Association for Information Systems

AIS Electronic Library (AISeL)

ACIS 2013 Proceedings

Australasian (ACIS)

2013

Crossing the Communication Barrier in Global Software Development Projects via Global Software Development Brokers

Lian Wen
Griffith University, l.wen@griffith.edu.au

Hadi Kandjani
School of ICT, Griffith University, esmailzadehh@yahoo.com

Peter Bernus
Griffith University, p.bernus@griffith.edu.au

ACIS 2013
RMIT MELBOURNE

Information Systems: Transforming the Future

24th Australasian Conference on Information Systems, 4-6 December 2013, Melbourne

Proudly sponsored by

ACIS 2013 Principal Sponsor

Crossing the Communication Barrier in Global Software Development Projects via Global Software Development Brokers

Lian Wen
Hadi Kandjani
Peter Bernus
Centre for Enterprise Architecture Research and Management (CEARM)
School of ICT, Griffith University, Brisbane, Australia
{ l.wen, h.kandjani, p.bernus}@griffith.edu.au

Abstract

As the key stakeholders in Global Software Development (GSD) projects are distributed across geographical locations, many GSD projects suffer from a communication barrier, which exists due to language-, cultural-, time zone (and possibly other) differences among key stakeholders. This barrier not only increases communication cost, it also decreases the efficiency and quality of stakeholder communication, adding extra risks to these projects, and decreasing the probability of success. So far, there is no simple solution to this problem. Using the Collaborative Networks paradigm, this paper introduces the concept of 'Global Software Development Collaborative Network' (GSD-CN) as a formal model to analyse communication cost and quality. The paper proposes a new entity (role) called Global Software Development Broker (GSDB). We argue and demonstrate in an example that the proposed GSDBs will (a) simplify the network structure, (b) decrease communication cost, and (c) improve communication quality – consequently increasing the probability of success of GSD projects.

Keywords

Global Software Development Projects, Supply Networks, Communication Barrier, Virtual Brokerage

1 INTRODUCTION

Global Software Development assumes that distributed teams from different companies and from different geographical locations collaborate to design, manage and execute life cycle activities of a joint GSD project. A GSD project is a complex network of distributed processes, which by its nature increases the complexity of software processes through increasing the complexity of "project coordination (through temporal and geographical distances), and communication" (Šmite and Borzovs, 2008). Due to these barriers (such as language and cultural diversity, distance, differences in time zones between parties, etc.), many GSD projects suffer from very high cost of communication. For example, GSD project managers have to travel internationally to get the key people together to make sure that the most important information is communicated accurately and sufficiently. In fact, many GSD projects suffer from inaccurate and inadequate communication that may lead to project delays or even failure.

Part of this communication problem is due to the dynamic dependencies among components of GSD products (Cataldo, Bass, Herbsleb, and Bass, 2006), which causes excessive complexity in the design of GSD projects, including project planning for project development, which in turn gives rise to uncertainty and ambiguity. As a result, the emergent communication structure among diverse roles in the project does not always follow a pre-determined and planned communication structure (Marczak and Damian, 2011). Some researchers showed that a considerable amount of the complex communication in GSD is due to the design and architecture life cycle activities of GSD projects (Cataldo, Bass, Herbsleb, and Bass, 2007). This communication needs to be planned for and subsequently moderated by GSD project management, in a deliberate attempt to reducing ambiguity and facilitating coordination, which two have been identified by Sangwan *et al.* (2006) as critical success factors in GSD projects. So far, there is no efficient and effective solution to this problem apart from training and improving the communication skills of the key personnel in a GSD project. However, this method is only affordable for large organizations with large projects. For example, if the project's language is English, it is not realistic to expect all Chinese or Russian project managers to speak English fluently, nor can one expect that all customers will master the communication skills of foreign developers.

To discuss this problem, we define two abstract measurements: *communication cost* and *communication quality*.

The communication cost between two or more partners is intuitively defined as the cost (usually expressed as a financial measure) for successfully exchanging a unit amount of information – although other cost measures may also be valid, such as expenditure of mental energy.. This concept is different from Shannon's Channel Capacity (Cover 2006), which measures the amount of information that could be transferred through a given channel in a unit of time. The communication cost is determined by the communication capabilities (skills, knowledge, background, etc.) of the communication partners as well as the properties of the communication channel.

For example, let us imagine a software user trying to explain a problem to a member of the software supporting team. Their knowledge, background, and communication skills will definitely affect how quick the supporting member can understand the problem. Also, the communication method (the way they communicate through the phone, or email, or remote desktop connection) and the environment (are they in the same room or far away in different time zone?) will also affect the time used for the support person to understand the problem (which in turn can be translated to a selected cost measure, such as dollars).

Communication quality measures the difference between the expected communication result and the actual communication result. Ideally communication quality is always 100%, and participants keep communicating (perhaps at an excessive cost) until complete mutual understanding is achieved. However, due to several factors this is not always the case (e.g., partners putting up with incomplete communication due to lack of time or resources, or simply not knowing that mutual understanding has not been achieved).

It is well known that in software engineering, different stakeholders usually have different expectations of the same system. This could be caused by stakeholders having different viewpoints, but can also be caused by the poor quality of the communication. For example, it is known that many software requirement defects are caused by inefficient communication between the customers and developers.

We now turn to the analysis of the problem in GSD. Using the Collaborative Networks (CN) discipline (Camarinha-Matos 2004), we look at the business model of how Global Software Development is performed. According to the CN view, it is possible to create a network of enterprises (where an enterprise may become member of the network through satisfying some entry criteria, based on competencies and other selection criteria). Such a network has the capability to create for each assignment a 'Virtual Organisation (VO)' (in the simplest case a project enterprise), which has temporary existence, and for the external world appears to be a single well managed entity, as if it were a single enterprise. Such VOs may be truly temporary, but may have longer life span, e.g., they may aggregate an ongoing development programme and its multiple current and future projects, or may provide ongoing services for a long time.

Applying the CN paradigm as reference model to Global Software Engineering, we define a 'Global Software Development Collaborative Network' (GSD-CN) as a specialization of CNs, which is a formal model through which we can describe and analyze communication cost and quality. In a GSD-CN, each node represents a stakeholder and each link represents the communication channel between the connected stakeholders.

Šmite and Borzovs (2008), after studying the structure of some GSD projects, discovered that a major difference between a GSD project and traditional software development project is that there exists an obvious communication barrier, which significantly increases the communication cost and decrease the communication quality. Empirical studies (Boden and Avram 2009) show that key personnel with good communication skills can work as a "bridge" to cross the communication barrier between offices distributed in different areas of a software company. The existence of those bridges is essential for the success of those companies. However, people with the demanded talents are not always available for most software companies. It could also be considered as introducing suitable Facilitating Mechanism (FM) (Bosura and Scheepers, 2007) or Knowledge Borkers (Markus, 2001) working in a knowledge-based Artifact Network (AN) .

A typical CN consists of member enterprises and a Network Office, which acts as a broker and is responsible for the creation of VOs. We generalize this concept and define a new entity type called Global Software Development Broker (GSDB), which is a new type of stakeholders for a GSD-CN. A Network office in CNs is usually placed on the 'edge' of the network (i.e. to communicate with external stakeholders – being a communication gateway for the CN). As opposed to this, GSDB Brokers may be placed *within* the network and serve as a communication gateway between (among) network partitions. We argue and demonstrate in an example that the proposed GSDBs will (a) simplify the network structure by removing many poor quality communication channels, (b) decrease the total communication cost in the network, and (c) improve the overall communication quality for the network. As a result, it will increase the probability of success of GSD projects.

This result predicts that in future, with the continuous growth of GSD, the role of GSDB could emerge as a new career / type of enterprise to reduce cost and increase the success rate for GSD projects.

The paper is structured as follows: Section 2 provides a brief background of collaborative networks; Section 3 introduces the new framework to analyse the communication cost and communication quality in a GSD project; Section 4 explains the role of GSDB as a new stakeholder; Section 5 uses an example to illustrate how a GSDB could reduce the communication cost and improve the communication quality; and finally, some conclusions and future works are given in Section 6.

2 COLLABORATIVE NETWORKS

In this paper we use the emerging paradigm of collaborative networks (CNs) and apply it to GSD. The significance of CNs and maintaining them has dramatically increased for participant enterprises due to imperatives such as globalisation and the need for specialisation to be able to maintain excellence (Scherrer-Rathje, Arnoscht *et al*, 2009).

Creating and maintaining CNs which can create dynamic responses to market opportunities (by creating so-called virtual organizations) emerged as a new paradigm for doing business (Camarinha-Matos and Afsarmanesh, 2004). The main objective of a CN is to be able to conceive, create, configure (or re-configure), maintain and operate Virtual Organisations (VOs) for service or manufacturing, and to do so dynamically, effectively and efficiently in terms of time and cost, while meeting high quality standards (Kandjani and Bernus, 2011a).

CNs can be thought of as an evolution of supply chain integration made possible partly by the worldwide removal of numerous economic and trade obstacles and partly by the opportunities that pervasive information and communication technologies create. The opportunities are appealing because they promise the ability for enterprises to create rich and timely responses to the variety of demands required by the environment, whereupon none of the members have the ability to do so; in Ashby's terminology the CN is able to create the 'requisite variety' to manage the complexity of the environment's expectations, and to maintain a dynamic equilibrium with its environment (Ashby, 1956).

Kandjani and Bernus proposed the use of Enterprise Engineering (EE) methods e.g. GERAM (IFIP-IFAC-Task-Force, 1999; ISO15704, 2005; Bernus and Noran, 2010) based on Extended Axiomatic Design Theory (Kandjani and Bernus, 2011b; Suh, 2001; 2005) with the objective being to limit the complexity of VOs – and that of the CN itself (Kandjani and Bernus, 2011a; Kandjani, Wen and Bernus, 2012). Kandjani, Bernus and Wen (2012a; 2012b) demonstrated a method of reducing the structural complexity of GSD-CNs, thereby promoting the reduction of complexity of the creation of GSD projects as well as their cost of communication. The examples given (*ibid*) have shown how the distribution of the GSD project design task among virtual brokers reduces the complexity that any GSD project creation authority needs to see. The same work proposed the application of Kolmogorov complexity to estimate this complexity as well as (using virtual brokerages) to reduce the cost, and improve the quality, of communication among stakeholders (Kolmogorov, 1965; Cover and Thomas, 2006; Li, and Vitányi, 2008).

Cataldo and Herbsleb (2008) discuss the evolution of communication networks in GSD projects in order to address the "limits of the modular design strategy". Their analysis showed the emergence of a group of developers as the liaisons between GSD teams and geographical locations over time to handle the communication and coordination load across teams and locations. This idea is similar to the proposed virtual brokerages. Virtual brokerages could be actual enterprises, but may be VOs themselves (i.e. being created, operated, maintained and decommissioned by the Network), making the CN an evolving entity from its management point of view (Kandjani, Wen and Bernus, 2012).

The various connections, collaborations and co-operations among participant enterprises in a CN (with the associated processes, structures and information channels) may be built by partners themselves, or with the involvement of various (specialised) hubs. For example, GSD-CNs may set up brokerages that create GSD projects in order to have the requisite variety to deal with the complexity of the environment's needs and to improve the communication among GSD-CN members. However, other hubs may be built as well: e.g. Milanovic, Stefanovic *et al*. (2011) define Aggregator Hubs, Broker Hubs, Collaboration Hubs and Translator Hubs, and each of these may exist on various levels of the Supply-Chain Operations Reference-model (SCOR) – such as Scope, Configuration and Business Activity (Bolstorff, and Rosenbaum, 2007). However, to ensure the desired qualities (dynamic, effective, efficient…) in responding to the demands of the environment, the communication among the GSD-CN members can be a problem and a main hindrance.

Heuristic methods and trial and error in the creation of GSD projects give rise to time delays, unanticipated costs and unsatisfactory quality of the software development service. A GSD-CN member may have several connections to many other potential partners, but in any given situation (such as at the time of dynamically planning a GSD project) the probability of being able to successfully use these connections, when searching for suitable and available partners, is not 100 percent. Communication barriers introduce delays into the creation of the GSD project (thus in case of a complex project the cost of communication for these searches can escalate), and can also decrease the quality of communication.

Research methodology

This paper endeavours to solve the problem of the communication barrier in GSD projects using the collaborative network paradigm (Global Software Development Collaborative Network (GSD-CN)) and virtual brokerage. Thus, we propose that injecting a new role called 'Global Software Development Broker' (GSDB) into the GSD-CN has multiple benefits, including (a) simplifying the network structure, (b) decreasing communication cost, and (c) improving communication quality. The outcome of this research is expected to be useful to significantly improve the quality and increase the chance of success for GSD projects.

3 THEORETICAL MODELS FOR COMMUNICATION BARRIER

Global Software Development Communication Network

In a GSD project, each stakeholder is represented as a node. For example, a customer and a developer are represented by two different nodes. If there is a need to communicate between two stakeholders, a link is drawn between the two corresponding nodes; we call the link a communication channel. After identifying all communication channels among stakeholders, we could draw all the nodes and links, resulting in a network, which we call a 'Global Software Development Communication Network' (GSD-CN).

Figure 1 shows a simple GSD-CN. In this network, there are seven stakeholders. The Investor, Customer1, Customer2 are from a European country, while the other four are from China. Each line in the figure between two nodes means a communication channel and there can exist multiple communications between the two stakeholders through that channel.

The double line in the figure indicates the major communication barrier, which lies between the two groups. A communication that does not cross the major barrier is denoted by a thin black line, and a communication that crosses the major barrier is denoted by a thick gray line.

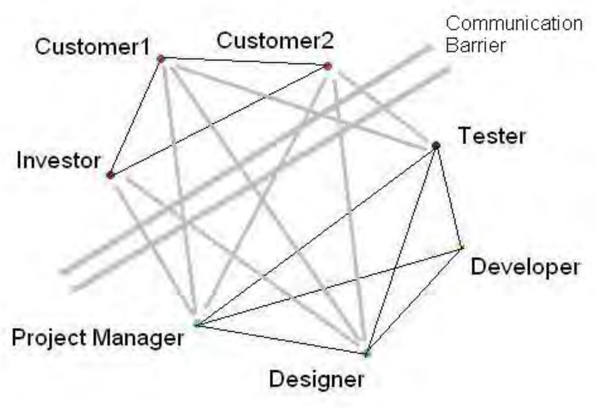

Figure 1. A simple GSD-CN

Communication Cost, Amount and Quality

Let N be a GSD-CN, x_1 and x_2 be two stakeholders, l be the communication channel between x_1 and x_2. Then there are three properties associated with the communication channel l. They are Unit Communication Cost (UCC), Communication Amount (CA), and Communication Quality (CQ).

The Unit Communication Cost (UCC) indicates how difficult it is for a unit piece of information to be communicated successfully through that channel. Usually, information is exchanged much easier between two people who speak the same language and have similar backgrounds. Also, the physical distance and the communication method between the two people will contribute to the difficulties of communication. For the purposes of this discussion we simply use an abstract unit to measure UCC, e.g. use a 'unit cost' of 1 to present very low cost and use 10 to represent that the cost is ten times higher. In a practical situation one would use a different measure, such as dollars / unit of information.

The Communication Amount (CA) is a measurement of how much information is to be exchanged between the two stakeholders. Similar to UCC, for the purposes of this discussion, the values of CA are only meaningful when compared relative to one another. For example, we may define the CA between a customer and the project manager as 100, while the CA between the tester and the programmer is 500. That means the information exchanged between the tester and the programmer is about five times as much as that between the customer and the project manager.

The Communication Quality (CQ) is the measure of accuracy of the communication between two stakeholders. It is measured as what percentage of information has been correctly passed between the two stakeholders. The ideal situation is that all the information has been correctly transmitted from one stakeholder to another stakeholder. However, in reality, miscommunication is common. Generally, information passing will be more accurate among people with similar background. This research measures CQ as a percentage value.

Total Communication Cost and Overall Communication Quality

Based on the value of UCC, CA and CQ for all communication channels in a GSD-CN, we can calculate the Total Communication Amount (TCA), Total Communication Cost (TCC), and the Overall Communication Quality (OCQ) for a project based on the following formulae:

$$TCA = \sum (CA) \qquad (1)$$

$$TCC = \sum (CA \times UCC) \qquad (2)$$

$$OCQ = (\sum (CA \times CQ)) / (\sum CA) \qquad (3)$$

where the sum is for all the communication channels.

The two parameters TCC and OCQ could play important roles for a Global Software Project as major indicators. TCC represents the total communication cost and it could be one of the important costs for that project. OCQ represents the overall communication quality; it could be an even more important factor. As we know, the low quality communication will result different understanding of the project; that would degrade the quality of the project, increase the risk and even fail the project.

Now we use one example to illustrate how to calculate TCC and OCQ. Suppose there is a GSP with the GSD-CN as shown in Figure 1. The CA, UCC and CQ are listed in Table 1. and are based on estimates by the authors (i.e. the numbers should be used solely for illustrative purposes).

Table 1 The communication parameters for the GSD-CN

#	Stakeholder	Stakeholder	CA	UCC	CQ
1	Investor	Customer1	10	2	95%
2	Investor	Customer2	10	2	95%
3	Customer1	Customer2	30	1	99%
4	P Manager	Designer	200	1	99%
5	P Manager	Developer	100	2	98%
6	P Manager	Tester	100	2	98%
7	Designer	Developer	200	1	99%
8	Designer	Tester	100	2	98%
9	Developer	Tester	200	1	99%
10	Investor	P Manager	10	20	50%
11	Investor	Designer	5	20	50%
12	Customer1	P Manager	100	10	60%

13	Customer1	Designer	50	10	60%
14	Customer1	Tester	20	20	50%
15	Customer2	P Manager	100	10	60%
16	Customer2	Designer	50	10	60%
17	Customer2	Tester	20	20	50%

Table 1 shows that the UCC within the local area is much cheaper than that across the communication barrier; similarly the CQ is much higher. Then based on Eq. (1), Eq. (2), and Eq. (3), we can calculate that the Total Communication Amount (TCA) is 1305, the Total Communication Cost (TCC) is 5370, and the Overall Communication Quality (OCQ) is about 87%.

4 GLOBAL SOFTWARE DEVELOPMENT BROKERS

For GSDPs, we propose a new role called 'Global Software Development Broker' (GSDB) as a solution to reduce the communication cost and to improve the communication quality. In this section, firstly we give an informal definition of GSDB through describing their functions and characteristics; then we elaborate the benefits and possible pitfalls, finally we use one example to support its benefits.

Functions and Characteristics of GSDBs

There are usually two major groups of stakeholders for a software project. The first group is the software service consumer, and the second group is the software service provider. However, a GSDB does not belong to either group; it is like a gateway or bridge between the two groups across the communication barrier. Similar to a travel agent, a travel agent doesn't directly provide air-travelling service, but a traveller might find it is more convenient to deal with a travel agent than directly contact an airline company.

A GSDB may have the following functions:

1. Identify suitable partners for a GSDP. It is not an easy task for a GSDP to find suitable partners in other countries. Especially for companies who don't already have these connections. For example, how could a Chinese Web Programming company find projects from the US or Europe, or an Australian manufacturing company find an Indian IT company to implement its new ERP systems? This could be a very challenging and high-risk task. While, a GSDB, based on its service areas, should have a reasonable size of database of software service providers as well as software service consumers in different regions.
2. Provide legal service. Different countries have different commercial laws. A GSDB may help negotiate with two or more parties from different countries and help develop a mutually beneficial contract.
3. Communication service. As the most important reason for creating GSDBs is to reduce the communication cost and to improve communication quality between different stakeholders for GSDPs, the most important function of a GSDB is to help communication flow within the GSDPs.

Based on the functions of GSDBs, we can foresee that a qualified GSDB should posses:

1. a considerable size of customer database,
2. excellent communication skills as well as customer service skills for two or more countries,
3. familiarity with software engineering processes and practices, especially software requirement engineering, and familiarity with the application domain,
4. good knowledge of the commercial law systems for multiple served countries,
5. strictly professional code of ethics.

Benefits and pitfalls for using GSDBs

As it has been discussed before, there are a number of benefits for using a qualified GSDB in a GSDP.

1. Help to identify suitable project partners and help to make a legal contract.
2. Simplify the entire communication network structure. In our previous work (Bolstorff, and Rosenbaum, 2007) , we have already proved that introducing a suitable Virtual Brokerage (VB) can reduce the structural complexity of a Collaborative Network (CN). In this paper, the relationship between a GSDB to a GSD-CN is a special case of a VB to a CN. Therefore the previous research result is still valid here and we will not repeat it in this paper.
3. Reduce the Total Communication Cost (TCC). Even though introducing a new role in the system may increase the Total Communication Amount (TCA), it may reduce the TCC as it breaks some communication channels with high UCC and replaces them with communications with low UCC. An example will be presented in the next subsection.
4. Improve the Overall Communication Quality (OCQ). Similar to the previous point, OCQ of a software project is affected by some communication channels of poor Communication Quality (CQ). The new inserted node will try to replace those channels with high quality channels to increase the OCQ. This point will also be illustrated by the example in the following subsection.

Apart from the benefits of using a qualified GSDB, there are still some risks we need to be aware of:

1. Unqualified GSDB may introduce unsuitable partners for a project
2. The communication skills for different countries are essential. A GSDB located in India may not have the required communication skills when dealing with French customers. Make sure that the selected GSDB has the right skill sets.
3. The selected GSDB may not have sufficient domain knowledge for the project. For example, a GSDB with expertise in general health application domain may not have sufficient knowledge for developing a software system for a particular type of medical device.
4. Unlawful GSDB behaviour may expose customers' commercial information to their competitors.
5. Generally, the qualification and reputation of the GSDB is essential for the success of the project.

5 AN EXAMPLE

This section uses a simple example to demonstrate some benefits of adding a GSDB in a GSDP.

We use the same example as in the previous sections. However, this time we have added a GSDB who has profound communication skills both to the Chinese development team as well as to the European customers. We also assume that the GSDB has good knowledge in the application domain and fully qualified as a requirement engineer. Based on these assumptions, the new GSD-CN is shown in Figure 2, and the new communication parameters are show in

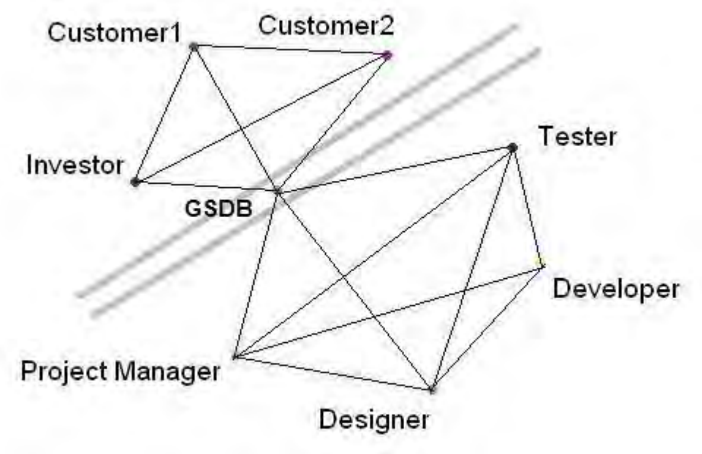

Figure 2. The GSD-CN with a GSDB

Table 2 Communication parameters for the new GSD-CN

#	Stakeholder	Stakeholder	CA	UCC	CQ
1	Investor	Customer1	10	2	95%
2	Investor	Customer2	10	2	95%
3	Customer1	Customer2	30	1	99%
4	P Manager	Designer	200	1	99%
5	P Manager	Developer	100	2	98%
6	P Manager	Tester	100	2	98%
7	Designer	Developer	200	1	99%
8	Designer	Tester	100	2	98%
9	Developer	Tester	200	1	99%
10	Investor	GSDB	15	2	98%
11	Customer1	GSDB	150	2	98%
12	Customer2	GSDB	150	2	98%
13	P Manager	GSDB	200	1	99%
14	Designer	GSDB	100	1	99%
15	Tester	GSDB	40	1	99%

Based on the definitions in the previous section, it is not difficult to calculate that the Total Communication Amount (TCA) is 1605, the Total Communication Cost (TCC) is 2240, and the Overall Communication Quality is about 98%. Compare the results with that in the previous section, even though the TCA is increased about 23%, but the total communication cost is reduced to about half (which could offset the cost for hiring the

GSDB), and most importantly, the OCQ is increased from 87% to 98%, which can significantly benefit the quality of the project outcome as well as reduce risk.

6 CONCLUSIONS AND FUTURE WORKS

For GSD projects, the communication barriers between different stakeholder groups significantly increase the communication cost and affect the communication quality. To solve this problem, the paper proposes a new role called 'Global Software Development Broker' (GSDB), which serves as a gateway for the different stakeholder groups to cross the communication barriers.

To prove the feasibility of this method, this paper introduced a new type of Collaborative Network (CN) called GSD-CN as a formal research framework. Based on this framework, by using a simple example, this paper proves that by introducing a GSDB, a GSD project can (a) simplify the collaborative network structure, (b) decrease the total communication cost, and (c) improve overall communication quality. That indicates a possibility for the improvement of the quality of the GSD project. The example shows that the proposed method could be a promising approach in the future. Also the dynamics and mechanisms of how GSD-CNs should create the requisite variety in order to manage the complexity of the environment will be further studied via Co-evolution Path Model in future work (Kandjani, Bernus and Nielsen, 2012; 2013).

We acknowledge that this research is still at a preliminary stage in the sense that a number of issues need to be explored further before attempting a practical application. We identified the following possible future research directions for this work.

- Design a formal and practical method to evaluate the unit communication cost, communication quality and communication amount.
- Establish a model to reveal the relationship between the communication quality and the quality of the corresponding software system.
- Collect data from real large case studies.
- Formally define the qualification of a GSDB to make this new role realistic in the industry.

REFERENCES

Ashby W.R. 1956, *An introduction to cybernetics*. Taylor & Francis.

Bernus P. and Noran O. 2010. A Metamodel for Enterprise Architecture. In Enterprise Architecture, Integration and Interoperability. *IFIP Advances in ICT*. Vol 326 Berlin : Springer.pp56-65.

Boden A. and Avram G. (2009) 'Bridging knowledge distribution: the role of knowledge brokers in distributed software development teams', Cooperative and Human Aspects on Software Engineering Conference, May, Vancouver, pp8-11

Bolstorff P. and Rosenbaum R. G. 2007. *Supply chain excellence: a handbook for dramatic improvement using the SCOR model.* New York : Amacom Books.

Bosua R. and Scheepers R. 2007, 'Towards a model to explain knowledge sharing in complex organizational environments', Knowledge Management Research & Practice, vol5, pp93-109

Camarinha-Matos L. and Afsarmanesh H. 2004. (Eds) *Collaborative networked oganizations: a research agenda for emerging business models.* London : Kluwer.

Cataldo M. and Herbsleb J. D. 2008. "Communication networks in geographically distributed software development." *In Proc2008 ACM Conf on Computer Supported Cooperative Work. ACM.* pp. 5792-588.

Cataldo M., Bass M., Herbsleb J. D. and L. Bass. 2007. "On coordination mechanisms in global software development, " *In Proc2nd IEEE Int Confon Global Software Engineering*.Washington : IEEE. pp 71-80.

Cataldo M., Bass M., Herbsleb J. D., and Bass L. 2006. "Managing Complexity in Collaborative Software Development: On the Limits of Modularity," Supporting the Social Side of Large Scale Software Development Worksh. *In Proc CSCW06. ACM.* p. 15.

Cover T.M. and Thomas J.A. 2006. *Elements of information theory*. New York : Wiley.

IFIP-IFAC-Task-Force. 1999. GERAM: Generalised enterprise reference architecture and methodology. V6.3 1999. (also published in Handbook on Enterprise Architecture. Berlin : Springer. pp22-65.

ISO15704. 2000, Amd.2005. Industrial automation systems – Requirements for enterprise-reference architectures and methodologies. Geneva : ISO TC184.SC5.WG1.

Kandjani H. and Bernus P. 2011a. Capability Maturity Model for Collaborative Networks based on Extended Axiomatic Design Theory. In L.M.Camarinha-Matos, A.A.Pereira Klen, H.Afsarmanesh (Eds.) Adaptation and Value Creating Collaborative NetworksProc. *IFIP Advances in ICT* Vol 362 Berlin : Springer. pp421-427.

Kandjani H. and Bernus P. 2011b. Engineering Self-Designing Enterprises as Complex Systems Using Extended Axiomatic Design Theory. *IFAC Papers On Line* V18 (Part1) Amsterdam : Elsevier, pp11943-11948.

Kandjani, H., Bernus, P., and Nielsen, S. 2012. Co-evolution path model: how enterprises as complex systems survive on the edge of chaos. *In ACIS 2012: Location, location, location: Proc23rd Australasian Conference on Information Systems*, pp. 1-11.

Kandjani, H., Bernus, P., and Nielsen, S. 2013. Enterprise Architecture Cybernetics and the Edge of Chaos: Sustaining Enterprises as Complex Systems in Complex Business Environments. *In HICSS 2013: Proc 46th Hawaii International Conference on System Sciences*, pp. 3858-3867.Kandjani H., Bernus P. and Wen L. 2012a. Global software development: measuring, approximating and reducing the complexity of global software development using extended axiomatic design theory. *In ACIS 2012: Location, location, location: Proc23rd Australasian Conference on Information Systems*, pp. 1-11.

Kandjani H., Bernus P., and Wen L. 2012b. Enterprise Architecture Cybernetics for Complex Global Software Development: Reducing the Complexity of Global Software Development Using Extended Axiomatic Design Theory. *In 7thIEEE Int Conf. on Global Software Engineering (ICGSE)*, Washington : IEEE, pp. 169-173.

Kandjani H., Wen L. and Bernus P. 2012. Enterprise Architecture Cybernetics for Collaborative Networks: Reducing the Structural Complexity and Transaction Cost via Virtual Brokerage, *IFAC Papers Online*, Vol 14, Part 1, pp 1233-1239.

Kolmogorov A. N. 1965. On the Logical Foundations of Information Theory and Probability Theory. *Problems of Information Transmission*. 1(1), pp1-7.

Li M. and Vitányi P. 2008, *An introduction to Kolmogorov complexity and its applications*. (3rd Ed). Berlin : Springer.

Marczak S. and Damian D. 2011. "How interaction between roles shapes the communication structure in requirements-driven collaboration,", *In Proc 19thInt Requirements Engineering Conf.* Washington : IEEE. pp. 47-56.

Markus LM, 2001, 'Toward a theory of knowledge reuse: types of knowledge reuse situations and factors in reuse success', Journal of Management Information Systems 18(1), pp 57-93

Milanovic I. , Stefanovic M. et al. 2011. Supply Chain Information Integration Methods from the Quality Aspect. *In Proc. 5thInt. Quality Conf.* University of Kragujevac : Faculty of Mechanical Eng. pp439-448.

Sangwan R., Mullick N., and Bass M. 2006. *Global software development handbook*. CRC Press.

Scherrer-Rathje M., Arnoscht J. *et al*. 2009. A generic model to handle complexity in collaborative networks. *In Proc. PICMET 2009*. Washington : IEEE., pp271-287.

Šmite D. and Borzovs J. 2008. "Managing Uncertainty in Globally Distributed Software Development Projects," University of Latvia, *Computer Science and Information Technologies*, vol. 733, pp. 9-23.

Suh N. 2001. *Axiomatic Design: Advances and Applications*. New York : Oxford Uni. Press.

Suh N. P. 2005. *Complexity: Theory and Applications*. New York : Oxford University Press.

COPYRIGHT