congress
The International Federation of Automatic Control
Cape Town, South Africa. August 24-29, 2014

Complex systems renewal: positioning, concepts and architectural issues

Marc Zolghadri, Florent Couffin, Patrice Leclaire *
Simon Collart-Dutilleul **

* *Lismma, Supmeca-Paris, 3, Rue Fernand Hainaut, 93407 Saint-Ouen, France (e-mail: Firstname.Name@supmeca.fr).*
** *Cosys-Estas, Ifsttar, Lille-Villeneuve d'Ascq, France (e-mail: Simon.Collart-Dutilleul@ifsttar.fr).*

Abstract: We look at understanding the main mechanisms behind the renewal of complex systems. These systems, such as trains or airplanes, have a long operation or service duration. They should be renewed and their renewal projects are often partial for economic reasons. A renewal act is performed through integration of engineering changes. Any functional or structural change could have cascading effects on consecutive subsystems. The system architecture can be used as the main driver of renewal acts definition and planning. We position the renewal problem and define some key concepts usable for complex systems renewal. They allow to define the renewal possibilities as scenarios that should be qualified for ultimate selection. This paper explores these mechanisms and suggests a first set of propagation mechanisms. Throughout the paper, we illustrate our purposes by using examples from a train renewal project.

Keywords: Complex systems, renewal processes, upgrading, architectures, design systems, system identification, multilevel systems, impact, invariants.

1. INTRODUCTION

1.1 Upgrading systems

In this paper we discuss the challenges and issues associated with upgrading or modernization of *complex systems* integrated in their hosting structuring systems. We define *structuring systems* as a sub-class of infrastructures that deliver services (such as water, transportation, energy) to a very large number of consumers over a long time, up to several generations, see (16). The structuring systems are therefore the critical infrastructures whose "incapacity or destruction would have a debilitating impact on the (*our* in the original text) defence or economic security", reported in (11). When studying structuring systems, we do consider not only the technical systems but also that organization that makes it run properly. Structuring systems (such as transportation or water supply) are implemented all over the countries and irrigate almost every corner. They have extremely large dimensions, are complex and deliver simple services. Critical infrastructures or structuring systems form a fundamental part of a country heritage, see (9), (3), (10). A structuring system uses *complex systems* to deliver services. Relying on the (5) standard, we distinguish the *primary* complex systems (they contribute directly to the success of the service delivery to customers or users) from the *enabling* systems. In the railway transportation, trains, railways and monitoring systems are the primary complex systems while maintenance workshops are enabling ones. This paper's focus is put on complex systems within their hosting systems. We use the rail transportation as an illustrative example throughout the paper. The idea is to upgrade a train coach. Trains have a very long operation duration, about 40 to 45 years. They are renewed or upgraded several times during their lifecycle, often partially, to minimize the immobilization period because of cost efficiency. Renewal and upgrading refer to partial re-design of systems or products. Even if, at a first glance re-design is nothing else than designing again, the reality is much more complex. Designing a product or system is to transform needs into a (physical) solution. This is done by using all existing knowledge, know-how and experience of designers. But, re-design means that designers have "knees deep in the mud" because the system does exist and designers have to minimize its immobilization while replacing worn parts or modernizing some others. The re-design is highly constrained; some hard constraints without any room to modify and some soft constraints. The final solution has to have better performances and to be more user-friendly. Some constraints are related to decisions made some decades ago at the preliminary design of the system (gauge size for instance) while others are due to the functional constraints imposed by new security directives.

1.2 Research motivations and methodology

In a renewal project, upgrading leads to changes that have to be implemented without degrading the global system *performance*. Implementing a change refers to the fact that it has to be integrated to the system's structure and or functionalities. Changes once validated might be propagated through the whole functional and physical structure of system, see, (15) and (14). These

⋆ Authors would like to thank Prof. C.Eckert from the Open University for her precious remarks.

Copyright © 2014 IFAC

cascading effects are desirable or not but in all cases they will lead to cost that have to be minimized. Often in such projects, only some of the outputs of the renewal activities are known while others could not be identified from the beginning of the change implementation project. Therefore, it is quite important, for any renewal project, to assess as precisely as possible the effects of such cascading effects, called change propagation.

The survey of the state-of-the-art we performed showed that most design tools and methods have been developed for new product/system development; however many design projects involve upgrading or renovating existing systems by proceeding to their redesign or re-engineering. New products are often developed as a modification of existing products or integration of partial solutions from one or more existing products. This survey together with the preliminary observations discussed just before show somehow that there could be niches of research challenges related to the upgrade of such systems. The goal is therefore to identify the *scope* and *depth* of the consequences of the integration of a change. Scope and depth are two parameters that allow to characterize the consequence of the *integration* of a change within the system. These two concepts will be defined later on in section 4. In short, they allow to specify the impacted sub-systems by the studied change. By change integration we mean the successful execution of renewal process. The renewal process modifies the system functions or structure or both to answer a set of renewal needs and expectations. The paper defines the methodology of system upgrading and does not contain any upgrading techniques; it covers the main issues identified as critical to perform such projects in an efficient manner.

1.3 Contributions and the paper structure

The main contribution of the paper is to provide a framework to upgrade a system. The concept that allows to guarantee the upgrading efficiency is the architecture of the system as the main skeleton of changes integration process. The architecture is "... what is essential about that system considered in relation to its environment", according to ISO/IEC/IEEE 42010. We will go through the concept of architecture which clarifies the dependencies between physical modules or functions. The scope and depth of change integration or change propagation can be modeled by studying the dependencies between the physical modules and the functions they perform. Another contribution concerns the dominance. This concept looks to deal with the phenomenon that intuitively puts a differentiation between "small" and "big" changes. For instance, we do understand that a validated change about a whole locomotive could be more critical than a validated change about shape of the coach seats. Some changes could deeply impact the system while others are only cosmetic. Some changes are interdependent while some changes dominate the others. For instance, changing the shape of a train cockpit clock to a circular one is dominated by a validated change regarding the whole cockpit. The changes associated with these dependent parts could have more or less deep consequences once propagated. The dominance is modeled in a first attempt by the scope and depth of the changes.

The paper discusses the needs of complex systems upgrading. Next section provides briefly the main concepts related to complex systems in our study. In section three, we will expose some exploratory results concerning upgrading and provide a taxonomy that allows to classify the upgrading activities in terms of structure and function. In section four, we will expose the idea that the usage of architecture could give a broad framework for understanding upgrading activities explored in section three. We discuss then the results reported in this paper by showing how their application could help to understand the upgrading projects.

2. COMPLEX SYSTEMS

2.1 Complex system in the literature

The complex systems we focus on are complex because (i) they are made of a very great number of physical and functional modules, (ii) they require high level of effort to be analyzed, understood and modelled, (iii) they are knowledge intensive and multidisciplinary, (iv) they involve lots of people to be run properly, (v) the knowledge and data, related to these systems are highly distributed and rarely detained by only one person, (vi) and they are highly dynamic. In (12) the complexity is defined as follows "Roughly, by a complex system I mean one made up of a large number of parts that interact in a nonsimple way". For further details about complex systems, readers could refer, for example, to these fundamental readings (8) and (12). These complex systems considered here is defined as follows: "A complex system once initially designed, is continuously renewed and upgraded in its functional and physical structure, [*renewable*]. There are hard constraints to minimize the service delivery interruption and only some of its parts are replaced or renewed at a time [*perpetual*]. It becomes an asset transmitted from one generation to another thanks to their long service delivery duration [*heritage*], and it is maintained and controlled by a complex organization of decisions [*management*] made by humans. A complex system is firmly intertwined with a main structuring system from one side and with other complex systems [*couplings*], and it has a complex structure formed by a network of networks [*web*]. "

2.2 Upgrade of complex systems

There are several terms used to refer to renewal of a system: renewal, refurbishment, modernization, rebirth, etc. (source: CNRS dictionary http://goo.gl/vab3M). According to Oxford dictionary renewal means (i) the action of extending the period of validity of a licence, subscription, or contract, or (ii) the replacement or repair of something. In order to go deeper in understanding the renewal basics, first we remember how a system is defined. According to Ulrich (13) "a system can be defined thanks to its structure, its functions and the mapping between these functions and the structure". We rely on this definition to build the renewal framework.

We distinguish four levels of renewal: Renovation, Modernization, Extension and Conversion. *Renovation* is to change worn parts. This is the traditional repair or maintenance. The parts have to be replaced basically due to two sets of dysfunctions:

(1) Functionality. The parts do not fill their functionalities or have a deteriorated performance.
(2) Structural. They have their internal or external structural decayed due to their utilisation. In both cases the dysfunction could cross the parts frontier affecting other dependent parts.

Renovation means therefore replacing worn parts by exactly the same parts to fulfill again the functionalities, functions' performances or structural roles within the product. *Modernization*, from a structural point of view, means that a part, worn or not, is replaced by another part using new technologies. The replaced part should fill all the previous functions and the exchanges with other parts through its interfaces have to be ensured. This happens when a new technology seems to be more reliable, desirable, or robust than the old one. Changing the analogical screen of a desktop computer with a LED screen is a structural modernization. Modernization can also target at one or several functions. This can be seen as adding new specifications for the final product. In this case, all the parts contributing to the considered functions could be impacted by the change. Moreover, if the function to be modernized has any contribution to other higher level functions, they will also be impacted. *Extension* refers to those situations where new functions are added to the existing ones. In fact, rarely, designers decide to add new parts or modules to an existing system or product if no new or complementary function is not to be fulfilled. Therefore, the main origin of extension scenarios are functional. However, again the impacted functions and parts have to be analysed in terms of possible necessary modifications. This is for instance to upgrade a car dashboard by adding a GPS to the dashboard of to replace the radio by an integrated computer. Finally, *Conversion* refers to those situations where the main functions of the system is modified in order to answer to new set of customers's expectations. Transforming a passenger train into a freight one is the conversion. Converting means then answering new expectations by adding new functions or improving existing ones which in its turn will create deep structural changes by eliminating, adding or improving some parts.

These four goals for a renewal project can be structured thanks to two main dimensions which are function and structure. At the same time the likelihood and impact level are represented. In fact, it is reasonable to see that the low-level improvements are more frequent that the high-level ones. This the main reason of representing these two characteristics in the opposite direction for the structural dimension (vertical axis) and the functional dimension. Therefore, the four aforementioned types of upgrade are positioned according to the structural and functional dimension of improvements. At the left bottom side of the figure, where the functional and structural improvements are at low-level impacts and high-level likelihood the renewal projects are renovations. If the renewal activities transform the system functions in a moderate or slight manner. This is functional *adaptation* (or evolution). On the contrary, if the system functions are deeply changed or if new functions are added to existing ones we talk about *rupture* (or radical advancement). In terms of structure transformation, in a same way, if the transformation of the structure is judged to be limited, it is adaptation or evolution. While the deep modification of the structure by renewal activities is called rupture. The suggested taxonomy of the renewal project goals are therefore is mapped on the Fig.1. At the same time, a modernization project could include renovations. In a same manner, conversion can contain the each of the other projects for various parts. These inclusions are represented by arrows on the figure which show they main trends of upgradings.

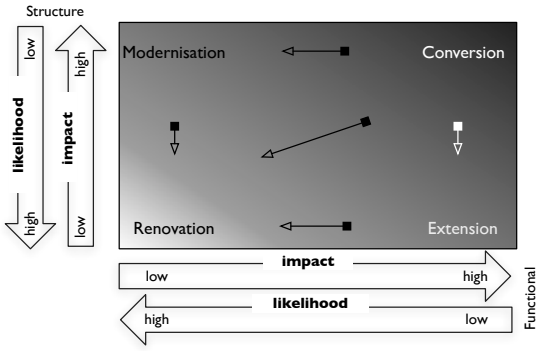

Fig. 1. The taxonomy of the renewal project goals

2.3 A data model of renewal project

We define a renewal project as "a set of *renewal acts* that has to be applied to a system to renew or upgrade its functionality and/or structure." A renewal act is a process made of necessary organizational and technical activities. Renewal acts are either *required* or *derived*. The required acts refer to those necessary acts to reach the upgrade targets. The derived acts have to be performed to support the required upgrading acts (i.e. the disassembly of the doors to renovate the seats). The derived renewal acts are time consuming, generate extra costs and the renewal actors tend naturally to minimize them. Some of the derived acts can be determined from the beginning of the planning of the required ones (for example washing a part before painting). They are named *foreseenable derived acts*. But most of them might be identified during the execution of required or other derived acts; i.e. they are contextual and called *unforeseenable derived acts*. They are both performed to eliminate all the undesired consequences (post-treatments for instance) of the required acts, to ease or even to make possible the execution of the required or other derived acts. The extra amount of efforts consumed during the renewal projects are related to these derived acts. Only some of the required acts add value to the system upgrading while the derived acts add only costs. The art of renewal is to be able to answer to customer and/or other stakeholders while minimizing the non-added value acts.

Required and derived acts are determined in order to answer the *renewal requirements*. A renewal requirement is the expression of the stakeholders renewal concerns about the system. The renewal acts are composed of *renewal operations*. These operations are performed with the same common goal of upgrading a system. It is possible to map the renewal operations with the renewal requirements. The data model of renewal projects is provided in the Fig. 2.

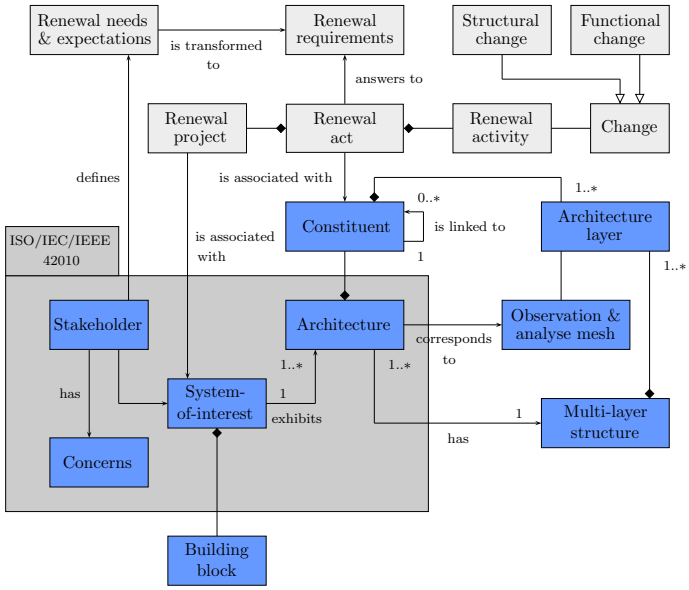

Fig. 2. Renewal data model

A renewal act targets at a given "part" of the system. In order to clarify this concept of part, we make use of the system architecture. As the renewal projects concern existing systems, it is necessary to find out what are the other parts of the system impacted by the renewal acts. This is an identification task. However, even if these parts, in some cases, could be identified more or less easily, not all their attributes (flexibility, security, behaviour, etc.) should be impacted. Some of them are even not observable. Therefore, the impacted attributes have to clarified too. To answer all these concerns about the renewal impacts on the system, the architecture definition provides a relevant analysis and conception tool.

3. HOW AN ARCHITECTURE CAN BE DEFINED?

In this section we discuss the architecture. In a first step, we present a brief survey of main ideas behind the architecture. This will show that even if the amount of work about the architecture is impressive, there is still a conceptual gap that has to be filled to allow the modelling of the architecture of complex systems for renewal purposes.

3.1 A brief survey of previous works

Clements (1) identifies roles of the architecture in project development: basics for communication, project blueprint, blueprint for product line development, embodiment of earliest design decisions, first approach to achieving quality attributes. For an existing complex system, no architecture description does often exist, or even if they do exist they are either partial or distributed in blueprints. The architecture identification is a main driver of the renewal project, if and only if it is described in a relevant way. Mainly, the "architecture of a system constitutes what is *essential* about that system considered in relation to its environment." But, "There is no single characterization of what is essential or fundamental to a system; that characterization could pertain to any or all of (i) system constituents or elements; (ii) how systems elements are arranged or interrelated; (iii) principles of the system's organization or design; and (iv) principles governing the evolution of the system over its life cycle. Architecture descriptions are used to express architectures for systems of interest.", in (4). Stressing the notion of essence or essential, Krob suggests that "the architecture represents the *time-invariance* part of a system. This is to say that part that can be reasonably considered as fix in time, see (7)." Finally, (2) define the system architecture as an abstract description of the entities of a system and the relationships between those entities. It might be understood that the biggest issues in system architecting, specifically for the complex systems is related to: (a) what is essential or what is exactly the time-invariance, (b) what are the architecture constituents, and (c) their links or inter-dependencies.

3.2 Architecture modelling

Respectively, in a functional or structural architecture, the constituents are *activities* or *modules*. The dependencies among the constituents of an architecture are the potential change propagation channels. The data transfert capabilities between the on-board computer and the engine control system defines a dependency which means that any change integrated in one of them could possibly impact the second one. Let us first go through the illustrative example to understand the idea of architecture determination. It concerns the architecture of a train. Suppose that the train is made of two locomotives (loco-1 and loco-2) at each extremity with two similar coaches (coach-1 and coach-2) in the middle, see figure 3. The composition of such trains remains unchanged for its whole service duration; it is *stable* or time-invariant. At the highest abstraction level, the train is seen as 4 main modules interconnected through physical, electrical and data couplings. The upgrade of coaches and locomotives do not have the same frequency. The locomotives have lots of moving elements under high temperature, pressure and tension constraints (ex. engines) while the usage rate of the coaches is linked to users (doors, seats, etc). Therefore, at a first glance, in an attempt to describe the train, we can only think of its most stable building blocks i.e. its 4 modules. This is the first architecture of the train. The stability or time-invariance of these modules is linked to the study or observation interval of time. If this time interval is about 40 to 45 years (typical lifecycle of a train), only the most stable parts of the train can be observed as unchanged. All the other parts are changed in upgrading projects. All the neglected elements of the train's structure are considered to be *insignificant* according to the study time interval. But, some of these insignificant details become significant if we reduce the time interval to 10 years. Reducing time interval means looking after less stable elements of the system structural architecture with a new time scale. In this case, we could find that doors and seats for instance remain stable while the internal decoration of the coach is more frequently changed due to their obsolescence, aesthetic needs, etc. In this second level of the structural architecture we have not only to think of the last 4 modules, but we have to add the doors and seats to them. Obviously these new significant constituents of the structural architecture should be linked

to the first set of constituents in a specified way (doors and seats are added to the coaches while the electrical system is added to the locomotives). This iterative procedure of architecture identification can be followed iteratively. The analysis is stopped when a reasonably detailed level of system's structure components linked together is reached. Through this example, it can be seen that the architecture could be expressed and modelled through several levels. This is also to stay that invariance or essence of the system-of-interest does not mean anything unless the analyst defines what the purpose is and what consideration level does (s)he use. In other words *essence or time-invariance depends on the observation and study levels.* An element is significant for a given observation level while it should be neglected for another one. We will use this *relative stability* as the key for the system architecture definition, see figure. 3.

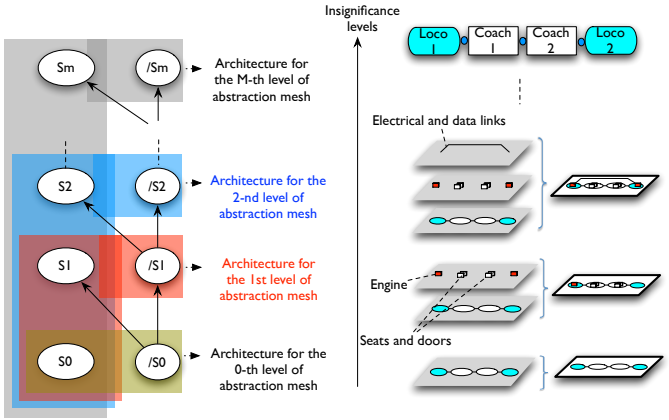

Fig. 3. Multi-level architecture

3.3 Modelling the architecture of a system

We present a modified version of the conceptual model of an architecture description suggested by the ISO/IEC/IEEE 42010 to go towards the relativity of the concept of essence and invariance. In Fig. 2, a small part of the original version of the standard is provided. The standard sees only one architecture associated with the system-of-interest. According to what we have shown in the last section, several architectures can be identified for a given system from a given point of view. That is the reason why the cardinality of the link between system and architecture is changed from 1 to 1..∗. These architectures in fact are various representations of one given system where its essence is understood and modelled according to various abstraction levels or study time interval. Sometimes, the essence should refer only to major and heavy trends of the system's dynamics, structure, behaviour, security, etc. while in other cases, analysts do need more details for the description of the concerned system. In a top-down modelling approach, only when the big picture is clear enough, analysts could tackle finer levels of analysis. This modelling process uses implicitly or explicitly the abstraction or filtering process, characterized here by a parameter called the *abstraction mesh*, $\mu \in \{0, ..., M\}$. The abstraction mesh is indexed by a non-negative integer with a highest limit of M. With a mesh equals to 0, we use the strongest abstraction filter models. It filters all those constituents, their parameters or behaviours that can be considered as insignificant for understanding the phenomena of interest. This process with this level of abstraction or filtering provides the main skeleton of the system and of its architecture with its most stable or significant constituents. Coming back to the concept of filtering of significant or insignificant means relevant or irrelevant to the abstraction level or to the chosen abstraction mesh $\mu = M$. The abstraction meshes act as criteria that allow to filter the complexity. By using the strongest abstraction mesh, $\mu = 0$, the set of system's constituents can then be subdivided into two subsets S_0 and $\bar{S_0}$ corresponding to the set of significant and insignificant ones constituents according to the abstraction mesh 0 . We then index the set of constituents C by the abstraction mesh level, see the right side of the Fig. 3:

$$C_0 = S_0 \cup \bar{S_0},\ C_1 = S_1 \cup \bar{S_1},\ ...,\ C_M = S_M \cup \bar{S_M}.$$

So at each step, by using a finer abstraction mesh, a newer set of significant constituents $\bar{S_{i+}}$ becomes available from the last set of insignificant constituents $\bar{S_i}$. Each layer is defined by the set of significant constituents, related to each other from one side and linked to the last layer too. Let take the layers. There is a mapping between these two layers μ and $\mu + 1$. This mapping is modeled by a matrix Ext_{r*s} where r is the number of constituents in layer μ and s is the number of constituents in layer $\mu + 1$.

According to the view of the architecture, the interpretation of these inter-layer links is different. For a structural architecture, these links define the way that the $(\mu + 1)$'s constituents are positioned on the μ's level constituents (seats are positioned inside coaches and engines inside locomotives). In a functional architecture, these links define the composition relation between detailed functions and their mother functions (in a IDEF0). The mapping between the constituents of a layer is modelled through a matrix called Int. Finally, the architecture of a system is defined hereafter:

Definition. The architecture of a system, filtered at the μ-th abstraction mesh, can then be defined as the stacking of successive layers of the system filtered from 0 to μ-th level: $A_\mu = \Gamma_{i=0}^{\mu}[S_i]$. Each layer is defined by a set of interconnected constituents. The first architecture level A_0 is defined by its inter-related constituents of C_0. The stacking function Γ is defined by the superposition of the significant constituents of layers of the system's significant constituents.•

The stacking function is the result of the successive *enrichment* of the most abstracted model of a system and is defined by the two mappings between the constituents of layers Ext and Int. These mappings reflects various dependencies among constituents of the architecture; they are the potential change propagation channels. The stacking function defines an implicit order relation (in a mathematical way) between the successively added layers. This order relation is called the *filtering dominance* of the architecture. The filtering dominance is an order relation Δ defined on a set $A = A_0, ..., A_M$ by its following properties possessing the algebraic properties of reflexivity, transitivities and anti-symmetry. The order relation $A_i \Delta A_j$ means that A_i dominates A_j (if there is not equality, then A_i

strictly dominates A_j). It is also to say that A_i defines a more stable view of the architecture than A_j.

3.4 Mechanisms of change propagation

The architecture description allows to define a first set of change propagation mechanisms:

(1) In a multi-layered architecture A_N, the privileged sense of change propagation is from the most stable layers A_0 toward the least stable ones, A_N.
(2) Within a given architecture layer, A_N, the privileged sense of change propagation is first within a given sub-system, $S_{N,i}$, and then towards the other coupled sub-systems, $S_{N,j}, j \in 1, .., |C_N|$.
(3) For a given multi-layered architecture, containing several abstraction layers, the privileged sense of change propagation is from the least detail layers $S_{N,i}$ towards the most detailed ones $S_{M,j}$ where $M > N$.

4. DISCUSSIONS AND PERSPECTIVES

This paper tackles the problem of complex system renewal. We consider complex systems with a long service duration within their hosting system. We studied renewal by defining renewal, change and architecture. A renewal project is launched to needs and expectations of some stakeholders and their translation into renewal requirements. We defined four levels of renewal: renovation, modernization, extension and conversion improving functions or the structure of the system or even both. Renewal is a set of acts aiming at integrating changes. The scope and depth of change propagation have to be answered. However, to do so, we rely on the use of the architecture because some changes could impact directly a very stable part of it (the outside shape of coaches for instance) while others could impact less stable parts such as the seats. We provided a brief survey to highlight that in system engineering, the architecture should model the essence of the system-of-interest. But the architecture should not provide a "flat model of a system" as the essence of the system depends to the abstraction level or to the observation time period. It is possible to think of the architecture as a two dimensional graph. The vertical dimension represents the significance. The concept of depth refers to this vertical dimension. The horizontal dimension refers to the internal structure of a layers that shows the constituents population. This is the scope. By modelling the architecture thanks to these two dimensions, we define the architecture as a stacking function. Based on this architecture definition, three main principles of change propagation were suggested. These principles allow to identify the best strategies of the change propagation scope and depth assessment.

In a practical situation, if designers are able to predict the impacts of a candidate-change, within a complex structure, it would help them to answer typical questions such as "validate or reject?" or "as-is or altered?". These questions are related not only to the feasibility or necessity of the change integration but also the cost and efforts required for it which are linked to the change integration project. The assessment of the resource capacity to perform the (required and derived) renewal acts can be done through traditional methods once the acts are identified. By searching the scope and depth of change propagations, the change integration project can be then analyzed simultaneously from the engineering and management points of view. Nevertheless, the engineering aspect of change integration in such complex systems has to be done through deeper concepts such as needs, requirements, logical and physical data, see (ANSI/EIA-632). These four sets are tightly connected together through design activities. The three propagation mechanisms can be therefore used as a guidance to find out the scope and depth of changes in terms of needs, requirements, logical and physical improvements. Our research is actually deeply involved in the modelling of these dependencies.

REFERENCES

Clements, P.C. (1996). Eighth International Workshop on Software Specification and Design , A Survey of Architecture Description Languages. March.

Crawley, E., Weck, O.D., Eppinger, S., Magee, C., Moses, J., Seering, W., Schindall, J., Wallace, D., and Whitney, D. (2004). Engineering systems monograph.

Edwards, P. (2003). Infrastructure and modernity: Force, time, and social organization in the history of sociotechnical systems. *Modernity and technology*.

ISO-IEC-IEEE (2011). Ingénierie des systèmes et des logiciels – Description de l'architecture. Tech. report.

ISO/IEC/IEEE-15288 (2008). Systems and software engineering - System life cycle processes. Tech. report.

ANSI/EIA-632 (1999). Processes for Engineering a System. Tech. report.

Krob, D. (2007). Eléments d'architecture des systèmes complexes. 1–20.

Le Moigne, J.L. (1990). *La modélisation des systèmes complexes*. Dunod edition.

Marsh (1997). Critical foundations, protecting America's infrastructures. Technical report.

Peerenboom, J. (2001). Infrastructure Interdependencies: Overview of Concepts and Terminology.

Rinaldi, S., Peerenboom, J., and Kelly, T. (2001). Identifying, understanding, and analyzing critical infrastructure interdependencies. *Control Systems, IEEE*.

Simon, H. (1962). The Architecture of Complexity. In *Proc. of the American Philosophical Society*, 467–482.

Ulrich, K. (1995). The role of product architecture in the manufacturing firm. *Research Policy*, 24(3), 419–440.

Eckert CM, Clarkson PJ, Zanker W (2004). Change and customisation in complex engineering domains. *Res Eng Des*, 15(1), 1–21

Jarratt TAW, Eckert CM, Caldwell NHM, Clarkson PJ (2011). Engineering change: an overview and perspective on the literature. *Res Eng Des*, 22, 103–124

Zolghadri, M., Couffin, F., Cheutet, V., Bourcier, C., Affonso, R., and Leclaire, P. (2013). Structuring systems and related research challenges. In *6th Int. Conf. MCPL*.