2011 Sixth IEEE International Conference on Global Software Engineering Workshops

Follow-The-Sun Methodology in a Stochastic Modeling Perspective

Ricardo M. Czekster, Paulo Fernandes, Rafael Prikladnicki,
Afonso Sales, Alan R. Santos, Thais Webber

Pontifícia Universidade Católica do Rio Grande do Sul
Av. Ipiranga, 6681 – Prédio 32 – 90619-900 – Porto Alegre, Brazil
{ricardo.czekster, paulo.fernandes, rafael.prikladnicki}@pucrs.br
afonso.sales@pucrs.br, alan.ricardo@acad.pucrs.br, thais.webber@pucrs.br

***Abstract*—Stochastic modeling of globally distributed projects has become a way to evaluate the performance of teams working in different time zones. The interest in extracting and analyzing quantitative data from geographically dispersed teams has grown throughout the years as major development companies were attracted by potential benefits. We direct our attention to Follow-The-Sun (FTS), a special case of globally distributed projects, where work is handed off every day from one development site to the next on a different time zone. The main benefit of FTS is to reduce project duration providing continuous software development. Our objective in this paper is to discuss a formal mapping of FTS characteristics to a stochastic model in order to predict performance indices of teams such as availability and risk assessment. The modeling effort aims to enhance understanding and feasibility evaluation for FTS projects.**

***Keywords*—Global Software Development; Follow-the-Sun; Performance Evaluation; Stochastic Modeling.**

I. Introduction

Companies are looking for ways to re-structure their IT area extending operations to offshore software development centers. Furthermore, there are research challenges in the software development field such as heterogeneous team configurations distributed in separate sites, or even different time zones, inter-cultural factors, different experience levels and technical background [1], [2], [3], [4].

Theoretically, with the use of 24-h software development, the software development cycle time can be significantly reduced. This kind of development may become a valid choice for software companies that can be applied in numerous projects [5].

Follow-The-Sun (FTS) is essentially focused on speed where the project is designed to reduce cycle-time (also known as time-to-market reduction, or duration reduction) [6]. FTS is a subset of global software development and shares many issues and challenges with global software development in terms of coordination, culture, and communication [6].

In some domains such as health-care and air traffic control, working throughout the day is a common practice [1]. In respect to software engineering, however, the benefits of around the clock development can be achieved through work transfer across time zones, thereby eliminating the need for unsociable working hours [1].

On this context, new methodologies are needed to evaluate the hand-off process in terms of performance and its impact on the overall project duration, considering communication issues and details of the task being shifted to the next site to be continued.

Carmel *et al.* [6] proposed a long-term research strategy for FTS, with the goal of improving measures. One of the action points identified was to develop research frameworks for time separation and to create mathematical and simulation models to better understand how to optimally architect FTS and how to evaluate its costs and benefits.

The objective of this paper is to present a formal mapping of FTS characteristics to a stochastic model in order to evaluate teams in FTS projects. Moreover, we demonstrate the benefits of using the Stochastic Automata Networks (SAN) formalism for the modeling and evaluation of distributed teams, since SAN provides a modular description with functional primitives. The modeling effort presented aims to enhance understanding and feasibility evaluation for FTS projects calculating probabilities for availability and project risk factor, such as hand-off efficiency, considering the conceptual framework proposed by Carmel. Conclusions point out future works directed to model extensions to capture advanced characteristics such as cultural issues, different levels of expertise and diverse communication problems.

II. Follow-The-Sun Methodology

Follow-The-Sun (FTS) development can be simply defined as software related tasks being hand-off from one site to the next, on a daily basis [7], [8], where 24-hour development can be achieved considering time differences among teams. Each team work on its own regular workday and, at the end of its shift, the task is handed off to the next team, which continues the work from the point the earlier team left off. In the context of global software engineering, Treinen and Miller-Frost [8] have discussed the time difference as an advantage to distribute the teams to achieve a 24-hour software development environment.

FTS is recently also referenced in the literature as *round-the-clock* [6], [9] and *around-the-clock* [10], [9]. The round-the-clock and around-the-clock are related to the 24-hour develop-

978-0-7695-4558-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ICGSE-W.2011.26

ment of a given project, in different work shifts. Visser [9] has defined FTS as specific software projects where development teams are spread across multiple time zones, focusing on the sharing of source code and relevant information at the end of each shift, summarizing 24 hours of continuous development.

Additionally, FTS development can be extended to different tasks besides the development. Gupta *et al.* [11] have introduced the 24-hour continuous development, considering the concept of knowledge being transferred, for example, among development and test teams. The tasks handed off between distributed teams, at the end of each workday, are basically related to the knowledge about the software itself and its requirements. Carmel, Dubinsky and Espinosa [6] have proposed FTS development as a kind of global knowledge workflow that can reduce project duration, depending on factors such as hand-off efficiency, effective communication and coordination.

In practice, FTS usage was firstly documented in 1997 when IBM decided to accelerate the development process in a given multi-site project [3]. The project was composed of five teams located on development centers of five distinct countries. Many problems were found during project execution, especially during the daily hand-offs, forcing the use of another GSD strategy, instead of FTS, to accomplish the tasks. But, FTS is being experimented to reduce development phase duration, opening research opportunities in the field of team building, global software development tools, and coordination strategies.

III. Stochastic Modeling Application

Analytical modeling in software engineering contexts has been successfully used to provide quantitative performance measures [12], [13]. Stochastic models and simulation schemes have been developed towards to the evaluation of software project dynamics [14], analysis of teams productivity variability [2], and different performance predictions about geographically dispersed teams [3], [15], [16], [17]. However, advances are still needed for the quantitative evaluation of teams using stochastic modeling as a tool [18].

Analytical modeling formalisms are commonly applied to describe systems in a state-based approach. Markov chains [19] and Markovian-based formalisms (such as, Queueing Networks [20] and Stochastic Automata Networks (SAN) [21], [22]) are already employed in several domains, such as economics, physics, engineering and bioinformatics, to cite a few. Nevertheless, SAN allows the stochastic modeling of systems in a high-level description format for representing a given system by subsystems (*i.e.*, components). The modeled subsystems can have an *independent behavior* and *occasional interdependencies* [21], not necessarily strict to the modeling of queueing behavioral aspects. Moreover, SAN formalism presents several computational advantages related to modularization and powerful features, such as *functional dependencies* and *synchronizing events* [22].

Furthermore, SAN is specifically suitable for modeling globally distributed projects due to the fact that teams can be abstracted in a modular manner with a discrete set of states and events. In essence, a component (or module) is described by a *stochastic automaton*, where the transitions between states are labeled with probabilistic and timing information. A SAN model also has a set of events, which can change the sate of one or more automata. Each event has an estimated rate (or duration), which indicates how often the event occurs. Numerically solving an analytical model, one can obtain its steady-state probabilities [19] and, from these probabilities extract measures of interest about the system under evaluation (*e.g.*, team's performance indices, resources availability, costs, and reliability). In addition, a SAN model easily allows to investigate different team configurations only changing input parameters to achieve better performance results [23].

Traditionally, scientific works on software engineering area present different applications for mathematical models, *e.g.*, automated software testing processes [24], [25], and quantitative evaluation of software development teams also evaluating project risks [26], [27]. Global software development modeling is a challenge already discussed by Czekster *et al.* [18]. On this context, the effort on mapping real characteristics of FTS projects to a stochastic model requires the modeling of factors, such as time zone differences among teams (sites), reporting time on the end of a work shift and the catching up time on the start of other work shift. As occurs in any other modeling exercise, the difficult part in modeling is to identify all variables (or the most representative ones) that should be abstracted as well as figure out their occurrence rates and/or probabilities. Specifically on FTS projects, the initial model mapping should consider hand-off durations and the frequency related to other activities to perform some basic scenario evaluations. Next section presents the FTS mapping to a stochastic model description using SAN.

IV. FTS mapping to stochastic models

This section presents a mathematical modeling of specific aspects of the conceptual framework for FTS proposed by Carmel *et al.* [28]. The framework aims to help the analysis of the conditions where FTS can be beneficial in terms of reducing the project duration, *i.e.*, time-to-market.

In this work we propose the mapping of FTS to an analytical model using the SAN formalism primitives and available tools for numerical solution [29], [30]. The strong feature of SAN is to model parallel entities where the problem can be modularized, easing the analysis of multi-site projects, where work is divided among geographically dispersed teams. Sometimes, teams must synchronize activities or assign new tasks, revisions or tests to be performed in other sites.

We have centered this first modeling effort on the hand-off efficiency concept [28] to evaluate the probabilities of each site being reworking activities or waiting for clarifications in a FTS project. Usually FTS projects have strict rules imposed on their interactions using asynchronous hand-offs between teams at fixed intervals of time.

Evidently, this attempt to model these contexts is far from being conclusive, but provides some insights about the modeling to measure the impact of hand-off efficiency on the time spent working/reworking by each site or team. For the composition of our model's states and events, we have made assumptions related to FTS characteristics and behavior (refer to Section II) as follows.

We assume in our abstraction that the workload (workday task) is equally complex for all distributed teams and that all the resources needed by a project are assigned at the start and readily available throughout its conclusion.

Also we are not considering for this initial research proposal how factors such as different time zones, different cultures, software quality aspects, teams size, tasks complexity and workload impact on the FTS model.

We instantiate a stochastic model having a set of s multiple sites (*e.g.*, each site or team encapsulating n members) working in different time-zones. We are considering multi-variate conditions, where sites can shift their work in an efficient (or inefficient) manner, depending on the project configuration.

The modeling assumptions for the FTS formal mapping in a stochastic automata network can be described as follows:

- each site acts as an agile team entity that works in a set of tasks, reworks tasks, spends some time closing a workday in hand-off, and stays off-line until the next workday in hand-off opening;
- different time zone for each site or team (2, 3 or even 4 sites in a 24-hour development);
- homogeneous task size - related to a workday task - *task time* assuming 6-hour *per* site, for instance, in a 4-site FTS project;
- after hand-off opening in a site, there is a probability for reworking tasks, as well as after a team spend some time working, there is a probability for reworking, and a probability for hand-off closing (*e.g.*, given team's expertise);
- there is an online shared repository which allows sites to *commit* work at the end of the workday;
- hand-offs are asynchronous (the work is shifted without direct teams' interactions, meaning the current team goes off-line after the hand-off process ended enabling to shift the work to the next team).

GSD projects have became an important research issue in IT companies [1], [2], [3], however there is still a lack of concern in relation to teams' performance globally distributed. On this context, stochastic modeling is a feasible and flexible alternative to abstract GSD teams configuration in environments such as follow-the-sun to measure teams' performance. Moreover, the stochastic modeling can be used during project planning phases in order to find relevant evidence in terms of performance analysis that could help the planning improvement.

The motivation for the modeling exercise is to provide a performance modeling and evaluation process by presenting an extensible mathematical model where its utilization is described in a few easy steps. The only concern for modelers will be directed towards the parametrization of the model with their own measures and how to extend it with new event compositions. Once the model is created, the next step is its solution, a mathematical procedure that converts the textual representation into numerical results.

The computed performance indices are actually a set of quantitative measures from the model, *i.e.*, transient or steady-state probabilities that allow us to estimate the impact of the hand-off efficiency on the overall project duration. If, however, some other FTS feature needs analysis (other than hand-off efficiency, the focus of our approach in this paper), the model could be used to inspire future instantiations of different models. In this sense, the approach adopted here allows a minimal FTS setting where modelers can extend it at will to convey other behaviors or to test novel configurations (perhaps optimal ones or settings where teams begin to somehow to be underperformed).

A. Asynchronous hand-off SAN model

Fig. 1 shows an example of FTS composed of three sites across the world. In the asynchronous hand-off point of view, the sites are unable to interact with the site that shifted the work due to different time zones, for example.

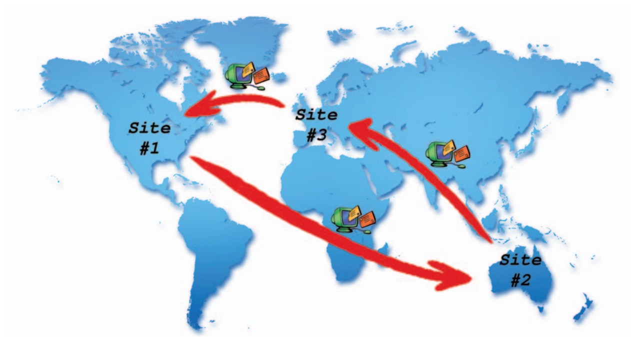

Fig. 1. FTS context with asynchronous hand-off

The stochastic automaton representing each site is a five-state model composed of states that can be assumed by the site at each time in a workday:

- ***Off-line*** state, the site is off-line due to time zone, so it is unreachable for collaboration, communication or interaction;
- ***Opening*** hand-off state, beginning of the workday;
- ***Working*** state, actual work is being performed in the site (active workday);
- ***Reworking*** state, misunderstandings occurred mainly in the hand-off process lead to rework [31];
- ***Closing*** hand-off state, end of the workday.

Fig. 2 shows the stochastic automata network model representing three sites in a FTS configuration. This model is composed of three sites, but it can be instantiated for 2, 3, or 4 sites (*e.g.*, considering 24-hour software development with 6-hour shift *per* site) for a model with four sites. The individual behavior of a given *Site #i*, where i is the index of the site, is represented by the following events: $open_i$, wk_i, rw_i, nt_i, cl_wk_i, cl_rw_i, and off_i. Following we explain in detail each event dynamics for enabling transitions in the mathematical model:

- ***open**$_i$* – local event for transferring the work/knowledge to the next site after the closing hand-off be completed. This event enables the transition from *Off-line* to *Opening hand-off* state in the *Site #i*, meaning that the site is *starting* a workday;
- ***wk**$_i$* – local event in a *Site #i* responsible for enabling the transition from the *Opening hand-off* state to the *Working* or *Reworking* states with a given probability π_i or probability 1-π_i, respectively;
- ***rw**$_i$* – local event responsible for enabling the transition from *Working* to *Reworking* state, where a task must be reworked;
- ***nt**$_i$* – when a rework is performed, this local event is responsible for enabling the transition from *Reworking* to *Working* state, representing a *new task* starting to be executed;
- ***cl_wk**$_i$* – local event in a *Site #i* that enables the transition from *Working* to *Closing hand-off* state;
- ***cl_rw**$_i$* – local event representing the change from *Reworking* to *Closing hand-off* state in a *Site #i*;
- ***off**$_i$* – after close a hand-off, this local event in a *Site #i* is responsible for enabling the transition from *Closing hand-off* to *Off-line* state, meaning that the site is *closing* a workday;

In this SAN model (Fig. 2), event wk has an associated function that allows (or not) the occurrence of this event. Since we are modeling an asynchronous hand-off, a site can open its hand-off process without changing the state of earlier or next sites. However, each site has a dependency of the state from the earlier site to start to work or rework a task. In this asynchronous model, a function is associated to event wk, which verifies if the earlier site's hand-off has been completed. For instance, event wk_1 can occur if *Site* #3 is in *Off-line* state, *i.e.*, if *Site* #3 has closed its hand-off. Otherwise, event wk_1 cannot occur and need to wait the closure of hand-off from *Site* #3.

The maximum number of sites in a daily cycle is finite, due to the 24 hours in a day. More sites provide more working capacity, however also require more overhead and increase the likelihood of mistakes [32].

With a better understanding of the benefits and constraints, adequate communication and coordination environments can be developed to support the tight coordination that is necessary to reap the benefits of the 24-hour model [33].

According to Jalote and Jain [33] a more realistic model will consider time slots that are overlapping. However in this paper we focus our attention on the asynchronous model and a synchronous model will be part of a future research.

This section presented the mapping of the interaction pattern of development sites under FTS methodology for a SAN model. The main entities (*Site* #1, *Site* #2 and *Site* #3) are represented by automata. The abstraction, for instance, represents the hand-off within a project: some of the time is spent on *Opening hand-off*, *Closing hand-off* and the rest in *Working* or *Rework* states.

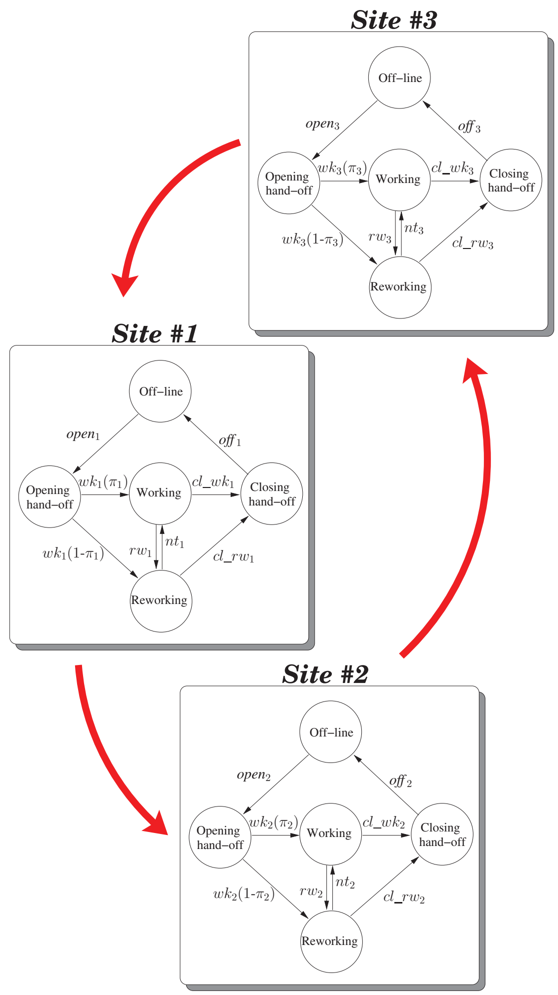

Fig. 2. SAN model with asynchronous hand-off

V. SAN MODEL PARAMETERISATION

An important phase of analytical modeling is the estimation of event rates, because the dynamic of analytical modeling is given by the event transitions between model's states.

In a SAN model, an event is the entity responsible for the transition between model states. Local type events (loc) are events that change a local state of a given automaton in the model and they are used to demonstrate the individual processing behavior of each automaton. Synchronizing type events (syn) are events that simultaneously change states of one or more automata and they are used to describe the interaction between two or more automata.

Table I shows examples of durations for the events in the Follow-The-Sun model previously presented, assuming as reference an *eight-hour workday*.

These values were chosen considering average values selected ad-hoc. The values of probabilities π_1, π_2 and π_3 are defined for example as 0.5, meaning in average that 50% of the time the site continues a task from the earlier site and 50% needs to rework a task.

TABLE I
ESTIMATED EVENT RATES - ASYNCHRONOUS HAND-OFF MODEL

Type	Event	Description	Average time
loc	$open$	After being off-line for (in average) 16 hours, a site initiates the hand-off opening process.	16h
loc	wk	A site spends in average 1 hour in the beginning of the workday performing the hand-off opening process.	1h
loc	rw	A site works in average 1 hour *per* workday before reworking a pending issue.	1h
loc	nt	A site remains reworking a task in average 0.5 hour before starting a new task.	0.5h
loc	cl_wk	Before starting the hand-off closing process, a site remains working in average 4 hours *per* workday.	4h
loc	cl_rw	Before going to a hand-off closing process, a site stays in average 0.5 hour reworking a task.	0.5h
loc	*off*	A site spends in average 1 hour in the end of the workday executing the hand-off closing process.	1h

Notice that some issues should be also considered for modeling states and related parameterisation:

- the time spent in actual work, or more specifically, the probability of a team being working, does not mean an index of productivity, but represent the amount of time in a workday a team has (in average) to be productive, which can be a guide for estimating project durations combined to a quantitative analysis of team's expertise;
- if the proposed model aims to predict overlap on team work shifts (in case of synchronous hand-off), it should be adapted to allow opening hand-off activity concurrent to the closing hand-off activity on the earlier site. Another possibility is to consider that teams agreed upon being flexible in terms of extending their working shifts for tasks clarification. Literature studies [8], [9] point out cases where is recommended a *sticky hand-off*, *i.e.*, intense interactions are more favorable than a *clean hand-off* (*drop-and-go* approach);
- to support team design decisions the model is easily extended to more or less sites, with different working hours, reworking probabilities, average time spent in hand-off activities, given a specific project (in this case is important to quantify team knowledge or past experiences). The modeler need to understand more deeply the project and the participants' profile to collect significant data from interviews, surveys or historical data in companies databases;
- cultural issues are not explored quantitatively in this paper, but remark that some risky aspects such as frequent clarifications can be related to this matter, as well as related to some lack of communication or coordination support, physical distance, level of domain knowledge, to name a few.

Once we have the model with states, transitions, events and their associated rates, the next stage is to extract the model's performance indices, running our proposed model on specialized software tools [29], [30]. The basic indices we could extract are the steady-state probabilities of each site being in the *Working* or *Reworking* states, since can occur some delay in the hand-off process. Using the probabilities of being in the *Opening hand-off* state we can extract costs and calculate delays related to the earlier site, as well as the probability of being in *Closing hand-off* state can give us indications that many clarifications are needed or, in the best case, a team is working faster on its tasks than the next site.

For this further analysis, new states could also be added in the model (expanding the current state abstraction) in order to verify other project aspects and possible bottlenecks.

Moreover, the impact of a late hand-off closing process on the working activity in a distributed site, or the probability of a hand-off opening process that lasts more than a given duration, both can be measured using our simple model, including comparisons to other hand-off configuration scenarios.

VI. CONCLUSION

In this paper we have presented an abstraction of how to model development teams in Follow-The-Sun (FTS) context and we have proposed a scenario that enables the investigation of overall FTS project performance. This is an initial approach that showed how the combination of stochastic modeling and FTS could be mixed to produce important considerations to be used by decision makers, *i.e.*, solving the stochastic models it is possible to obtain evidence of bottlenecks in a given project before the project execution.

We are aware that this paper does not capture all important dimensions of FTS projects such as specific cultural and language barriers, communication patterns considering social networks, or other difficulties in teams coordination. Stochastic Automata Network (SAN) allows new system compositions and behaviors by simply appending new states and relationships among previously defined entities. This is the major characteristic of SAN where modelers can profit these features to easily depict more intricate behavior. The main contribution of our work is to propose a conceptual modeling for the analysis of the hand-off efficiency for projects in a FTS global software development context. As future work users can also numerically analyze the system major bottlenecks, *i.e.*, where the lack of hand-off is bound to happen for a given project setup.

The proposed model focused on the hand-off efficiency and discussed how to parameterize asynchronous models with different measures or observations from different project aspects.

In this sense, we would like to stress that the model proposed here could be extended to a creation of new synchronous model and re-parameterize to inspire new analysis and new configurations. It could be also used, for instance, to numerically evaluate the effectiveness of FTS itself and which types of projects are more suitable for this approach and the ones where global software development would perform better.

ACKNOWLEDGMENTS

Paulo Fernandes is funded by CNPq-Brazil (PQ 307284/2010-7). Afonso Sales receives grants from CAPES-Brazil (PNPD 02388/09-0). Ricardo Czekster and Rafael Prikladnicki are funded by the PDTI program, financed by Dell Computers of Brazil Ltd. (Law 8.248/91). The order of authors is merely alphabetical.

REFERENCES

[1] A. Taweel and P. Brereton, "Modelling software development across time zones," *Information and Software Technology*, vol. 48, no. 1, pp. 1–11, 2006.

[2] S. Faraj and L. Sproull, "Coordinating expertise in software development teams," *Management Science*, vol. 46, no. 12, pp. 1554–1568, 2000.

[3] E. Carmel, *Global software teams: collaborating across borders and time zones*. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1999.

[4] C. U. Smith and L. G. Williams, "Software Performance Engineering: A Case Study Including Performance Comparison With Design Alternatives," *IEEE Transactions on Software Engineering*, vol. 19, no. 7, pp. 720–741, 1993.

[5] P. Sooraj and P. K. J. Mohapatra, "Modeling the 24-hour software development process," *Strategic Outsourcing: An International Journal*, vol. 1, no. 2, pp. 122–141, 2008.

[6] E. Carmel, Y. Dubinsky, and J. A. Espinosa, "Follow The Sun Software Development: New Perspectives, Conceptual Foundation, and Exploratory Field Study," in *Proceedings of the 42nd Hawaii International Conference on System Sciences*. Waikoloa, Big Island, HI, USA: IEEE Computer Society, 2009, pp. 1–9.

[7] I. Gorton and S. Motwani, "Issues in co-operative software engineering using globally distributed teams," *Information and Software Technology*, vol. 38, no. 10, pp. 647–655, 1996.

[8] J. J. Treinen and S. L. Miller-Frost, "Following the sun: case studies in global software development," *IBM Systems Journal*, vol. 45, pp. 773–783, October 2006.

[9] C. K. Visser, "Connecting Time Zones and Languages in Follow-the-Sun: a routing model," Master's thesis, Delf University of Technology, Nederlands, 2009.

[10] M. Yap, "Follow the Sun: Distributed Extreme Programming Development," in *Proceedings of the Agile Development Conference (ADC '05)*. Denver, CO, USA: IEEE Computer Society, 2005, pp. 218–224.

[11] A. Gupta, E. Mattarelli, S. Seshasai, and J. Broschak, "Use of collaborative technologies and knowledge sharing in co-located and distributed teams: Towards the 24-h knowledge factory," *The Journal of Strategic Information Systems*, vol. 18, pp. 147–161, September 2009.

[12] J. D. Herbsleb and D. Moitra, "Global software development," *IEEE Software*, pp. 16–20, 2001.

[13] J. D. Herbsleb, A. Mockus, T. A. Finholt, and R. E. Grinter, "Distance, dependencies, and delay in a global collaboration," in *Proceedings of the 2000 ACM conference on Computer Supported Cooperative Work (CSCW'00)*. Philadelphia, PA, USA: ACM, 2000, pp. 319–328.

[14] A. Mockus and J. D. Herbsleb, "Expertise browser: a quantitative approach to identifying expertise," in *Proceedings of the 24th International Conference on Software Engineering*, 2002, pp. 503–512.

[15] N. S. Shami, N. Bos, Z. Wright, S. Hoch, K. Y. Kuan, J. Olson, and G. Olson, "An experimental simulation of multi-site software development," in *Proceedings of the 2004 conference of the Centre for Advanced Studies on Collaborative research*. IBM Press, 2004, pp. 255–266.

[16] H. Zhang and B. Kitchenham, "Semi-quantitative Simulation Modeling of Software Engineering Process," in *Proceedings of the International Software Process Workshop and International Workshop on Software Process Simulation and Modeling*, vol. 3966. Shanghai, China: Springer-Verlag Berlin Heidelberg, 2006, pp. 242–253.

[17] S. Ferreira, J. Collofello, D. Shunk, and G. Mackulak, "Understanding the effects of requirements volatility in software engineering by using analytical modeling and software process simulation," *Journal of Systems and Software*, vol. 82, pp. 1568–1577, October 2009.

[18] R. M. Czekster, P. Fernandes, A. Sales, and T. Webber, "Analytical Modeling of Software Development Teams in Globally Distributed Projects," in *International Conference on Global Software Engineering (ICGSE'10)*. Princeton, NJ, USA: IEEE Computer Society, 2010, pp. 287–296.

[19] W. J. Stewart, *Introduction to the numerical solution of Markov chains*. Princeton University Press, 1994.

[20] G. Bolch, S. Greiner, H. de Meer, and K. S. Trivedi, *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. New York, NY, USA: John Wiley & Sons, Inc., 1998.

[21] B. Plateau, "On the stochastic structure of parallelism and synchronization models for distributed algorithms," *ACM SIGMETRICS Performance Evaluation Review*, vol. 13, no. 2, pp. 147–154, August 1985.

[22] L. Brenner, P. Fernandes, and A. Sales, "The Need for and the Advantages of Generalized Tensor Algebra for Kronecker Structured Representations," *International Journal of Simulation: Systems, Science & Technology*, vol. 6, no. 3-4, pp. 52–60, February 2005.

[23] P. Fernandes, A. Sales, A. R. Santos, and T. Webber, "Performance Evaluation of Software Development Teams: a Practical Case Study," in *Proceedings of the 5th International Workshop on Practical Applications of Stochastic Modelling (PASM)*, Karlsruhe, Germany, March 2011, pp. 5–21.

[24] C. Bertolini, A. G. Farina, P. Fernandes, and F. M. Oliveira, "Test Case Generation Using Stochastic Automata Networks: Quantitative Analysis," in *Proceedings of the 2nd International Conference on Software Engineering and Formal Methods (SEFM'04)*, Beijing, China, 2004, pp. 251–260.

[25] A. G. Farina, P. Fernandes, and F. M. Oliveira, "Representing software usage models with Stochastic Automata Networks," in *Proceedings of the 14th International Conference on Software Engineering and Knowledge Engineering (SEKE'02)*, Ischia, Italy, 2002, pp. 401–407.

[26] K. A. Buragga, "An analytical model for requirements activities and software product quality," *Journal of Computational Methods in Science and Engineering*, vol. 6, pp. 205–215, 2006.

[27] A. Avritzer and A. Lima, "An Empirical Approach for the Assessment of Scheduling Risk in A Large Globally Distributed Industrial Software Project," in *Proceedings of the 4th International Conference on Global Software Engineering (ICGSE'09)*, Limerick, Ireland, 2009, pp. 341–346.

[28] E. Carmel, Y. Dubinsky, and J. A. Espinosa, "Follow The Sun Workflow In Global Software Development," *Journal of Management Information Systems*, vol. 27, no. 1, pp. 17–38, 2010.

[29] L. Brenner, P. Fernandes, B. Plateau, and I. Sbeity, "PEPS 2007 - Stochastic Automata Networks Software Tool," in *Proceedings of the 4th International Conference on Quantitative Evaluation of SysTems (QEST 2007)*. Edinburgh, Scotland: IEEE Computer Society, September 2007, pp. 163–164.

[30] R. M. Czekster, P. Fernandes, and T. Webber, "GTA express - A Software Package to Handle Kronecker Descriptors," in *Proceedings of the 6th International Conference on Quantitative Evaluation of SysTems (QEST 2009)*. Budapest, Hungary: IEEE Computer Society, September 2009, pp. 281–282.

[31] J. A. Espinosa and E. Carmel, "The impact of time separation on coordination in global software teams: a conceptual foundation," *Software Process: Improvement and Practice*, vol. 8, no. 4, pp. 249–266, 2003.

[32] R. van Solingen and M. Valkema, "The impact of number of sites in a follow the sun setting on the actual and perceived working speed and accuracy: A controlled experiment," in *Global Software Engineering (ICGSE), 2010 5th IEEE International Conference on*, aug. 2010, pp. 165 –174.

[33] P. Jalote and G. Jain, "Assigning tasks in a 24-hour software development model," in *Asia-Pacific Software Engineering Conference*, 2004, pp. 309–315.