Supporting Communication and Cooperation in Global Software Development with Agile Service Networks

Damian Andrew Tamburri and Patricia Lago

VU University Amsterdam
The Netherlands
{d.a.tamburri,p.lago}@vu.nl

Abstract. Current IT markets exhibit many constraints (e.g. budget, staff shortage, etc.). These constraints force IT companies to increase productivity using globally distributed manpower. Literature shows that global software development (GSD) indeed raises productivity but reduces communication and collaboration between teams. Consequently, the risk of failure increases. To ease communication and collaboration among teams, novel engineering methods must be provided. To address this problem, we propose using Agile Service Networks (ASNs). ASNs are an emergent paradigm in which service oriented applications (network nodes) collaborate through agile and dynamic service interactions (network edges). Agile interaction among ASN nodes, allow mitigating distance (typical of GSD) by dynamically adapting communication and collaboration as needed. Through ASNs, GSD can be seen as a global network of resources (teams, documentation, knowledge, etc.) among which agile interactions allow flexible knowledge exchange and team collaboration. To establish feasibility of our proposal, we investigated how ASNs can support GSD. Based on existing works in the fields of both ASNs and GSD, we mapped GSD challenges on ASNs key features and devised a meta-model showing how ASNs are used to support GSD requirements.

1 Introduction

Our global economy is constantly challenged by time-to-market and budget issues. Moreover, the availability and cost of manpower rapidly change. To maximize productivity in these conditions, IT companies carry out software development globally. Ideally, by using teams in different sites and timezones, all 24 hours in a working day can be rendered productive. Unfortunately, when doing so, the issues in knowledge exchange and synchronization among teams are often underestimated. These problems regard people rather than technology, and hence they are very difficult to study. In addition, costs inevitably raise because of increased travel needs (e.g. for management and architects' meetings etc.). Consequently, workforce becomes ineffective, costs prohibitive and ultimately, projects fail [6,13]. The problem we want to address is the lack of practices and tools to support these issues in GSD.

I. Crnkovic, V. Gruhn, and M. Book (Eds.): ECSA 2011, LNCS 6903, pp. 236–243, 2011.

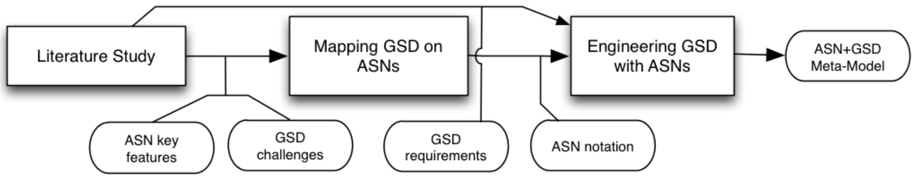

Fig. 1. Research Approach

ASNs are networks of service-oriented applications (network nodes) created by collaborative service interactions (network edges) among many cooperating industrial parties. Through ASNs, complex yet agile and adaptable business transactions take place on a global scale.

Similarly to GSD processes, ASNs stem from collaborative business processes [4,2], distributed on a global scale. Since GSD is indeed a business process (complying with the definition in [10]) ASNs can be used to model the business process of developing software globally. Their networked and agile nature can be enriched to support both social and technical requirements of GSD. In this paper we investigate how can ASNs support GSD processes. ASNs were only recently introduced, and using them to support GSD was never researched so far. Therefore our investigation faces challenges such as limited literature on ASNs and no related work. Another interesting challenge regards the social aspects of GSD: these must be represented and supported through ASNs, which are defined as a technical system. Two main contributions are offered: *(i)* a mapping of GSD challenges on ASN key features, showing that ASNs can support GSD; *(ii)* a meta-model that shows ASNs supporting GSD requirements. Figure 1 shows our research steps (rectangles) as well as inputs and outputs (rounded rectangles). First we carried out a literature study obtaining GSD challenges, GSD requirements and ASNs' key features. Then we showed feasibility of our proposal by mapping GSD challenges on ASNs' key features. Finally, we devised a meta-model to show how GSD can be supported by ASNs. This meta-model was obtained extending an existing ASN notation [15,1] to support GSD requirements.

2 Literature Study

This section surveys Agile Service Networks and Global Software Development. To gather clear-cut literature for ASNs, we applied the topic search string (i.e. "Agile Service Networks") to major scholarly search engines (Google Scholar, IEEExplore, ACM Digital Library, Wiley Interscience, Microsoft Academic Research). For GSD, we consulted experts in the field. The resulting publications were [4,12,14,2] for ASNs and [7,3,6,8,13,5] for GSD. To these publications we added [15], a publication from the S-Cube consortium (available at www.s-cube-network.eu) discussing Service Networks.

2.1 Agile Service Networks

Analyzing the selected papers we have identified the following key features exposed by ASNs.

ASNs are dynamic: All the papers describe ASNs as being highly dynamic entities. In [4,14,15] dynamism is seen as essential part of service interactions in collaborative industrial networks (i.e. industrial value networks [11]). Dynamic agility in this context is regarded as the immediate ability to adapt to dynamic changes in demand and offer.

ASNs are business-oriented: All papers promote the concept of ASNs from a business perspective. ASNs emerge from business corporative collaborations [4] and represent complex service applications interacting in a networked business scenario involving multiple corporations or partners in different sites (i.e. different geolocations) [2]. Within ASNs, business value can be computed, analyzed and maximized [4,12].

ASNs are collaborative: In all papers, ASNs are defined as interoperating business alliances. Each member cooperates with others to achieve a common goal (e.g. service level, value increase). Therefore, ASNs are collaborative.

ASNs are emergent: There are no engineering and design methods specific to ASNs. They form spontaneously as a consequence of business alliances teaming-up to collaboratively increase business value through corporative partnership [4,14,15,2].

In addition to these key features, we used the ASN notation in Figure 2, taken from [15]. The main architectural elements for ASNs in the notation are **Participant**s (ASN nodes) and **Relation**s (ASN edges). For the sake of space, we do not further discuss this notation and urge the reader to refer to [15] and [1] for further details.

2.2 Global Software Development

Analyzing the suggested papers we have identified the following challenges in GSD.

Social Aspects are important to enable teams to integrate and exchange knowledge correctly [5,7]. In [5] GSDs are comprised of globally distributed teams

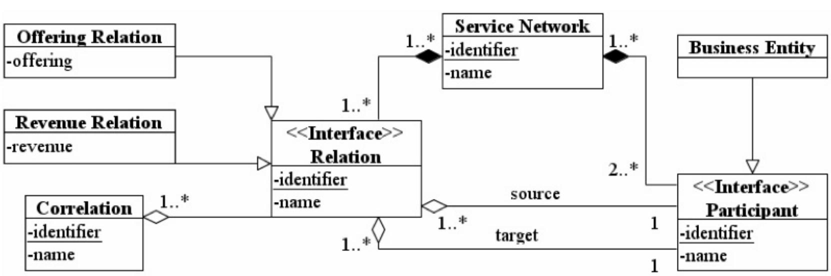

Fig. 2. Service Networks Notation from [15] and [1]

carrying out an objective collaboratively. Collaboration is increased by socialization in teams and social networking [7].

Collaboration increases productivity by raising team interaction, awareness and responsibility on the project [7]. In [6] the key issue for GSD is coordination in dynamic contexts. Collaborative effort, is required for GSD to succeed.

Flexibility in management, to coordinate multisite development [8,6]. Ideally GSD should be able to use all available resources regardless of geographical location and coordinate these collaboratively. Management should be flexible enough to provide fine grained control over all types of resources (e.g. documentation, people down to individual skills). Knowledge localization is challenging since granularity of management and control over resources and people is limited [8].

Reduced dependency among teams, so that productivity of one team is not impacted by productivity of others. Distance can be compensated with tactics to increase communication, loosen teams dependency and limit participants' cultural difference [3].

Coordination of all resources available, i.e. manpower, tools, document artifacts, knowledge, to timely allocate resources and maximize productivity [3,6]. GSD often fails because many of the mechanisms that coordinate work in co-located projects (e.g. stand up or colloquial meetings, informal "water-cooler" talk etc.) are absent or disrupted.

Geolocalization to allow project awareness among teams. Since teams are geographically dispersed and often unknown to each other, they need intercommunication and awareness infrastructures to actively participate on the project [13,9,7].

Finally, from these papers we elicited requirements for GSD processes (for the sake of space the list is not present here and is availableonline[1]). We obtained these requirements by: *(a)* scanning through the literature, coding text describing requirements or needs for GSD processes; *(b)* analyzing industrial case studies from [13]. In total, we obtained 17 Requirements from literature coding, and 12 requirements from the real-life industrial GSD scenarios in [13].

The entities and relationships occurring in GSD processes (according to requirements) can be summarized in the following scenario:

"Company X develops software globally by using N globally distributed ***teams****. Each team is made of* ***engineers*** *with individual* ***skills, social background, roles****, etc. A* ***global team map*** *is used to track* ***location, timezone*** *and* ***knowledge*** *of every team (e.g. skills, documentation available, progress made on artifacts, etc). One or more teams are* ***core*** *teams since their task is managing the whole process, checking* ***shared documentation****, deciding a project-wide* ***technical space*** *and planning* ***travel budget****. Travel budget is needed for the frequent "****awareness****" meetings among teams.* ***Shared documentation*** *is needed to document the project and also to increase awareness of every member. A common* ***technical space*** *is needed to ease communicability (e.g. common formats) and knowledge exchange (e.g. common platforms). As soon as* ***requirements*** *are agreed with the* ***stakeholders*** *they are used to generate a* ***global***

[1] `http://www.picfront.org/d/87GP`

***architecture**. Once the global architecture is defined, it is split into **project units**. Project units are allocated to **engineering** teams, responsible for their development. **Service** teams update shared documentation to allow consistency and further increase **project awareness**".*

Words in bold in the scenario represent the entities taking part in GSD processes. This scenario and the GSD requirements it represents, are used in the definition of our meta-model in section 4.

3 Mapping GSD to ASNs

This section shows that ASNs can be used to support GSD processes. To this aim, GSD challenges were matched with ASNs key features (both presented in Section 2). Table 1 summarizes results. Column 1 represents GSD challenges,

Table 1. Mapping of ASN characteristics on Global Software Development

GSD needs...	*... ASNs are...*	*Rationale*
social aspects	*business-oriented*	ASNs stem from the business strategies for collaborative value increase. These strategies are modeled around social demands and user profiles (social context, background, social extraction, etc.). This means that ASN nodes are modeled to satisfy customers' (social) characteristics[4,15].
collaboration and awareness		Agile Service Networks are generated through collaboration of networked service applications[4]. Formally, collaboration terms are stated in service level agreements [2,14]. This means that every ASN node must collaborate with other to achieve the network's goal (similarly to GSD Collaboration needs). In so doing, formal service level agreements must be in place so that collaborating nodes know what are the terms of the collaboration (similarly to GSD awareness needs).
collaborative coordination	*collaborative*	ASNs are collaborative and adaptable to context change. Service applications coordinate spontaneously to achieve results in accordance to fixed service level needs[2,14]. Dynamic adaptation of both nodes and interactions allows dynamic coordination.
reduced dependency among teams		ASNs provide clear-cut definitions of network nodes (i.e. service applications)[14]. Agile interactions between nodes enable loose dependency: if one (service) node is not available, another node can be called up [2].
management flexibility	*dynamic*	Agile Service Networks provide a dynamic infrastructure, adaptable to context change. Agile interactions among nodes allow for flexible management of the network.[2].
geo-localization of resources	*emergent*	agile service networks are emergent through service discovery, localization and management of serving nodes [4,15,14]

while column 2 shows ASNs' matching key feature. Column 3 provides rationale. The table shows that all GSD challenges found can be supported by ASNs key features.

4 Engineering GSD with ASNs

To show how ASNs can support GSD, requirements for GSD processes must be satisfied through ASNs. In this section we show a meta-model in which entities

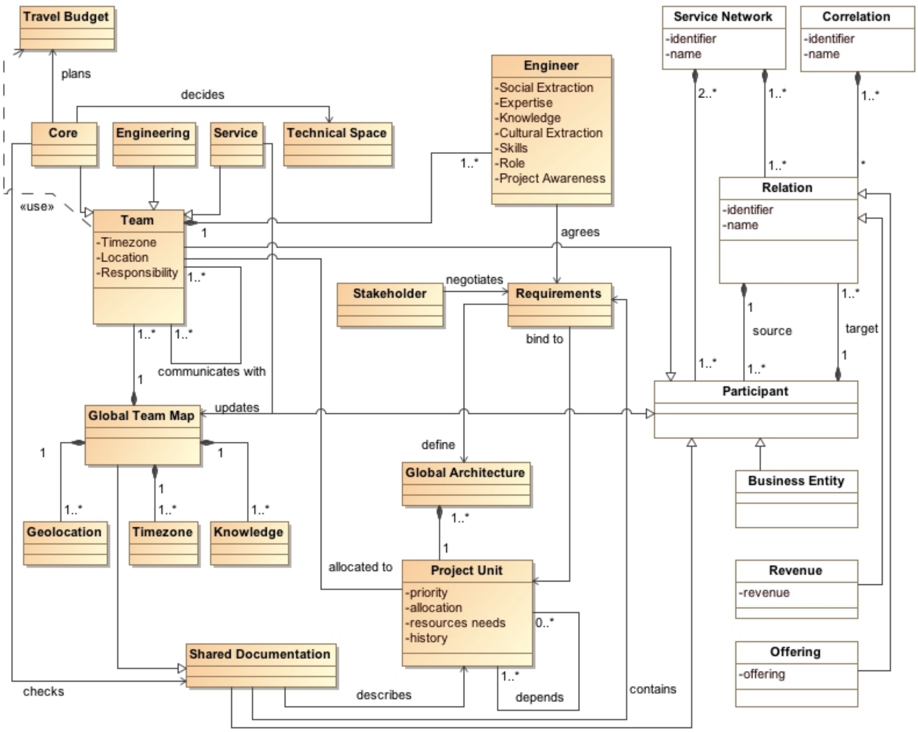

Fig. 3. ASN notation for GSD

and relations from GSD requirements (as summarized in the scenario closing section 2.2) are modeled through an ASNs notation (see Figure 2). To build this meta-model, we first reproduced entities and relationships stemming from GSD requirements. Then we reproduced the ASN notation in the meta-model. Finally, we extended the ASN notation by specialization (i.e. by drawing a generalization from GSD specific concepts to ASN generic concepts). More formally, the following "merging" rule was applied:

"*specialize the* **Participant** *class from Figure 2 with all entities that take active part in GSD according to requirements (i.e. that are participants in an ASN). Specialize the* **Relation** *class with all relations among resulting* **Participant***s*".

This rule is both necessary, and sufficient. It is necessary since all the active contributors in GSD must be **Participant**s in an ASN; it is sufficient, since all remaining elements to be merged (i.e. relations between **Participant**s) are **Relation**s in the ASN.

Therefore, the concept model in Figure 3 was obtained by drawing the entities and relationships required for GSD (i.e. stemming from the requirements we elicited) and then applying the rule defined above.

On the left hand side, The model shows the entities and relationships stemming from the requirements (filled), while the ASN notation (originally in Figure 2) is on the right hand side (non filled). The two are merged by specializing the

Participant class on the right, with **Team**s, **Global Team Map** and **Shared Documentation** classes on the left. Since these three entities carry out (either directly or indirectly) the software development, indeed they are the active participants in GSD, according to requirements. Relations taking place among these elements are ASN transactions (i.e. **Relation**s). For the sake of clarity in Figure 3 we do not show the relations on the GSD side (left, filled) specializing the **Relation** class on the ASN side (right, non-filled).

Indeed this meta-model shows that an ASN to support GSD processes can be created by modeling **Team***s,* **Global Team Map***s and* **Shared Documentation** *as active* **Participant***s within the ASN. Consequently, the relations between these are ASN collaborative transactions (i.e.* **Relation***s).*

5 Conclusions and Future Work

In this paper we wanted to establish if and how ASNs supported GSD. To this aim we systematically searched for literature in ASNs and GSD. From the gathered literature we obtained ASN key features, GSD challenges and GSD requirements. Mapping GSD challenges on ASNs' key features led us to conclude that ASNs indeed support GSD. Moreover, extending an ASN notation to meet GSD requirements, we have shown how this support can be concretized.
This notwithstanding, it can be noticed that ASNs are still missing some important architectural elements, e.g. social aspects of GSD. These aspects are key to provide added-value support tools. Since GSD actors are teams part of organizational structures (i.e. corporations, software companies etc.), a systematic literature review into Organizational Social Structures is being carried out. From this study we hope to develop a socio-organizational context model to enrich ASNs. Moreover, since Figure 3 is a meta-model, i.e. a model for a model, further exploration of model-driven engineering methods for GSD through our ASN_{GSD} meta-model is in order. Moreover, validation of this meta-model should be put in place to make its support to GSD meaningful. For this, industrial case studies should be developed and results should be analyzed against industrial expectations (e.g. a focus group). Further on, more experimentation should be invested in simplifying / improving the model in Figure 3 (e.g. action research)[2].

References

1. Bitsaki, M., Danylevych, O., Heuvel, W.-J., Koutras, G., Leymann, F., Mancioppi, M., Nikolaou, C., Papazoglou, M.: An architecture for managing the lifecycle of business goals for partners in a service network. In: Mähönen, P., Pohl, K., Priol, T. (eds.) ServiceWave 2008. LNCS, vol. 5377, pp. 196–207. Springer, Heidelberg (2008)

[2] The authors acknowledge the support of the European Community's Seventh Programme FP7/2007- 2013, grant agreement 215483 (S-Cube), for partially funding this project.

2. Bucchiarone, A., Cappiello, C., Di Nitto, E., Kazhamiakin, R., Mazza, V., Pistore, M.: Design for adaptation of service-based applications: main issues and requirements. In: Proceedings of the 2009 International Conference on Service-Oriented Computing, ICSOC/ServiceWave 2009, pp. 467–476. Springer, Heidelberg (2009)
3. Carmel, E., Agarwal, R.: Tactical approaches for alleviating distance in global software development. IEEE Software 2(18), 22–29 (2001)
4. Carroll, N., Whelan, E., Richardson, I.: Applying social network analysis to discover service innovation within agile service networks. Service Science 2, 225–244 (2010)
5. Ebert, C., De Neve, P.: Surviving global software development. IEEE Software 18(2), 62–69 (2001)
6. Herbsleb, J.D.: Global software engineering: The future of socio-technical coordination. In: Briand, L.C., Wolf, A.L. (eds.) FOSE, pp. 188–198 (2007)
7. Herbsleb, J.D., Mockus, A., Finholt, T.A., Grinter, R.E.: An empirical study of global software development: distance and speed. In: ICSE 2001: Proceedings of the 23rd International Conference on Software Engineering, pp. 81–90. IEEE Computer Society, Washington, DC, USA (2001)
8. Herbsleb, J.D., Moitra, D.: Guest editors' introduction: Global software development. IEEE Software 18, 16–20 (2001)
9. Herbsleb, J.D., Moitra, D.: Global software development. IEEE Software 18(2), 16–20 (2001)
10. `http://www.opengroup.org/projects/soabook/` Soa source book
11. Jetter, M., Satzger, G., Neus, A.: Technological innovation and its impact on business model, organization and corporate culture - ibm's transformation into a globally integrated, service-oriented enterprise. Business & Information Systems Engineering 1(1), 37–45 (2009)
12. Metzger, A., Pohl, K.: Towards the next generation of service-based systems: The S-cube research framework. In: van Eck, P., Gordijn, J., Wieringa, R. (eds.) CAiSE 2009. LNCS, vol. 5565, pp. 11–16. Springer, Heidelberg (2009)
13. Sangwan, R., Bass, M., Mullick, N., Paulish, D.J., Kazmeier, J.: Global Software Development Handbook. Auerbach Series on Applied Software Engineering Series. Auerbach Publications, Boston (2006)
14. van den Heuvel, W.-J., Zimmermann, O., Leymann, F., Lago, P., Schieferdecker, I., Zdun, U., Avgeriou, P.: Software service engineering: Tenets and challenges. In: Proceedings of the 2009 ICSE Workshop on Principles of Engineering Service Oriented Systems, PESOS 2009, pp. 26–33. IEEE Computer Society, Washington, DC, USA (2009)
15. Andrikopoulos, V., Benbernou, S., Bitsaki, M., Danylevych, O., Hacid, M.S., van den Heuvel, W.J., Karastoyanova, D., Kratz, B., Leymann, F., Mancioppi, M., Mokhtari, K., Nikolaou, C.N., Papazoglou, M.P., Wetzstein, B.: Survey on business process management. Deliverable 1. S-Cube Consortium (July 14, 2008)