Emergent properties and requirements evolution in engineering systems and a roadmap

Jia Luo, Abd-El-Kader Sahraoui, Ali Hessami

Emergent Properties in Engineering Systems and a Roadmap

Jia Luo*, Abd-El-Kader Sahraoui

LAAS-CNRS, 7 avenue du Colonel Roche, F-31400 Toulouse,

Université of Toulouse, Université Jean Jaures, LAAS Toulouse

Waseda University, Japan on leave to LAAS-CNRS

Ali G. Hessami
London City University,
Beijing Jiaotong University ,Vega Systems, UK

Abstract: **Emergence and emergent properties have been a popular subject in systems of systems design. This work is an extension of Hinton's work on "under-specification". Even though emergence is a universal phenomenon that can hardly be modeled, there are still attempts to understand, propose solution what is known by non-desired emergent properties in industrial systems design. This paper focuses also on under specification and impact of requirement evolution. Some approaches even appear to be infeasible as in natural systems; however industrial systems are designed following various methods and techniques and the interface to user is often poorly specified and developed. In that respect, interfaces between systems and human are very hard to model as emergent behavior is hardly predicted and hence largely unknown. The paper attempts at characterizing such properties and initiate the debate on a roadmap for such research to support case studies and applications .**

Keywords : emergence, emergent properties, requirement evolution , systems of systems, safety

I. Introduction to characterizing emergent properties

Problem statement: First we revisit a definition that addresses the emergent properties proposed by Johnson [1] "*Emergent properties' represent one of the most significant challenges for the engineering of complex systems. They can be thought of as unexpected behaviors that stem from interaction between the components of an application and their environment. In some contexts, emergent proper-ties can be beneficial; users adapt products to support tasks that designers never intended. They can also be harmful if they undermine important safety requirements. There is, however, considerable disagreement about the nature of 'emergent properties'. Some include almost any unexpected properties exhibited by a complex system. Others refer to emergent properties when an application exhibits behaviors that cannot be identified through functional decomposition.*"

More simply put, the systems is not sum of parts since safe components do not give rise to a safe system; similarly in operations research field a global optimization does not arise from local optimization of the components and conversely the local optima does not ensure a global optimum.

However, we would challenge the constrained definition of emergence given by Johnson [1]. Emergence is universally the key characterizing feature of a purposeful aggregation of parts often with specialised functions and interworking topological relationship as a system. Given a system exists within a hierarchy of supra and sub-systems, there's a fundamental level below which the phenomenon of emergence cannot tangibly take place. . It is a serious misunderstanding of nature and scope of emergence to restrict it simply to undesirable properties and behaviors of a product, system or process. A systems based definition of emergence is given in the Karcanias and Hessami [2, 3] paper that for ease of reference is repeated and extended here.

Class: Emerging Property	
Attributes:	• A physical or virtual feature arising from a whole system • Not present in constituents alone and is not reducible to the parts • May be physical or virtual • May not be discernable to the observer • Has varying degrees of strength currently viewed as weak and strong
Operations:	• Is context dependent • Is lost when the whole is taken apart • Is weakened or lost when the whole is at fault (in constituent or topology) • Is mainly dependent on critical constituents

This more holistic view of emergence and emergent properties is consistent with Plato's Cosmology [4] and Aristotle's Metaphysics and his theory of universals [5]. Emergence relates to the irreducible collective properties arising from the properties of interworking parts and has no notion of observer preferences, desirability or otherwise as a universal phenomenon. This collective property cannot be reduced to the properties of the constituents hence Aristotle's view that "*the whole is something besides the parts*".

The emergent property is not addressed explicitly in EIA- 632 [6] as all requirements don't mention explicitly but we can see , however, from all requirements related to verification and validation process that such intent is embedded in the outcome for each requirements. Let take requirement 24 risks analysis in standards ANSI EIA-632 "The developer shall perform risk analyses to develop risk management strategies, support management of risks, and support decision making" with one task among five related tasks to consider.
.

II. Relating Emergent properties and safety

The emergence issue has been addressed at theoretical and general level and also among community of systems of systems. Safety is a facet of emergence as argued under the holistic perspective earlier. It sits alongside other emergent properties of a product, process or system such as reliability, security, maintainability, quality, sustainability and a whole host of other emergent features. Safety is also a desirable positive attribute or quality that emerges from a comprehensive life-cycle based set of activities, processes and materials. The interpretation given by Johnson [1] tend to relate to undesirable hazardous states that are violations of safety in principle. The challenge is to devise processes, materials and employ requisite levels of competence by the human resources deployed to specify, design, develop and deploy systems that do not suffer from unacceptable risk of harm due to known and unknown hazardous states. Much of such knowledge is codified within the safety standards even though emergence of safe performance cannot be guaranteed but only assured.

III. UNDER SPECIFICATION

In his work, under-specification, composition and emergent properties, Hinton considers emergent properties, mainly undesired behaviors, are often of the result of under-specification of the system or assumptions about made about the environment.

A. *Under-specification, requirements evolution and emergence*.

In systems engineering practice, a non requirement can lead to a desired or undesired behavior. In recent literature, the evolution of requirements during system development, among other issues, reflects the changing needs of system stakeholders, organization and work environment. Software projects begin with unclear defined, fuzzy, and incomplete requirements .The sources of changes may comes from dynamic environments (Buren and Cook, 1998).

1.1 The effects of Requirements Volatility

Previous study of RV has only focused on the impact of requirements volatility on software productivity

The first research question addressed was: is the degree of requirements volatility negatively associated with software project schedule and cost performance? There are three major dimensions that were exploited to investigate and explain requirements volatility: potential for changes, requirements instability and requirements diversity (the extent to witch stakeholders disagree among themselves deciding in requirements).

The second research question addressed by these studies was: what are the requirements engineering practices that contribute to the volatility in software requirements?

- Statistical study of Requirements volatility

In [7] , a proposal for roadmap and issues in systems engineering and present their findings in terms of the change process model, the change request arrival rate, the requirements volatility measure, and taxonomy; they present that the rate of change requests increased sharply when requirements analysis and documents reviews (i.e. requirements specification, feature proposal, and functional specification) were being completed. Most of the requests resulted in additions and deletions of requirements. The arrival of change requests decreased as the project was getting closer to the end of its lifecycle. However the rate of change requests increased again during the end of detailed design review and system integration testing. And they presented that the only high peak (of the rate of volatility) was at the end of requirements analysis stage and at the beginning of the design stage.

1.2 Defect density of Requirements volatility

In an ideal situation the requirements for a software system should be completely and unambiguously determined before design, coding and testing take place.

Some authors present the influence of requirement changes at different times by presenting the consequences of software additions, removal and modifications.

Defect density is an important measure of software quality. Many studies suggest that changes to the requirements specification also have a significant impact on density of defect. Requirements volatility is a measure of how much program's requirements change once coding begins.

Requirements specifications are often written in natural language. Even when more precise techniques are used these specifications tend to change as program development and testing progresses.

B. Extension of the approach

Our approach and contribution will focus mainly on impact of requirement change and development a methodology for requirements change. This will be carried on the basis of

- traceability model
- Configuration management model
- A formal framework for the requirement change

1.3 General approach

We are investigating many approaches to such issue. However the global approach is thought as an operational view as illustrated by the following figure. The formal basis for such approach is not tackled yet but some items are thought to be useful and to be discussed in latter section.

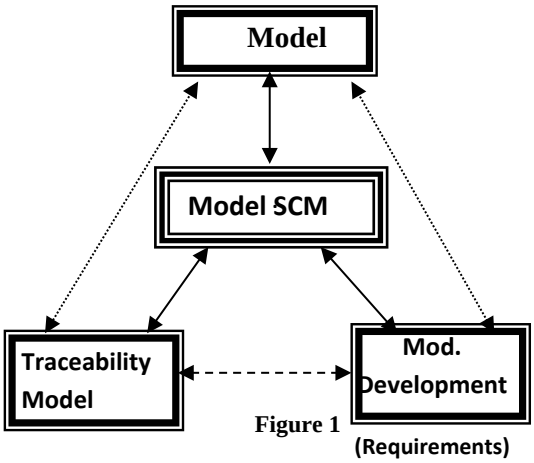

Figure 1

This preliminary approach is systems engineering context oriented as presented in the previous part (cf. part 2) and characterised by the interaction by four models; these involve respective processes. Our concern is the development of the change model and its interaction with all other models. We will present the traceability model dynamics that have been used in earlier and make abstraction of the development and system configuration management models as their basic characteristics are known for long time.

We know that any requirement change will concern and trigger all four models. In our first approach we will be concerned the change, traceability and development models. However, some principles will guide towards the deepening of the approach as future work will focus mainly on refining the approach.

- Any change request either at any step of development model suppose the availability of a traceability model.

- A change request for an operation module will necessarily require tracing back the original requirement

- Make distinction between functional and non function requirements

- Identify security/safety requirements

- create link between associated function and safety requirements

***** Traceability model

As discussed earlier, providing traceability of requirements to their sources and the outputs of the system development process can be along several dimensions. Different stakeholders contribute to the capture and use of traceability information, often with different perspectives. A user has a different vision from an audit specialist, a system designer or a validation engineer. Some typical questions are often asked:

What are the systems components that are affected by a specific requirement?

Why are the components affected by such requirements?

How are the components affected by such requirement?

What are the sources of a low level requirement?

Why and how two requirements are related?

And so on …

An object can belong to one of the following classes: requirement, design, components, system/subsystem, etc. Attributes and operations (activities) are associated with each class, subclass.

Sources are all available information as documents, phone calls, E-mail about the object lifecycle. Traceability concerning specific decision made can be found through the relation *documents*.

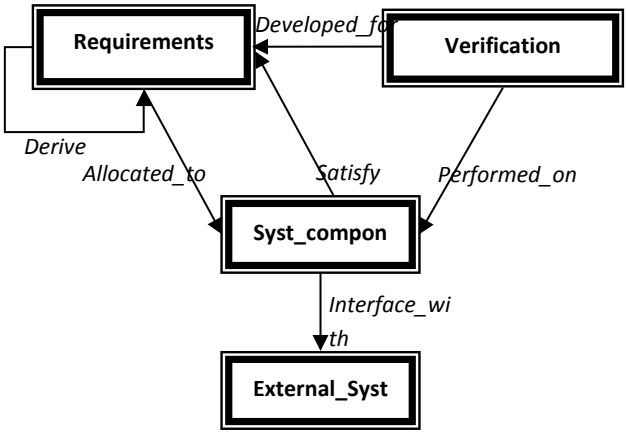

Figure 2

We can hence use this traceability model to identify any link that may be subject and constrained by a requirements change.

- Formal framework

In this part we focus to change requirements of a system without affecting safety. What is desired is to apply a modification to a requirement whilst keeping system safe. In this aim we use the safety requirements when we want to apply any modification on our system, and we consider that we have formal safety requirements.

A requirements can be describes by many model, we haven't the same representation for requirements in each discipline. We have much representation for requirements. In this paper we have a formal representation of requirements to guarantee the coherence between all requirements such as each requirement have a number, a description and other traceable attributes.

We can find a Change Process Model, this process is the responsibility of a project manager throughout the development life cycle, and they present four main phases of the change request process:

- **Phase 1: change request initialisation:** this is the initial phase of the change process where any project engineer or development team members can submit a proposed change and enter the change request into project database.
- **Phase 2: Change Request Validation and Evaluation**: the purpose of this phase is to validate the change request form in terms of detailed description of the proposed change, its impact on schedule.
- **Phase 3: change implementation**: this phase starts when the proposed change has been accepted and approved and the change becomes part of the system development.
- **Phase 4: change verification:** the objective of this phase is to verify that the change was made correctly. In this paper we focus on the first phase.

- The approach revisited

When we consider the change of requirements, we focus mainly in this part on three types of changes:

- Requirements Addition

- Requirements Deletion

- Requirements Modification

We are mainly concerned for the third type of change, we consider that we apply a modification for requirements on implementation but in order to take the decision to take a modification, we must apply 5 levels:

Level 1: utilize the phase 1 of the change process model to know the origin, the cause of change.

Level 2: know the corresponding requirements from our specification.

Level 3: apply a traceability model to know all requirements that are in relation with requirement to be changed that we want to change.

Level 4: know all safety requirements are in relation with all requirements that are in relation with our modification.

Level 5: study of the deployment and of the propagation of this modification, and we the risk if we apply this modification on the system, environment…etc.

After this study that is dividing in 5 levels, we can decide if we can to apply this modification on an implementation or not

IV. A ROADMAP

A. Introduction

We propose four areas that we plan to tackle in such road map. We propose to define the problem, propose the research approach and forecast the expected results ; an overviews of key areas was largely mentioned in [8]. These four areas are

- Dependability in supply chain

- systems of systems and composite systems, methodology and tools [9]

- Co-simulation issues

- Predicting Emergent properties

B. Dependability in supply chain

- Definition of the Problem : Emergent properties arising in supply chain (SC) is very common as any system and mainly in SC dealing with critical products [10].
- Research approach : The SC is highly dependent on SC architecture, we propose a formal approach to structure such architecture and mainly Petri nets in order to ensure dependability; stochastic colored Petri nets will be used.
- Expected results : the environment will enable simulation at both high and low level of abstraction. To enable the SC architect to plan all possible redundant items ranging from producers, providers, distributors/transporters

C. Systems of systems , methodology , methods and tools support

- Definition of the Problem : There is a need of methodology for SoS design, many approaches have been proposed but mostly relying on composing traditional approach. In view of the emergent property and safety issue, there is open field for such challenge. Theoretical basis is still needed
- Research approach : . But the methodology can be made universal to be adapted to each application : health care, transport, supply chain , embedded systems.
- Expected results : Specific domain notation and associated tools with corresponding theoretic basis are the possible outcomes

D. Co-simulation issues

- Definition of the Problem : Simulation is now a universal tool for design and assessing solutions for each domain ranging from mechanical , synchronous asynchronous, and all type of technologies and disciplines; the need for co-simulation
- Research approach : The possible approach is instead of using fusion model ; we may go to universal models of computation
- Expected results : Interfaces cause many problems when conducting with many models and Surely that an emergent properties can arise only through composting systems and hence , we neglected often when composing systems to model interfaces and development the functional mock-up interface with such criteria on interface may lead to see how classical approach with Cosimate failed on that aspects

E. Predicting emergent properties

- Definition of the Problem : there was a tentative approach in [11] to detect undesirable emergent properties. Emergent properties is the key element in complex system analysis. Detecting and predicting emergent properties is highly based on the knowledge we have about the systems, its requirement and traceability model.
- Research approach: We need to characterize such emergent properties and focus only possible weak-level properties. A Traceability model of the system can be enabled only if semantics of the model are well defined and neglect syntactic link between entities [12] . In that respect, the design of systems must be revisited so that safety requirements cannot be violated as may use “obligation proofs “ like as in VDM when de composing systems and preserving properties as invariant safety properties
- Expected results : Methodology to predict emergent properties by specific tools apart from simulation tools . Interface modeling between interacting systems can be traced back.

V. CONCLUSION

An overview has been given about requirements issues and their impact on emergent properties. A tentative roadmap has been given on certain applications related to systems of systems in general and to emergent properties in particular.

REFERENCES

[1] C.W. Johnson : What are emergent properties and how do they affect the engineering of complex systems?. Reliability Engineering & System Safety journal, Vol 91, No.12, 2006, pages 1475-1481.

[2] N. Karcanias, A.G. Hessami : System of systems and emergence Part1 : principles and framework. IEEE 4th international conference on emerging trends in engineering and technology ,2011. Pages 27-32

[3] N. Karcanias, A.G. Hessami : System of systems and emergence Part2 :synergetic effects and emergence. IEEE 4th international conference on emerging trends in engineering and technology,2011. Pages 33-38

[4] Cornford F.M., “Plato’s Cosmology: The Timaeus of Platos translated with a Running Commentary”, Routledge and Kegan Paul Ltd. 1966.

[5] Kenny, A. , “A New History of Western Philosophy, Volume 1, Ancient Philosophy”. Oxford University Press, ISBN 0-19-875272-5, 2006.

[6] EIA -632 : Processes to engineer a system. ANSI standards 2004

[7] A.E.K Sahraoui, D. Buede, A.Sage : Systems engineering research. International journal on systems science and systems engineering. Vol.17, N°3, pp.319-333, September 2008.

[8] M. Jamshidi : introduction to Systems of Systems., John and Wiley, 2009

[9] J. Black, . P. Koopman :System Safety as an Emergent Property in Composite Systems. IEEE/IFIP InternationalConference on Dependable Systems & Networks, 2009. DSN , June 2009, Pages369 - 378.

[10] B. Behdani : Evaluation of paradigms for modeling supply chains as complex sociotechnical systems. Winter simulation conference, 2012

[11] S. Ferreira, M. Faezipour, H. Corley : Defining and addressing the risk of undesirable emergent properties. IEEE International Systems Conference (SysCon), April 2013 .Pages 836 - 840

[12] Sahraoui, AEK "Requirements Traceability Issues: Generic Model, Methodology And Formal Basis", ; International Journal of Information Technology and Decision Making, 2005, pp.59-80.