A Method for Work Distribution in Global Software Development

Dilani Wickramaarachchi
Department of Computer Science and Computer Engineering,
La Trobe University
Melbourne, Australia
dwickramaarachchi@students.latrobe.edu.au

Richard Lai
Department of Computer Science and Computer Engineering,
La Trobe University
Melbourne, Australia
r.lai@latrobe.edu.au

***Abstract*—Global Software Development (GSD) has recently evolved and has been embraced by the competitive software industry today. The major attraction of GSD is due to the greater availability of human resources in distributed zones at a low cost and advances in communication technology. However, recent research reveals that expected benefits of GSD are not always realized as predicted since additional costs are often involved for the communication and coordination activities between the dispersed groups. Therefore, the main challenge of GSD today is to minimize the additional costs and maximize the benefits. A proper work distribution mechanism is particularly important to reduce the additional challenges facing GSD. In this paper, we present a method for work distribution to multiple locations. It starts with the identification of work as stages/phases in the Software Development Life Cycle (SDLC) and grouping them according to the Software Process Model (SPM) used. A final suggestion for the work distribution is based on multiple criteria such as work dependency, site dependency, site specific and work specific characteristics. The priority given to the criteria depends on the project objective.**

Keywords— Global Software Development, Work Distribution, Task Allocation, Software Process Model, Software Development Life Cycle

I. INTRODUCTION

Global Software Development (GSD) can be defined as "any aspect of software engineering that involves the combined efforts of software professionals in different locations separated by significant distances" [1]. A tight budget and a shortage of resources and time have motivated many enterprises to start looking for outside partners. With the assistance of the internet, it is now possible to share information with anyone around the world, which has opened the way to locate digital services in distributed locations. Currently, offshoring IT services is becoming a necessary step for companies in order to survive in a competitive software market.

In spite of the growing attention to GSD, there is a limited understanding of how to make GSD a success and why it fails [2]. Several studies illustrate that half of the organizations which shift processes offshore fail to generate expected financial benefits [3] and some of them fail completely [4]. GSD projects and case studies range from announcements of remarkable victory to total collapse [5]. The cost-benefits trade-off in GSD is still not fully understood [6]. No research so far has imparted a clear vision of the true amount of investment necessary to make GSD projects a success [5]. Despite these threats, offshoring is becoming a norm in the software industry [7]. Therefore, the challenge is to minimize the threats in order to increase the likelihood of high profits.

The distribution of tasks/work to sites/locations has a direct impact for the effort overhead in GSD. A large number of possible combinations for work distribution/task allocation makes the process more complicated. For example, if a software development project consists of five different tasks which need to be distributed among four development sites, theoretically there are 4^5 =1024 different combinations available for work distribution. It is evident that overseeing and evaluating all possibilities is impossible and it needs an approach to distribute the work to the available sites.

In this paper, we present a method for work distribution to different locations while minimising additional overheads. The contribution of this paper is twofold. First, it suggests the categorization of work for offshoring based on a software process model. Second, it proposes a method to distribute suitable work to suitable sites considering multiple criteria. Ultimately, the work distribution mechanism will help to realise more benefits than challenges which can lead to the success of GSD. It should be noted that we have used some words interchangeably such as work distribution/task allocation and site/location in this paper.

II. RELATED WORK

Over the last decade, a number of research initiatives have been introduced in GSD. However, there has not been proper systematic review on work distribution in GSD. We have reviewed the related work in this domain and present a summary in the following section.

It was found that some empirical studies in GSD aim at identifying the practices of work distribution and trying to classify criteria [8]. Individual expertise and availability have been identified as the most popular criteria for work distribution [9]. After conducting an analysis of existing work distribution approaches for different domains, Bokhari's algorithm has been identified as a promising model [10]. However, there is no information as to whether the use of Bokhari's algorithm would address the issues in GSD work distribution. A qualitative study aiming to identify and understand the criteria that are used in practice was conducted by Lamersdorf, et al. [11]. The results show that the sourcing strategy and the type of software to be developed have a significant effect on the applied criteria. However, these studies only analysed current practices of work distribution and there is no evidence for decision support. Nevertheless, a survey on existing work practices would support the identification of important criteria for work distribution.

A planning tool to identify task assignment based on multiple criteria and weighted project goals was developed by Lamersdorf and Munch [12]. The model is called TAMRI (Task Allocation based on Multiple cRIteria) and its implementation is based on Bayesian networks. The application of the tool requires a large amount of knowledge on casual relationships in distributed development. It is doubtful whether companies possess the maturity and data required to gather this knowledge. Lamersdorf developed a different approach in 2009 [13], with further details on the model published in 2010 [14]. In contrast to the previous approach, the focus here was based on a multiple criteria model for task allocation using Bokhari's algorithm and Bayesian networks. However, dependency on expert estimation and the large amount of empirical data required restricts the suitability of the model for recently established companies. The decision support system recently presented by S. Beecham, et al. helps project managers navigate through the alternative recommendations by the Global Teaming Model (GTM) [15]. This is mainly for project management and not for task allocation.

Moreover, the literature shows that the unavailability of proper methods has resulted in the use of ad-hoc methods which rely only on single criteria such as development cost which has proved to be very risky and often does not lead to a successful solution in the long run [10, 11, 14, 16]. Risks and other relevant factors such as work force capabilities, innovation potentials, differences between locations (physical, cultural, language, time), task dependencies and characteristics of the task are often not recognized sufficiently. However, the importance of site compatibility for better completion has been highlighted in the literature [17].

The above survey shows that none of the models in the literature provide an approach for a work distribution at the early stage of the software life cycle. Therefore, our main focus is to suggest an approach for decision support in work

III. THE METHOD

A. *The Focus*

Work distribution in GSD should be based on different criteria. There are different configurations of the GSD model in the literature, such as phase based, module based and follow-the-sun [18]. The follow-the-sun concept aims to reduce the total time of the project while working 24 hours across the time zone. According to this method, one team will work when the other team sleeps and vice versa. However, the realization of the above objective is minimal since real world project tasks are often interrelated and require communication between the teams in distributed locations. Further, lower communication quality may lead to miscommunication which can result in work needing to be performed again [19]. Therefore, interdependent work performed in distributed mode takes more time than in collocated mode. This has been confirmed by the research findings [20], which show that in distributed development, the work completion time appears to take nearly two and a half times longer than for collocated development. Therefore, the suitability of the follow-the-sun concept is limited to projects with less complicated and less dependent tasks which require fewer interactions. Only projects of this nature can be completed in a minimal time using the follow-the-sun concept. As most real world projects do not fit into above configuration, we have excluded it and focused on other configuration methods such as phase-based and module-based configurations. In this paper, we present a method for work distribution using phase-based configuration. Module-based work distribution will require a method for modularization and it will be presented in a separate paper.

In relation to work distribution design issues in GSD, there are two major problems which need to be addressed. They are:

1. How to divide the work?
2. How to distribute the divided work to different locations?

We have provided answers to the above questions through our approach.

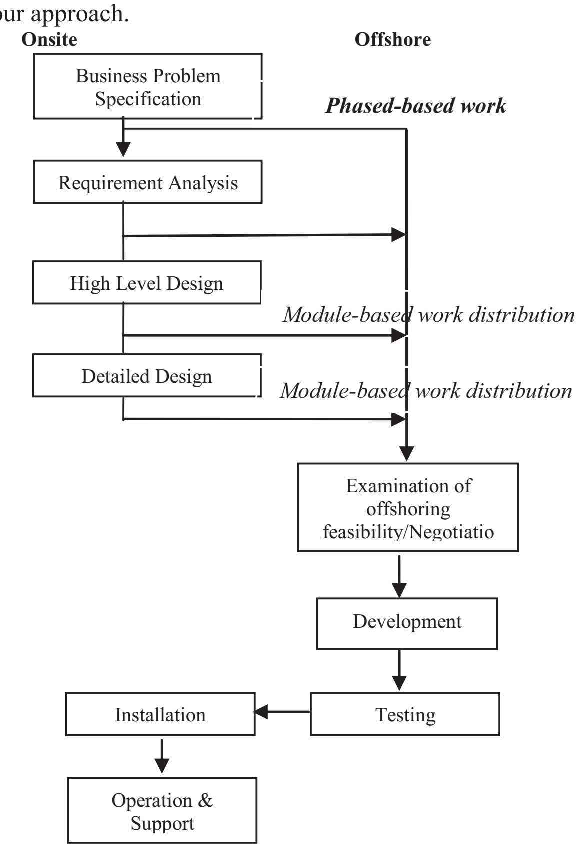


Fig 1: Work Distribution in Offshoring Life Cycle
A generalised offshoring life cycle model [21]

Why phase-based configuration?

Phase-based configuration can be used to allocate work at the early stage of the software development life cycle. Large projects are more frequently distributing work among locations/teams by phase, rather than along the lines of specific functionality [22]. Figure 1 illustrates suitable stages where different configuration methods can be adopted to the offshoring life cycle. Our main focus in this paper is to suggest a work distribution mechanism for phase-based configuration. It can be noted that the same distribution method can be used for module-based configuration after using a proper method for module formulation.

B. The Approach

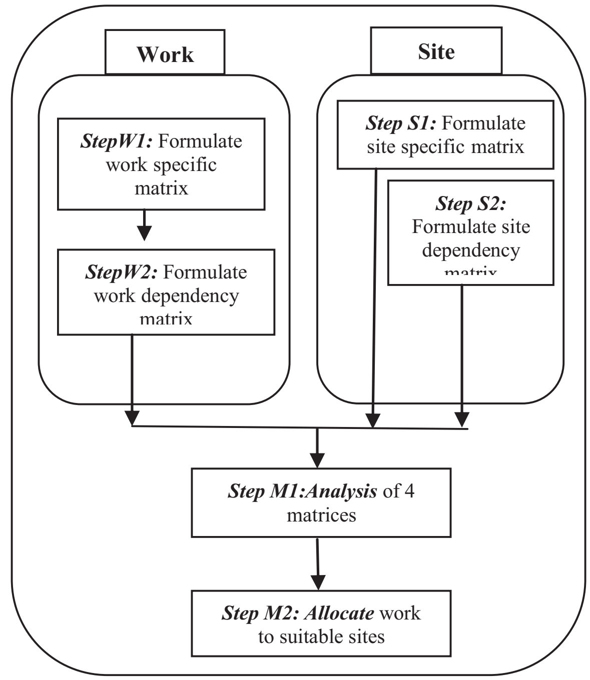
The main challenge in work distribution is to find the most suitable site for the given work. The suitability of the location will be decided considering both work-related factors and location-related factors. Major factors which are important to GSD compared to collocated software development are ***work dependency*** and ***site dependency.***

By using work dependent characteristics and site dependent characteristics, more dependent work can be allocated to sites which have more common characteristics. In this way, work distribution can result in less communication overheads and the total effort is minimised. Our method mainly focuses on the additional challenges introduced in GSD. Therefore, the distribution model that has been formulated incorporates these factors. In addition, other factors which are common in collocated software development have been considered in order to devise a realistic method. Considering all the above, we have introduced a new method for work allocation incorporating *work dependent, site dependent, work specific and site specific characteristics.* Further, the priority that is given to the above factors will vary according to the main objective of the project. The work distribution process to multiple sites consists of six steps.

1. Work specific characteristics identification
2. Work dependency identification
3. Site specific characteristics identification
4. Site dependency identification
5. Analysis of four matrices
6. Allocate work to suitable sites

The proposed framework, illustrated in Figure 2, shows the main classification with sub classifications.

The framework starts with two independent stages: architecture/work analysis and team/site analysis. Once these analyses have been completed, the investigation of the created matrices and mapping tasks to sites will be performed. It is important to note that this framework is designed to suggest an approach as an aid for the project manager. The project manager is allowed to modify any recommendation at any phase according to the availability of information, resources and project objectives.



Architecture/Work Analysis

***Step W1**: Formulate work-specific matrix*

According to the phase configuration approach, work can be defined as different phases/stages in SDLC. Work-specific characteristics are identified and a work matrix is formulated. The factors related to each phase are: rough estimated time, order of execution, specific skill required and complexity of the work. Complexity of the work is presented as a qualitative attribute (high, medium & low). The main phases of software development life cycles have been considered such as Requirement Analysis (RA), System Design (SD), Development (D) and Testing (T). A sample work-specific matrix is presented in Figure 3.

	RA	SD	D	T
Rough Estimated time	2months	3months	8months	3months
Order of execution	1	1,2	2,3	3,4
Specific skill				
Complexity	M	H	M	M

Fig 3: Work Specific Matrix

Step W2: Formulate work dependency matrix

The relationships between different phases are decided, based on the Software Process Model used by the company. Software Process Models help to arrange development activities into a specific order [21]. A large number of software process models have been proposed since the beginning of software engineering, categorized in many ways and described by attributes such as: linear vs. iterative development, sequential vs. incremental, plan-driven vs. agile development, model driven vs. evolutionary development [21].

Software development projects that lend themselves easily to offshoring are projects that automate well documented business functions or processes where little day-to-day interaction is required [21]. Unfortunately, most of the projects are not of that nature, instead they require a lot of interaction. Therefore, project teams have to be set up in such a way that onsite and offshore personnel can communicate easily. In addition, the customer's or end user's requirements need to be communicated effectively [21]. According to the software process model, interaction requirements will vary. For example, we have considered two scenarios where a sequential model and an iterative model are used in relation to work which needs to be distributed to four different locations. Suitable grouping in different phases can be represented as below to have a better functionality among distributed locations.

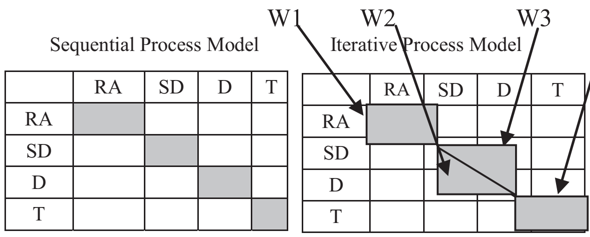


Fig 4: Grouping for work distribution

It is necessary to perform suitable grouping/ categorization which considers the software process model, the nature of the software project and the characteristics of each phase to minimise coordination and communication complexity. The number of groupings will be decided based on the number of distributed locations. In Figure 4, the matrix on the left represents the grouping for the sequential process model and the matrix on the right represents the grouping for the iterative process model. The grouping is based on the dependency among stages in the SPM. However, selecting the most suitable software process model and grouping different stages in the life cycle is a decision made by the project management.

Site/Team Analysis

Step S1: Formulate the site dependency matrix

The relationship between available sites is identified through site dependency characteristics. According to our systematic literature review, the main site dependency characteristics have been identified [23-26]. They are time difference, culture difference, language difference and physical distance. In addition, collaboration maturity has been identified as an important factor which shows how long two sites have been working together. The political relationship between two countries shows the level of compatibility. All these factors can be used to measure how compatible two sites are. The political relationship between countries can be good, average and bad which is represented as high, medium and low, and is used as a common scale for all factors. The most favorable combination would be when collaboration maturity and political relationship are high and the other factors are low.

These site dependency qualitative variables are quantified in Table 1.

TABLE 1: QUANTIFICATION OF SITE DEPENDENCY CHARACTERISTICS

	TD	**CD**	**LD**	**PD**	**CM**	**PR**
High	0	0	0	0	1	1
Medium	0.5	0.5	0.5	0.5	0.5	0.5
Low	1	1	1	1	0	0

TD-Time Difference, CD-Culture Difference, LD-Language Difference, PD-Physical Difference, CM-Collaboration Maturity, PR-Political Relationship

Quantification in Table 1 can be used to identify compatible sites (locations), where the highest value indicates the most compatible sites.

Assume the work in Figure 4 needs to be distributed to four different sites located in Sri Lanka (SL), India, China and Australia (Aus). Site dependency between the four different locations will be quantified, based on Table 1 and the total value is shown in the last column in Table 2. If these sites have not worked together before, collaboration maturity will be low.

TABLE 2: LEVEL OF SITE DEPENDENCY

	TD	CD	LD	PD	CM	PR	Total
A-I	0.5	0	0	0	0	0.5	1
A-SL	0.5	0	0	0	0	0.5	1
I-SL	1	0.5	0.5	0.5	0	1	3.5
C-A	0.75	0	0	0	0	0.5	1.25
C-SL	0.75	0	0	0.5	0	0.5	1.75
C-I	0.75	0	0	0.5	0	0.5	1.75

Quantified site dependency information will be gathered in a site dependency matrix using the data from the last column in Table 2. A higher value shows the more compatible sites and a lower value shows less compatible sites. According to the matrix in Figure 5, India and Sri Lanka would be the most compatible sites.

	Aus	India	SL	China
Aus		1	1	1.25
India			3.5	1.75
SL				1.75
China				

Fig 5: Site Dependency Matrix

Step S2: Formulate site-specific matrix

A site-specific matrix is formulated by identifying attractive attributes in the process of offshoring location choice. According to the literature, labour cost [27] [28], skill level [29], availability of the site, economic stability and political stability [30] are identified as site specific parameters. Further, client proximity is an important factor in deciding where to allocate the work relating to system analysis and design. Based on the importance, site specific characteristics are classified as major factors and minor factors where labour cost, skill levels, client proximity and availability of the site are classified as major factors and economic stability and political stability are classified as minor factors. Figure 6 shows a sample site matrix with data relating to the four different locations previously mentioned.

	Aus	India	SL	China	
Labour rate	H	M	L	M	Major factors
Skill Level	M	H	H	H	
Availability	H	L	M	M	
Client proximity	H	L	L	L	
Economic stability	H	L	L	M	Minor factors
Political stability	M	L	L	M	

Fig 6: Site Matrix

Mapping Work to Site

Step M1: Analysis of four matrices

The distribution method can be seen as a function of work, work dependency, site, site dependency which needs to be mapped to achieve the project goal. The objective of the function can be defined as maximizing the goal vector.

f: (Work x Work dependency x Site x Site dependency) Mapping-> Goal

The analysis will start by suggesting work distribution considering one characteristic related to work and one characteristic related to site. Typically, the priority that is given to each suggestion in relation to work distribution is based on overhead reduction and optimization of cost, quality and time. Further, priority will depend on the project policy.

Step M2: Allocate work to suitable sites

Work distribution is done in a way which supports the achievement of the main objective of the project. The main objectives of any software projects are cost, quality and time. It is evident that in any given project, all these three factors are important. Depending on the project, the priority that is given to each of these three factors can vary. Therefore, there can be some changes to the work distribution mechanism, based on the main objective of the project and the project manager can define any project policy accordingly.

It is very clear that the optimization of one factor can result in reducing the acceptance level of other factors. Therefore, an over-focus on one factor may not lead to an acceptable product if the other factors are far below the expected standard. Hence, it is advisable to consider all the main factors at least to the minimum standard while placing more focus on the main objective. As the model is transparent, the project manager can take control to avoid this kind of situation.

IV. Conclusion And Future Work

The paper presents a work distribution mechanism considering phase-based configuration at an early stage of the Software Development Life Cycle (SDLC). The proposed method consists of two main phases: categorization and work distribution. The decision given by the method is based on multiple criteria and priority given for each criterion depends on the project objective. This framework is designed to suggest an approach for work distribution as an aid for the project manager. However, as the suggestions given by the method are transparent, the project manager is allowed to modify the recommendations at any time. The method provides an acceptable suggestion to increase the chance of successful GSD, based on the information available at an early stage,

Although the proposed method is mainly for phase-based configuration, it can be customized to use in module-based configuration as well. Moreover, the logic behind the work distribution mechanism can be used in any other industry which involves global delivery. Our future research will attempt to empirically validate the method using data from real world projects.

References

[1] P. Taylor, D. Greer, P. Sage, G. Coleman, K. McDaid, and F. Keenan, "Do agile GSD experience reports help the practitioner?," 2006, p. 93.

[2] W. DeLone, J. Espinosa, G. Lee, and E. Carmel, "Bridging global boundaries for IS project success," Proceedings of the 38th Hawaii International Conference on System Sciences, 2005.

[3] R. Aron and J. Singh, "Getting offshoring right," Harvard business review, vol. 83, p. 135, 2005.

[4] N. Huda, N. Nahar, J. Tepandi, and P. S. Deo, "Key barriers for global software product development organizations," in Management of Engineering & Technology, 2009. PICMET 2009. Portland International Conference on, 2009, pp. 1081-1087.

[5] D. Šmite and J. Borzovs, "Managing Uncertainty in Globally Distributed Software Development Projects," Datorzin tne un inform cijas tehnolo ijas, p. 9, 2008.

[6] P. Ågerfalk, B. Fitzgerald, H. Holmström Olsson, and E. Ó Conchúir, "Benefits of global software development: The known and unknown," Making Globally Distributed Software Development a Success Story, pp. 1-9, 2008.

[7] D. Damian and D. Moitra, "Guest Editors' Introduction: Global Software Development: How Far Have We Come?," Software, IEEE, vol. 23, pp. 17-19, 2006.

[8] R. E. Grinter, J. D. Herbsleb, and D. E. Perry, "The geography of coordination: dealing with distance in R&D work," 1999, pp. 306-315.

[9] K. J. Edwards HK, Park S, AI-Ani B, "Global software development: project decomposition and task allocation," International conference on business and information 2008.

[10] A. Lamersdorf, J. Munch, and D. Rombach, "Towards a multi-criteria development distribution model: An analysis of existing task distribution approaches," in International Conference on Global Software Development, 2008, pp. 109-118.

[11] A. Lamersdorf, J. Munch, and D. Rombach, "A survey on the state of the practice in distributed software development: criteria for task allocation," in International Conference on Global Software Development, 2009, pp. 41-50.

[12] A. Lamersdorf and J. Munch, "TAMRI: a tool for supporting task distribution in global software development projects," in International Conference on Global Software Development, 2009, pp. 322-327.

[13] A. Lamersdorf, J. Münch, and D. Rombach, "A decision model for supporting task allocation processes in global software development," Product-Focused Software Process Improvement, pp. 332-346, 2009.

[14] A. Lamersdorf and J. Münch, "A multi-criteria distribution model for global software development projects," Journal of the Brazilian Computer Society, vol. 16, pp. 97-115, 2010.

[15] S. Beecham, J. Noll, I. Richardson, and D. Dhungana, "A Decision Support System for Global Software Development," in International conference on Global Software Engineering Workshops, 2011, pp. 48-53.

[16] J. Münch and A. Lamersdorf, "Systematic task allocation evaluation in distributed software development," 2009, pp. 228-237.

[17] S. Doma, L. Gottschalk, T. Uehara, and J. Liu, "Resource Allocation Optimization for GSD Projects," Computational Science and Its Applications–ICCSA 2009, pp. 13-28, 2009.

[18] S. Setamanit, W. Wakeland, and D. Raffo, "Using simulation to evaluate global software development task allocation strategies," Software Process: Improvement and Practice, vol. 12, pp. 491-503, 2007.

[19] J. A. Espinosa and E. Carmel, "The effect of time separation on coordination costs in global software teams: A dyad model," 2004, p. 10 pp.

[20] J. Herbsleb, A. Mockus, T. Finholt, and R. Grinter, "An empirical study of global software development: distance and speed," 2001, pp. 81-90.

[21] K. Kurbel, "Process Models and Distribution of Work in Offshoring Application Software Development," Diskussionspapiere/Europa-Universität Viadrina Frankfurt (Oder), Wirtschaftswissenschaftliche Fakultät2007.

[22] R. Madachy, "Cost modeling of distributed team processes for global development and Software-Intensive Systems of Systems," Software Process: Improvement and Practice, vol. 13, pp. 51-61, 2008.

[23] E. O. Conchuir, H. Holmstrom, J. Agerfalk, and B. Fitzgerald, "Exploring the assumed benefits of global software development," in International conference on Global Software Engineering (ICGSE), 2006, pp. 159-168.

[24] D. Smite, C. Wohlin, R. Feldt, and T. Gorschek, "Reporting empirical research in global software engineering: a classification scheme," 2008, pp. 173-181.

[25] H. Holmstrom, E. O. Conchuir, J. Agerfalk, and B. Fitzgerald, "Global software development challenges: A case study on temporal, geographical and socio-cultural distance," 2006, pp. 3-11.

[26] P. J. Ågerfalk, B. Fitzgerald, H. Holmström, B. Lings, B. Lundell, and E. O. Conchúir, "A framework for considering opportunities and threats in distributed software development," 2005, pp. 47-61.

[27] D. Farrell, "Offshoring: Value creation through economic change," Journal of Management Studies, vol. 42, pp. 675-683, 2005.

[28] A. Lewin, M. Peacock, C. Peeters, J. Russell, and G. Sutton, "2nd bi-annual offshore survey results, December 2005," Duke University CIBER/Archstone Consulting, Durham, NC, 2005.

[29] J. P. Doh, "Offshore outsourcing: Implications for international business and strategic management theory and practice," Journal of Management Studies, vol. 42, pp. 695-704, 2005.

[30] G. G. S. Suder, Terrorism and the international business environment: the security-business nexus: Edward Elgar Publishing, 2004.