Capability Maturity Model for Collaborative Networks based on Extended Axiomatic Design Theory

Hadi Kandjani and Peter Bernus

School of ICT & Centre for Enterprise Architecture Research and Management, Griffith University, Brisbane, Australia, {h.kandjani, p.bernus}@griffith.edu.au

Abstract The paradigm of forming and sustaining Collaborative Networks as environments that create Virtual Organisations (VOs) assumes that effective (and efficient) enterprise engineering (EE) capabilities and processes are available. However, these processes are only effective if they produce VOs which have sufficiently limited complexity, because as complexity grows, the VO's behaviour becomes increasingly harder to predict under all circumstances. This paper proposes the use of EE methods based on Extended Axiomatic Design Theory to limit the complexity of VOs – and of the CN itself. We introduce process- and people capability maturity levels, whereupon higher maturity implies higher probability of success of CNs in creating and maintaining VOs, and success of the VOs themselves, and formulate strategies for capability-improvement, intended to achieve higher levels of EE maturity.

Keywords: Extended Axiomatic Design Theory, Process and People Capability Maturity Models, Concurrent Collaborative Networks Engineering.

1 Introduction

The establishment of Collaborative Networks which serve to create dynamic responses to market opportunities (through the creation of virtual organisations) emerged as a new paradigm for doing business [4]. The field raised a number of research questions, one of these (complexity management) is addressed below.

Two Collaborative Networks (CNs) with the same objectives and requirements may apply the same design methodology, but may still develop different virtual organisations (VOs), in terms of quality of models, processes, etc. as well as 'systemic' properties of the VOs. We argue that this is (at least partially) due to the difference in levels of enterprise engineering (EE) capability/maturity of the two CNs. Thus the same design methodology and requirements may lead to different results (a property called multifinality [6]).

Management literature calls this form of non-determinism 'path dependency' which arises because the result also depends on decisions that were made along the way of evolutionary history prior to the present state (see Liebowitz and Margolis *et al.* [13,14]). Part of this path dependency / multifinality is due to the evolutionary history of the CN which led to the present EE capability that is a determining factor in the CN's (and its VOs') effectiveness. Conversely, CNs as open systems need to be assumed to follow the principles of equifinality (or 'convergence'), *i.e.,* finding equally valid alternative ways of attaining the same end state if starting from a given

initial state (or from different initial states) (see von Bertalanffy [3]). The principle applies to open systems when a given end state is reached by many potential means. The intention of developing a (process and people) EE capability maturity model for CNs is to improve the understanding and to inform and direct the evolution of CNs.

The Software Engineering Institute's Capability Maturity Model Integration (CMMI) [4,19] is a process improvement reference model informing enterprises wishing to improve their process capabilities and organisation (and through it their performance). CMMI integrates multiple bodies of knowledge developed separately (software engineering, systems engineering, acquisition,...)[5]. (ISO/IEC 15504:2007 defines 'process capability' as "…the capability of a process to meet its purpose …" [20].) In our case, the purpose of the EE processes (ISO 15704:2000 [8,10]) is to design (and maintain for life) the CN and its VOs through their stages of evolution.

An important problem facing a CN is complexity, because uncontrolled complexity can cause undesired CN- and VO characteristics. A number of relevant complexity measures may be considered: (according to Lloyd [15] and interpreted by the authors [11]) these can be classified as those characterising the difficulty to describe a) the function, behaviour, and states of the system, b) the architecture (relationship between physical and functional structure), and c) the process to create the system. Categories a) and b) measure the complexity of the CN and of its VOs, and c) measures the complexity of VO creation and CN creation projects.

Axiomatic Design (AD) [18] claims to codify in a discipline-independent way what a 'best design' is, aiming to avoid unnecessary complexity. Li and Williams [11] refer to the possibility of using AD in EE, but to avoid complexity of category (c) we extended AD by introducing the Recursion Axiom stipulating that the system that designs the system must also obey the axioms of AD. The question arises: how to maximise the probability of the success of the EE practice used to design VOs/CNs?

Many EA researchers and practitioners fail to recognise that they are in fact applying methods and models derived from laws and theories of cybernetics. Cybernetics is a pluralistic theory (and interdisciplinary movement) of generic laws and theories of information processing. We observe that applied cybernetic laws and theories in EA lack harmony and our aim is to introduce *EA Cybernetics* as a field of EA research that harmonises, formalises and synthesises the results of cybernetics and demonstrate their applications in EA practice.

Stafford Beer [1] was perhaps the first person to apply cybernetics to management. However, EA [2] not only embraces the application of models, methods and theories of management & control, it also incorporates models, methods and theories of the service, engineering and production to design an effective enterprise, including resources, organisation, products etc. of the enterprise. Therefore, EA cybernetics is distinct from Management Cybernetics, but considers the management & engineering views of enterprises and demonstrates how to apply cybernetic laws and theories in Enterprise Architecture. EA is a discipline that invokes other disciplines to analyse, design, construct, maintain and evolve an enterprise throughout its life history to accomplish the enterprise's short and long term aims. One such worthy aim is of course long term viability and sustainability (threatened by untamed complexity). This is why the authors situate the presented analysis as a contribution to EA Cybernetics (aimed at building EE capability to reduce unnecessary complexity) – the kind that helps EE steer the enterprise to viable futures.

2 Axiomatic Design and Enterprise Engineering in Collaborative Networks

Axiomatic Design (AD) is a theory of complex systems (that can not be predicted for sure to always satisfy their functional requirements [18]). AD explains reasons of emerging complexity, and offers a formal design theory and two design axioms that system designs must satisfy to minimise complexity (measured by the probability that the structure always performs the function (category b)).

Axiom I: Independence Axiom [18]. 'The independence of Functional Requirements (FRs) must always be maintained.' (An FR_i is independent of others if there exist 'design parameters' [DP] so that if changing one FR_i only one DP_i must change, whereupon [FR] = [[A]] * [DP]. Here [FR] is the vector of FRs, [DP] is the vector of DPs and [[A]] is the matrix mapping DPs to FRs. If [[A]] is diagonal matrix then the design is uncoupled (full independence is achieved). If [[A]] is triangular then the design is decoupled (the implementation process is 'serialisable'). Otherwise the design is coupled (the implementation process of DPs is not 'serialisable').

Axiom II: Information Axiom [18] 'Out of the designs that satisfy Axiom I that design is best which has the minimal information content.' (Suh defined information content (IC) as the negative logarithm of the 'probability of success'.)

We see Axioms I and II as intending to minimise the complexity of the system's architecture (complexity type 'b') and can be used to design less complex CNs and VOs. We observe that complexity type 'c' (of CNs and VOs) is not automatically addressed by using AD into EE practice, so we recently proposed [12] that Axioms I & II must also be applied to projects creating Vos / changing the CN. This is expressed in a 'recursion' axiom: change projects (as a system of systems) not only must follow Axioms I & II, but they themselves need to be 'axiomatically designed'.

Axiom III: Recursion Axiom [12]: 'The system that designs a system must satisfy the two Axioms of design.' NB**:** systems that satisfy Axioms I & II do not necessarily satisfy Axiom III: while at a given moment in time in its life history a system may be considered moderately complex, this system may be very hard to create or change.

Example: Denote three consecutive stages of an evolving CN as S_1, S_2 and S_3. In stage S_1 the CN is operating and has a design satisfyingi Axioms I & II. Let S_2 be the stage of change (S_2 is the original CN extended by a change project P). The task of P is to create S_3. When S_1 creates P it can mandate that P *use* Axioms I & II to design S_3. However, P (and thereby S_2) may not satisfy Axioms I & II, so P can be more complex than necessary; even if its mandate is to design S_3, the likelihood of success of this endeavour may be less than desired, *i.e.* P does not satisfy Axioms I & II (even if it applies them to design S_3). Axiom III states that S_1 not only must mandate that P use Axioms I & II, but S_1 must design P *using* Axioms I & II (in the interest of successful evolution). NB. the change system of systems is called the set of 'supporting systems' in ISO 15288 [9]. Thus, "among the design processes that apply axioms I & II to design a system, that is best which *itself* satisfies axioms I & II" [12].

If a CN wishes to reduce its own complexity and to subsequently maintain reduced complexity through life, it may wish to adopt AD as a strategy. Therefore it is legitimate to ask whether the CN is ready to use such practices and to increase the probability of success (i.e., what is its 'AD process maturity'). A model of such maturity is a type of 'EE Capability Maturity Model'. To differentiate this from other CMMs the authors abbreviate it as EE-AD-CMM. Given the equifinality and multifinality properties of enterprises, we do not claim that such maturity model would be useful for CNs that have no desire to apply complexity reduction measures or use AD practices. The value of this EE-AD-CMM is that untamed complexity is a dangerous phenomenon that may put at risk not only the VOs but the CN itself.

3   Enterprise Engineering Maturity Based on Extended Axiomatic Design Theory

Several authors have developed Capability Maturity Models and formulated strategies to increase the level of engineering capability and maturity of the enterprise. Hintersteiner and Zimmerman [7] developed an Axiomatic Design Capability Maturity Model to provide a roadmap for implementing AD practices in systems engineering. What this paper is proposing is two qualitatively different additional capabilities: 1) the enterprise (in this case the CN) not only has to design 'systems' (such as VOs), but it also has to design, or re-design, itself, *i.e.* the CN needs *self-design capability*, and 2) the CN also has to have the *capability to obey the third design axiom.*

Our proposed model extends the set of necessary EE capabilities and associated maturity levels, and the model is intended to help enterprises to avoid complexity of categories b) and c). Importantly, category c) complexity of CNs and VOs has not been addressed in previous maturity models. Note: category a) complexity can only be mitigated, but not avoided, unless we change what functions we expect from the systems of interest (i.e., the CN or its VOs).

3.1 Enterprise Engineering Process Maturity

Level 0:   Not Performed (AD is not applied anywhere)
Level 1:   Informal (non-institutionalised practice or pilot project applying AD) This is the starting point of CN-level intention to practice AD in designing CN entities. Pilot projects help reduce the risk of failure and the impact on the CN and help create best practice.
Level 2:   Defined (a program is defined to apply AD in all VO design projects) Here there is commitment of CN management (or of participants) to apply AD principles (projects observe AD principles when developing the CN and designing VOs). The practice is *mandated* (ensuring to include experts / staff, resources and appropriate processes and tools to support AD). This level is process-following: AD is followed, but not in a tacit manner.

Level 3:   Recursively-Defined   On Level 3 there is awareness by CN participants of ineffectiveness and inefficiencies of existing CN and VO design practice. Thus the CN is in a stage of aiming at self-improvement.

Level 4:   Established Extended AD practice A well-defined CN and VO design methodology is used in all VO design projects, and all CN change projects.

Level 5:   Continuously Improving   On Level 5 the performance of the current AD-based CN design methodology is monitored and continuously improved.

3.2 Enterprise Engineering People Maturity

The CN and its participants can not acquire, attain, practice and institutionalise AD capability and achieve higher levels of maturity in one day, so we introduce the AD based People CMM. Each level of the model focuses on developing enterprise environment and culture supporting distributed design process in EE practice.

Level 0:   No AD Competency   An enterprise at Level 0 has no AD competency in the network or in participating organisations.

Level 1:   AD Competency Acquisition and Outsourcing At Level 1 there is awareness of AD and its advantages by a subset of network participants. The CN may use a pilot project to assess CN-wide adoption possibilities. Some AD skills & competencies may have to be acquired from outside of the CN.

Level 2:   AD Competency Training At Level 2 the CN may be in training mode and use a CN-wide education program to develop AD-competencies of key staff. Projects may use mentors to monitor and manage the new type of design processes.

Level 3:   AD Competency Building At Level 3 the CN runs projects to gain CN-wide experience of AD processes.

Level 4:   AD Competency Improvement   Here the CN benefits from empowered and experienced workforce openly collaborating and having mastered AD-based EE processes. The CN at this level still needs dedicated AD experts and a reward- and motivation system for the rest of the workforce to successfully manage enterprise engineering practices.

Level 5:   AD Tacit Competency / Continuous Improvement At Level 5 enterprises do not need enterprise engineering experts to manage and control enterprise engineering practice: advantages of AD-based CN design methodology are obvious to the entire enterprise and all workforce throughout the CN, its members have experienced them and tacitly apply them in a collaborative and distributed way. This means that while practitioners' knowledge of this methodology is available in explicit form, when practicing it, the methodology if followed in a tacit way. From an external observer's point of view practitioners appear to be following a methodology most of the time, but the behaviour does not appear to be strictly following a process as practitioners can adjust, optimise, or tailor the methodology without diverting from its original intent and principles: the workforce at this level is self-motivated and self-organised in AD.

4 Concurrent Enterprise Engineering: Ability to Self-Design & Self-Organise

Pennell and Winner [16] describe concurrent engineering as "*a systematic approach to the integrated, concurrent design of products and their related processes, including, manufacturing and support. This approach is intended to cause the developers from the very outset to consider all elements of the product life cycle, from conception to disposal, including cost, schedule, quality and user requirements.*" In other words, concurrent engineering is an approach in which all phases of engineering potentially operate simultaneously. Product and process design run in parallel and design processes are closely coordinated to achieve optimal matching of requirements for cost, quality, and effective delivery.

The concept can be applied to CN design: when moving towards level 5 of the EE process CMM. At the same time, Level 5 of the EE people CMM emphasises collaboration and teamwork as an integral part of concurrent EE. At this level, all components of the ability to self-organise in terms of team cooperation are visible (*cf.* 'the 7Cs of collaboration in concurrent engineering' [17]):

- flexible, unplanned and continuous **c**ollaboration,
- **c**ommitment to meet the goals,
- **c**ommunication (exchange of information),
- ability to make **c**ompromises,
- **c**onsensus in spite of disagreement,
- **c**oordination (managing interdependencies between activities), and
- **c**ontinuous improvements in order to increase productivity and reduce process times.

5 Conclusion

This article proposed a Capability Maturity Model based on extended axiomatic design for the use of capability assessment and strategy making for Collaborative Networks. This maturity model includes both Process- and People maturity levels. The model is based on the promise of Extended Axiomatic Design Theory which aims at guaranteeing the probability of success of CNs and of the virtual organisations which emerge from the CN. These models may be used as a roadmap for incorporating extended axiomatic design practices and techniques into enterprise engineering in general and into the practice of Collaborative Networks and its member enterprises.

Part of the maturity model includes the ability of the CN to self-design / improve as it ascends to higher levels of maturity. According to this model, a CN ultimately attains process capabilities to simultaneously maintain, manage or change the CN design methodology itself, as well as to apply the methodology to all CN entities (including the CN itself). Based on the maturity model proposed in this paper, dedicated AD-based enterprise engineering staff and resources are no longer needed

as on Level 5 the AD-based enterprise engineering methodology is well-entrenched and consistently performed by everyone in the CN.

We incorporated the extension of axiomatic design theory to CN design practice and the results were demonstrated in terms of 6 different maturity levels – each with different levels of process and people capability / maturity. For future research we suggest the verification and validation of these developed models through empirical applications, case study and other relevant methods. In the final section, the article also briefly introduced the notion of concurrent CN engineering process capability but details have been relegated to future research.

References

1. Beer, S. Cybernetics and Management. English Universities Press, London (1959)
2. Bernus, P., Nemes, L., Smidt, G. (Eds): Handbook on Enterprise Architecture. Springer, Berlin (2003)
3. Bertalanffy, L. General Systems Theory: Foundations, development, applications. Braziller, New York (1968).
4. Camarinha-Matos, L.M., Afsarmanesh, H., Löh, H., Sturm, F., Ollus, M. A Strategic Roadmap for Advanced Virtual Organizations. In L. Camarinha-Matos and H. Afsarmanesh Eds. Collaborative Networked Organizations. pp289-312. Springer, Berlin (2004)
5. Chrissis, M., Konrad, M., Shrum, S. CMMI Guidlines for Process Integration and Product Improvement. Person Education, Boston (2003)
6. Hanson, B., General systems theory beginning with wholes. Taylor&Francis, London (1995)
7. Hintersteiner, J., Zimmerman, R.C. Implementing Axiomatic Design in the Systems Engineering Process: An Axiomatic Design Capability Maturity Model. Proc ICAD'2000, Axiomatic Design Inc., Cambridge, MA. pp9-17 (2000)
8. IITForce. GERAM: Generalised enterprise reference architecture and methodology (V1.6.3) (1999). [Also in Handbook on Enterprise Architecture. Springer, Berlin. pp22-64 (2003)]
9. ISO15288:2008. System Life Cycle Processes. ISO, Geneva (2000, rev. 2008)
10. ISO15704:2000, Industrial automation systems -- Requirements for enterprise-reference architectures and methodologies. ISO Geneva (2000, Amd 1: 2005).
11. Kandjani, H., Bernus, P. Engineering Self-Designing Enterprises as Complex Systems Using Extended Axiomatic Design Theory. Proc 18th IFAC World Congress (to appear)
12. Li, H., Williams, Th. A formalization and extension of the Purdue enterprise reference architecture and the Purdue methodology. TR 158 Purdue Lab. of Applied Industrial Control. Purdue University, West Lafayette (1994)
13. Liebowitz, S. Margolis, S. Path dependence, lock-in, and history. *Journal of Law Economics & Organization.* 11(1): pp205-226 (1995)
14. Liebowitz, S., Margolis, S. Encyclopedia of Law and Economics. E Cheltenham (2000)
15. Lloyd, S. Measures of complexity: a nonexhaustive list. IEEE Ctrl Sys Magazine 21(4) pp7-8 (2000)
16. Pennell, J., Winner, R. Concurrent engineering: practices and prospects, Proc Globeco'89, IEEE pp647-655 (1989)
17. Prasad, B., Concurrent engineering fundamentals- Integrated product and process organization, Prentice Hall, New Jersey (1996)
18. Suh N.P., The Principles of Design. Oxford Univ Press, New York (1990)
19. Team, CP. CMMI for Development, version 1.2. Preface, SEI, CMU (2006)

20. Van Loon, H., Process Assessment and ISO/IEC 15504: a Reference Book. Springer, Berlin (2007).