The Journal of Systems and Software 100 (2015) 182–194

Contents lists available at ScienceDirect

The Journal of Systems and Software

journal homepage: www.elsevier.com/locate/jss

Using SAN formalism to evaluate Follow-The-Sun project scenarios

Alan R. Santos*, Afonso Sales[1], Paulo Fernandes[2]

Pontifícia Universidade Católica do Rio Grande do Sul, Avenida Ipiranga, 6681 – Prédio 32, 90619-900 Porto Alegre, RS, Brazil

ARTICLE INFO

Article history:
Received 21 January 2014
Received in revised form 21 October 2014
Accepted 23 October 2014
Available online 31 October 2014

Keywords:
Projects simulation
Follow-The-Sun
Stochastic modeling

ABSTRACT

Performance evaluation of projects can be used by companies and institutions as a tool to help the decision making process of Follow-The-Sun (FTS) projects. This paper main goal is to discuss a stochastic model definition to evaluate the performance of different aspects of FTS projects. Examples that can be addressed using the FTS model are provided with results comparing different model instances to evaluate aspects such as project execution time and project costs composition.

1. Introduction

Nowadays companies have been working to restructure their IT departments by extending their operations using software development centers geographically distributed in different timezones. In this context, software engineering has a field known as *GSD – Global Software Development*. For Sarker and Sahay (2002): *"GSD refers to software development geographically, remotely or globally distributed, with the goal of rationalize the process of product development"*. There are many challenges in this research field that must be better studied and explored, such as different timezones, different cultures, different levels of experience and different technical backgrounds (Taweel and Brereton, 2006; Faraj and Sproull, 2000; Carmel, 1999).

GSD has recently become an active research area (Prikladnicki and Audy, 2010). *Follow-The-Sun* (FTS) is a type of global software development that: aims the use of a 24h workday; focus on projects that search for velocity by the reduction of project life cycle *time-to-market* (Carmel et al., 2009, 2010), and shares many challenges and issues of global software development, such as coordination, cultural factors and communication issues (Carmel, 1999; Carmel et al., 2010; Espinosa et al., 2003).

Treinen and Miller-Frost (2006) presented timezone difference as an advantage for teams distribution in order to create a 24 h development environment. However, according to Prikladnicki and Carmel (2013) *"differences in time zones are more difficult to overcome than distance separation"*.

A strategy that also takes time zone differences as an advantage is Round-the-Clock. FTS is about unfinished work that is handed off to the next site on a daily basis, whereas Round-the-Clock covers global help desks, support centers, and others (Carmel and Espinosa, 2011).

We belief that some FTS challenges and issues can be mapped through the use of analytical models for analysis of projects behaviors in order to facilitate companies and institutions decision making process, providing a better understanding of possible issues that can occur in geographically distributed projects.

Houston et al. (2001) have described an approach for modeling risk factors simulating their effects as means of supporting certain software development risk management activities. Related works to stochastic modeling and simulations have been developed through the dynamics of software projects (Cummings et al., 2009; Padberg, 2002) as the usage of analytical modeling for variability performance analysis of software development teams (Czekster et al., 2010; Avritzer and Lima, 2009). SAN - *Stochastic Automata Networks* (Plateau, 1985; Brenner et al., 2005) is a formalism based on Markov Chains (Stewart, 1994) that provides a modular and a high-level abstraction of the model description.

New methodologies are needed to evaluate and simulate projects performance scenarios, considering cost, time and quality. This research aims to apply a SAN model to predict behavior of FTS software projects. The use of Follow-The-Sun strategy for the entire project life cycle can be complex and maybe not feasible, few studies have explored FTS development and there is little evidence of success (Carmel et al., 2010). In this sense there is a research space to use SAN as a tool for performance evaluation in order to enhance the decision making process of FTS projects.

---

* Corresponding author. Tel.: +55 51 3320 3611; fax: +55 51 3320 3621.
E-mail addresses: alan.santos@pucrs.br (A.R. Santos), afonso.sales@pucrs.br (A. Sales), paulo.fernandes@pucrs.br (P. Fernandes).

[1] Afonso Sales is also funded by CNPq-Brazil (Universal 470096/2013-6).

[2] Paulo Fernandes is also funded by CNPq-Brazil (PQ 307602/2013-3).

http://dx.doi.org/10.1016/j.jss.2014.10.046

1.1. Follow-The-Sun (FTS)

Innovation has been moving companies and institutions to create new strategies and techniques for software development. Timezone differences, geographical diversity and other aspects actually can help companies to improve their business. These techniques and strategies can move companies to use an entire day of work with the Follow-The-Sun approach.

Follow-The-Sun is a type of complex software development approach. In fact, FTS can be defined as software development activities being handoff from one site to the next one on a daily basis, where it is possible to work on software development 24 h a day using timezone differences between teams as an advantage (Gorton and Motwani, 1996; Treinen and Miller-Frost, 2006). When a team finishes its regular working hours, another team located in another location and time zone starts its workday (Kroll et al., 2013), *i.e.*, each team works on a timezone and in the end of each day tasks are handoff to the next team that will continue working on tasks started by the previous team. Handoff is the process of activities transition from the current team shift to the next team shift.

Carmel et al. (2009) proposed FTS development as a type of global knowledge *workflow* that can reduce projects duration, depending on factors such as handoff efficiency, effective communication and coordination. An IBM FTS experience was published on 1997, the experience was based on five different teams in five different software development centers and during the project they had several coordination issues on daily *handoffs* (Carmel et al., 2009).

Espinosa et al. (2007) conducted a FTS study with a research question that guides whether there are gradual differences across timezones which would impact in team performance. In their study they conducted a laboratory experiment with 42 dyadic teams. The teams were randomly assigned into four timezone overlap conditions: full overlap, 2/3 overlap, 1/3 overlap and no overlap. Using a fictional map task, they found that a small time separation has no effect on accuracy, but that more time separation has a significant effect on accuracy. Another fact was that a small amount of time separation had a significant effect on production speed.

Recently Kroll et al. (2013b) conducted a case study at Infosys Technologies to examine the feasibility and outcomes of FTS. The study presented software practices and solutions performed to overcome challenges found to develop a software application in the FTS approach, discussing issues and lessons learned.

Simulation and performance evaluation of different FTS projects configurations can be an alternative for feasibility studies of FTS practices. One aspect proposed by Carmel et al. (2009) is the usage of mathematical models for FTS performance evaluation. An approach available to achieve it is the use of stochastic modeling.

1.2. Stochastic automata networks (SAN)

Structured formalisms have been applied and used during years increasing the abstraction level and offering another modeling alternative instead of traditional Markov Chains formalism (Brenner et al., 2005). There are different analytical methods used to obtain the performance indices of stochastic models. The most used are based on Markov Chains (Stewart, 1994). The Markov Chains formalism describes how a system works based on a group of possible states as well as the transitions between those states, following rates defined by exponential rules (Stewart, 1994).

During the modeling phase, certain abstraction level must be considered, because the model does not contain all system characteristics. It only contains relevant information to be modeled (Brenner et al., 2003; Fernandes et al., 1998; Raj, 1991). Characteristics to be modeled should be careful selected in order to provide simulation results that represent an execution of the system which has been simulated. According to Raffo and Setamanit (2005): *“Software Process Simulation Models can be used as a platform to combine and synthesize previously developed theories and models, and incorporate a wide range of relevant factors”*.

In this context, recent studies have applied stochastic automata networks in software engineering area to evaluate software development projects (Czekster et al., 2010, 2011; Fernandes et al., 2011). Besides, according to the literature review there are several challenges to simulate projects with teams geographically distributed, such as time, cost and teams composition.

This research focuses on identifying FTS project model requirements and on FTS practices available in the literature. In this study, we reviewed simulation, quantitative studies and models characteristics that can be used to understand FTS scenarios. Finally, we propose a performance evaluation model to explore Follow-The-Sun aspects. Conclusions point out numerical results as well as future works for new models.

2. FTS performance evaluation

Czekster et al. (2010) presented in recent research stochastic models for software development projects, in order to evaluate communication and availability of global software development teams. Following this context, it was created a basic stochastic model for FTS projects evaluation (Czekster et al., 2011) and it still exists opportunity to capture other important aspects of global software development projects using FTS configuration, such as handoff efficiency, teams capacity and others.

A project is a temporary endeavor to generate a product, a service or a result. Projects are done around the world by companies and institutions to achieve several goals. In the software development industry, project management is a common practice and it has been widely used for software development projects. Software development is defined as a knowledge intensive and as a complex activity (Avram, 2007), and software development projects can be done distributed on different sites. According to Prikladnicki et al. (2010): *“As with many other industries today, software development must increasingly adapt to teams whose members work together but are geographically distributed”*.

The number of uncertain variables and scenarios to manage global software development projects is bigger than managing local software development projects. There are several studies in literature related to the complexity of global software development projects: Carmel et al. (2009), Gupta et al. (2009), Prikladnicki et al. (2010), Avram (2007), Gorton and Motwani (1996), Treinen and Miller-Frost (2006), Sooraj and Mohapatra (2008), Raffo and Setamanit (2005) to cite a few.

2.1. Project view

The performance evaluation of projects under a portfolio can contribute to predict portfolio results helping companies and institutions to better drive their investments.

As mentioned on previous sections, FTS projects have several characteristics and challenges. As part of performance evaluation process, projects scenarios must be created. Fig. 1 presents a project view for this work performance evaluation stochastic models. In our conceptual modeling we focus our attention on the following variables:

- **Cost**: Dollar value per hour, to be assigned to each development site;
- **Time**: Project time in number of hours;
- **Quality**: Measured as the amount of work spent on development *versus* the amount of time on rework (bug fix) activities.

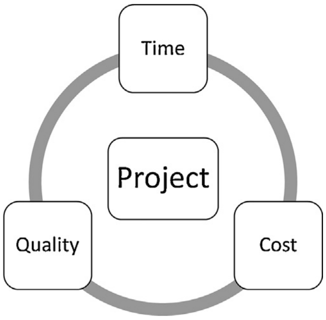

Fig. 1. Project view.

In this context, Follow-The-Sun projects are executed in different geographical locations(sites), transitioning activities from one site to anoter (a process named as handoff). Usually FTS projects have strict rules imposed on their interactions using handoffs between teams at fixed intervals of time.

2.2. Site view

Site is the location where one or more participants of a distributed project will be performing projects activities. Each site has a team of N people executing a variety of projects tasks and each project member can have a different project allocation.

According to Jalote and Jain (2004): *"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-h model"*. In this context a more realistic model also considers time slots that are overlapping. However this paper scope will not consider sites overlap.

Fig. 2 represents the site view. The performance evaluation model will consider the following characteristics as part of scenarios coverage:

- **Country** (team location);
- **Shift size** (number of work hours per day);
- **Team size** (number of people engaged on the project);
- **Average cost** (average dollar rate per hour);
- **Number of sites** that will be part of the scenario;
- **Handoff frequency** time spent on handoff(between sites);
- **Timezone differences** between sites;

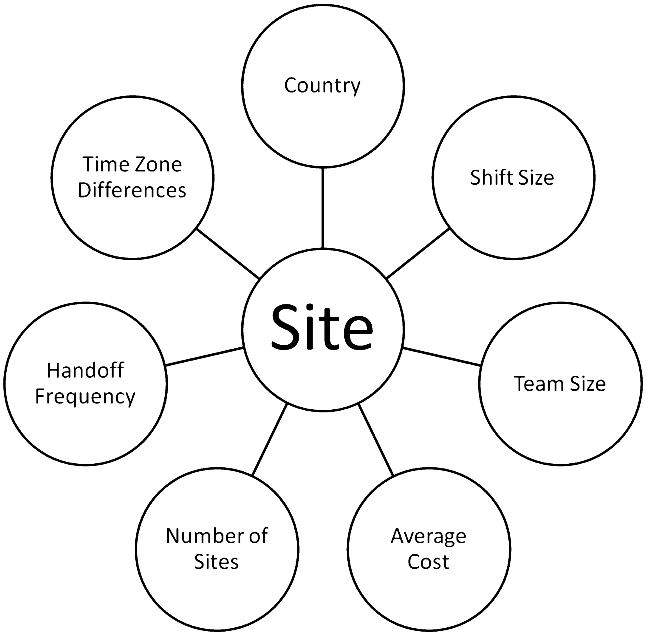

Fig. 2. Site view.

After the description of aspects to be considered at scenarios level, a SAN model should be created. Traditionally, scientific works on software engineering present different applications for mathematical models, *e.g.*, automated software testing processes (Bertolini et al., 2004; Farina et al., 2002), and quantitative evaluation of software development teams also evaluating project risks (Buragga, 2006; Avritzer and Lima, 2009).

This paper demonstrates the usefulness of analytical modeling applied to Follow-The-Sun projects, presenting the solution of a FTS software development life cycle. It focuses on the impact of sites interactions for time, cost and quality. This work does not have a goal to analyze the impact of coordination or cultural diversity, even though these aspects are often relevant. This modeling effort only aims at the impact of different FTS teams configurations in software development projects.

This section presented project aspects that will be used as part of performance evaluation scenarios. Global software development stochastic modeling and FTS stochastic modeling are challenges already discussed by Czekster et al. (2010, 2011) and Urdangarin et al. (2008).

2.3. Research problem

Management of Follow-The-Sun projects involves a high degree of uncertainty which generates risks in terms of cost, quality and project time. To help projects risk mitigation it is possible to use simulation techniques, applying models with values distribution within FTS development cycle. There are different methods to evaluate adherence of a given probability model.

2.4. Research question

The issue of this research is to understand how to qualify the decision making process on *Follow-The-Sun* projects, evaluating variables such as cost, time and quality, because FTS is a research area with aspects to be explored.

Question: *How stochastic automata networks can help FTS projects decision making?*

This question is related to how the use of SAN performance evaluation can help companies and institutions on Follow-The-Sun projects decision making. This study has no hypothesis defined due its exploratory approach.

2.5. Research goal

The goal in this study is to propose a stochastic model and a performance evaluation for Follow-The-Sun projects. In this sense, the following specific goals were defined:

- To perform a study of analytical modeling and performance evaluation for a better understanding of the functioning and application on software development projects;
- To identify Follow-The-Sun aspects that can be used on the analytical model simulation;
- To create a stochastic model to verify through performance evaluation practices that can influence FTS teams performance results;

2.6. Research justification

There are several practical implications on the coordination of FTS projects, specially on planning and execution project phases.

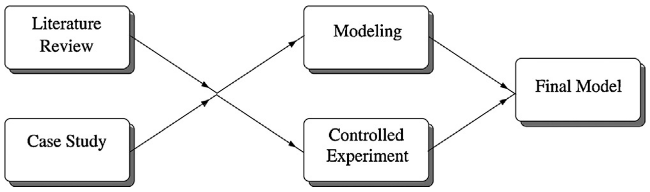

Fig. 3. Research design.

According to Ramasubbu et al. (2011) temporal dispersion reduces the possibilities of communications attributes for design activities *"Temporal dispersion refers to the timezone differences among project members"*. However temporal dispersion may allow distributed groups to accelerate the completion of development tasks using approaches such as *"Follow-The-Sun"*.

The use of Follow-The-Sun as a strategy for software development is not a common practice. It is difficult to find industry cases of software development teams using FTS approach, and according to Nguyen et al. (2008) *"in contrast to work in industry commercial environments, open source projects do not seem to have problems with distributed communication, collaboration and development following the sun"*.

Although FTS related researches exist (Carmel et al. (2009), Gorton and Motwani (1996), Treinen and Miller-Frost (2006), Gupta et al. (2009), Denny et al. (2008), to cite a few), there is an opportunity to research and to simulate Follow-The-Sun software development projects. Simulation and numerical analysis can help to identify project gaps and project opportunities before they start.

3. Research design

The first stage of this research design was composed of a literature review and a case study, the second stage was composed of an controlled experiment and a modeling exercise and the third stage is presented on this paper. Fig. 3 presents this research design.

3.1. Literature review

The literature review aims to find existing studies related to the research goal presented before. This research began with a literature review with focus on: analytical modeling, stochastic automata networks, global software development and Follow-The-Sun.

The literature review used two digital libraries, IEEExplore e Elsevier ScienceDirect. The search was executed between the second semester of 2010 and the first semester of 2011.

Using different research strings 248 studies were listed and after checking titles, keywords and abstracts and also after removing duplicates, 21 studies were selected and reviewed (Table 1). Reviewing the selected studies and based on author names and on their references other studies were found, creating a list of available references available at the bibliography section. Once the literature review was completed, it was performed an analysis to identify practices that could be used as part of this study. Based on this information, it was created a SAN model for FTS projects which will be presented in details at section *The SAN Model*.

3.2. Case study

Fernandes et al. (2011) reported findings about the use of SAN for analytical modeling of software development teams in order to predict their performance in different scenarios. Results were based on a case instance of a multi-site project analyzing the effect of availability and levels of support provided by a centralized management entity. In order to verify the prediction accuracy, the numerical results obtained from the proposed model were compared with actual hours spent in phases of a real project.

Table 1
Research strings.

String	Reviewed
"Follow-The-Sun"	Visser and Solingen (2009), Carmel et al. (2009), van Solingen and Valkema (2010), Yap (2005), Djavanshir (2005), Carmel (1999), Treinen and Miller-Frost (2006), Nicholson and Sahay (2004), Clear and MacDonell (2011)
"Follow-The-Sun" AND *"SAN"*	no results found
"Follow-The-Sun" AND *"stochastic automata networks"*	no results found
"Follow-The-Sun" AND *"simulation"*	Setamanit et al. (2007)
"Follow-The-Sun" AND *"stochastic modeling"*	no results found
"Follow-The-Sun" AND *"stochastic simulation"*	no results found
"global software development" AND *"SAN"*	Taweel and Brereton (2006), Sere et al. (2011)
"global software development" AND *"stochastic automata networks"*	no results found
"global software development" AND *"stochastic modeling"*	Houston et al. (2001)
"global software development" AND *"stochastic simulation"*	Houston et al. (2001)
"global software development" AND *"simulation"*	Setamanit et al. (2007), Sooraj and Mohapatra (2008), Dafoulas et al. (2009), Laurent et al. (2010), Patil et al. (2011), Lehman et al. (1998), Houston et al. (2001)

The contribution of that study Fernandes et al. (2011) was to put into a real project scenario a theoretical modeling effort to describe a complex environment of software development. The obtained numerical results demonstrated an accuracy when compared to actual project outcome. Another contribution was the numerical demonstration of how much of a project success remains on the number of experienced (senior) participants. Not even with a near ideal situation of management central team availability and quality, the problems brought by a large number of junior professionals can be overcome.

3.3. Controlled experiment

Kroll et al. (2012) reported a controlled experiment which had as a goal to verify if an adaptive methodology had more benefits than a prescriptive methodology for a FTS strategy. Findings presented that teams on adaptive approach had 15% less quality, in contrast, those teams had 16% more speed.

Kroll et al. (2012) experiment promoted some insights on how FTS could be used in the software industry. And it has contributed to this paper events parameters settings and transitions design, specially to set the *start handoff (SH) and finish handoff (FH)* event rates probabilities. And it also has provided background to this paper model abstraction with no overlap prediction.

3.4. Modeling

Czekster et al. (2011) presented the mapping of the interaction pattern of development sites under FTS methodology for a SAN model, demonstrating benefits on using SAN formalism for modeling and evaluation of distributed teams; calculating probabilities for availability and project risk factor, such as handoff efficiency; and considering the conceptual framework proposed by Carmel et al. (2009). Conclusions pointed out future works directed to model extensions to capture other characteristics.

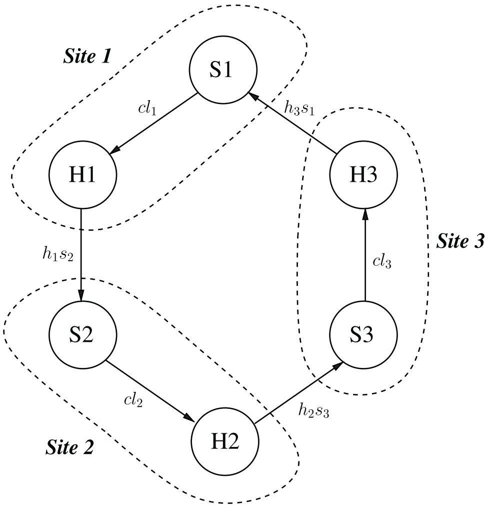

Fig. 4. Clock memory automaton.

4. The SAN model

This section presents a Follow-The-Sun analytical model using the stochastic automata networks (SAN) formalism. The main goal of this model is to instantiate multiple sites (composed by *N* team members) in order to study the handoff behavior between these sites. The main entities represented in this model are described as automata.

4.1. Automata definitions

Automaton is an entity in a SAN model that describes the system behavior by simple primitives such as states and transitions. Fig. 4 graphically shows a three dispersed software development sites interaction in FTS.

In this model (Fig. 4), it is assumed that development work cycle is divided between different sites. Thus, it considers sites self-contained, *i.e.*, sites only work on tasks under their shift assignments. There is a handoff to transition activities between shifts from previous site to the next one. Handoff interactions occur between *Site 1* and *Site 2*, *Site 2* and *Site 3*, *Site 3* and *Site 1*. Team members do not interact with another site that is different from their previous handoff or their next handoff.

The description of the Clock Memory automaton states is available in Table 2.

Beyond the description of the SAN model entities and its relationships it is necessary to assign durations probabilities that every entity rests in a given state named as model parameters and this is paramount within analytical modeling because of its expressiveness (Czekster et al., 2010). For each transition between states there is an event associated and each event has a rate of occurrence. Clock Memory automaton has *six* events associated to it as described in Table 3.

Table 2
Description of the *Clock Memory* automaton states.

State	Description
S1	Site 1 is working
H1	Handoff between Site 1 and Site 2
S2	Site 2 is working
H2	Handoff between Site 2 and Site 3
S3	Site 3 is working
H3	Handoff between Site 3 and Site 1

Table 3
Description of the *Clock memory* automaton events.

Event	Description
cl_1	This is a synchronizing event and it executes the transition from close handoff to offline on Site 1.
h_1s_2	This is a synchronizing event and it executes the transition from Site 1 offline to start handoff on Site 2.
cl_2	This is a synchronizing event and it executes the transition from close handoff to offline on Site 2.
h_2s_3	This is a synchronizing event and it executes the transition from Site 2 offline to start handoff on Site 3.
cl_3	This is a synchronizing event and it executes the transition from close handoff to offline on Site 3.
h_3s_1	This is a synchronizing event and it executes the transition from Site 3 offline to start handoff on Site 1.

We based our model on Carmel's Infosys study (Carmel, 2006) and we cover some development cycle activities such as design, (*i.e.*, requirements gathering), coding and test. In this sense *Site i* automaton, (*i.e.*, the *Site* automaton of index *i*) abstracts the activities that will be performed by the sites instantiated in the model:

- **Handoff**: transition of activities between shifts of different sites.
- **Requirements gathering**: requirements management activities.
- **Development**: development and unit test activities.
- **Test**: all test related to the activities.
- **Bug fix**: rework related to the activities.

In this context, the mapping of development team interaction of site *i* (where *i* is the index of the site) to a SAN model is straightforward, as presented in Fig. 5.

Each site has *N* team members that could be instantiated in order to evaluate different development velocities per site, helping the decision making process to be more efficient in terms of resource management. The abstraction considers software development life cycle represented in each site by *seven* states as described in Table 4.

There are more types of activities that could be executed within a site and could be included in the model abstraction, but for the scope of this research it will be considered *seven* different activities states per site.

Moreover, there is an automaton (Fig. 6) for project flow in order to control the handoff (activities transitions) between sites. For instance, if a site is working on development, Project Memory automaton controls that *development* is the current task, and when the current shift handoff is complete and the next shift starts

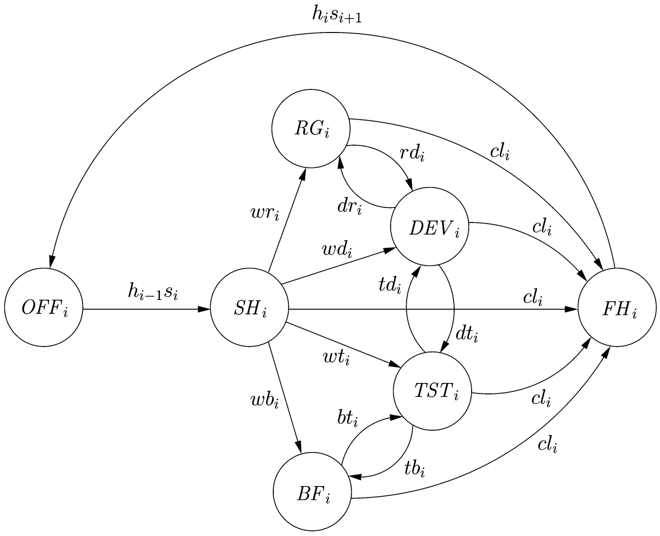

Fig. 5. Automaton related to the tasks of team from Site *i*.

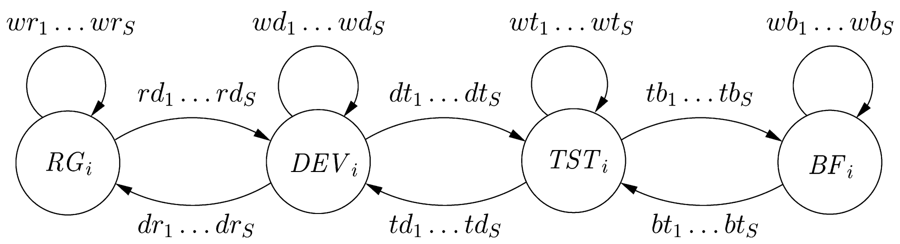

Fig. 6. Project memory automaton.

Table 4
Description of the states related to Site *i*.

State	Description
OFF_i	Site *i* is offline and it does not perform any project activity.
SH_i	Start handoff completing the work transition from previous shift to current shift.
RG_i	Requirements gathering and business system analysis.
DEV_i	Software development and unit testing.
TST_i	Quality assurance testing.
BF_i	Rework and bug fix.
FH_i	Handoff to close activities from current shift to next shift.

handoff, then the next site will follow the activity pointed by Project Memory automaton.

All events in the *site* automaton are classified as synchronizing because they are responsible for synchronising information with the *Project memory* automaton. Each *Site* automaton has *thirteen* events as described in Table 5.

Project members can have different allocations on different projects. In this sense team members are abstracted in this model as an automaton $T_j^{(i)}$ with *two* states representing the availability and unavailability of a given team member, where j indicates the index of the team member ($j = 1 \,..\, N$) from site i.

The description of the states presented in the automata of Fig. 7 are described in Table 6.

It is important to remark that for each instance of a team member automata it is possible to set different events probabilities enabling the evaluation of different team members project allocation as well as different development speed, because according to the number of members instantiated per site it also enables different development speed per site.

Table 7 presents a summary of events from *Team Site i*, previously illustrated in Fig. 7.

Once the parameterization is complete, (*i.e.*, each event rate is assigned), the model can be executed in specialized software tools for numerical solution, *e.g.*, PEPS software tool (Brenner et al., 2007)

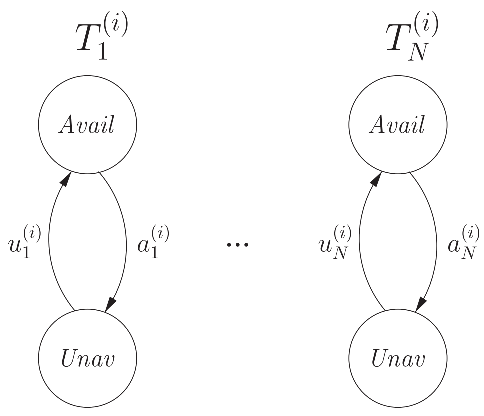

Fig. 7. Team site *i*.

Table 5
Events from site *i*.

Event	Description
$h_i s_{i+1}$	**Start handoff**: this event synchronizes *i-th Site* automaton with *Clock Memory* automaton, coordinating start handoff activities between sites.
$h_{i-1} s_i$	**Finish handoff**: this event synchronizes *i-th Site* automaton with *Clock Memory* automaton, coordinating finish handoff activities between sites.
cl_i	**Closing activities**: when this event occurs the *i-th* Site automaton goes to the state where it needs to finish handoff.
wr_i	**Resume working on requirements gathering**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton, enabling the site to work on requirements gathering activities once the start handoff was completed.
wd_i	**Resume working on development**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton, enabling the site to work on development activities once the start handoff was completed.
wt_i	**Resume working on testing**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton, enabling the site to work on testing activities once the start handoff was completed.
wb_i	**Resume rework**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton, enabling the site to work on bug fix activities once the start handoff was completed.
rd_i	**Move to development**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from requirements gathering to software development.
dr_i	**Move from development**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from software development to requirements gathering.
dt_i	**Move to testing**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from software development to testing.
td_i	**Move from testing**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from testing to software development.
tb_i	**Move to bug fix**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from testing to bug fix.
bt_i	**Move from bug fix**: this event synchronizes *i-th Site* automaton with *Project Memory* automaton by the transition of activities from bug fix to testing.

Table 6
Description of the *Team site's* states.

State	Description
Avail	This state indicates that a participant is available to work.
Unav	The participant is unavailable and cannot perform any project task.

Table 7
Description of the *Team site's* events.

Event	Description
$a_j^{(i)}$	This is a local event and it executes the transition of the j-th team member from *available* to *unavailable*.
$u_j^{(i)}$	This is a local event and it executes the transition of the j-th team member from *unavailable* to *available*.

or SAN Lite-Solver (Sales, 2012), in order to extract performance indices and measures (Czekster et al., 2010).

4.2. Model options and constraints

This work presents a three sites model and a four sites model as options because it is the closest way to cover 24 h of a workday. Some authors have found indications that when the number of sites in a daily cycle increases, on average the overall working speed of sites also increases (van Solingen and Valkema, 2010), however they have mentioned that: *"The maximum number of sites in a daily cycle is finite, due to the 24 h in a day. More sites provide more working capacity, however also require more overhead and increase the likelihood of mistakes"*.

When instantiating the stochastic model it needs to consider the number of resources working on each site for performance evaluation numerical results. Sangwan et al. (2006) suggested a rule of thumb where teams should be no larger than 10 people because *"jumping into a large distributed project without having past experience is not recommended"*.

The proposed model does not predict shifts overlap. However literature studies (Treinen and Miller-Frost, 2006; Visser and Solingen, 2009) point out cases where it is recommended a *sticky handoff*, *i.e.*, intense interactions are more favorable than a *clean handoff* (*drop-and-go* approach). The average event rates used are based on the following literature references: Javed et al. (2004), Basili et al. (1995), Carmel et al. (2009).

- Javed et al. (2004), a work based on four industry projects.
  Average RG time: 23%
  Average DEV time (Design + Coding): 42%
  Average TST time: 24%
  Average time others: 11%
- Basili et al. (1995), a work based on a lab experience.
  Average RG time: 12%
  Average DEV time (Design + Coding): 42%
  Average TST time: 20%
  Average time others: 26%
- Carmel et al. (2009), a foundation for understanding FTS.
  Average RG time: 10%
  Average DEV time (Design + Coding): 55%
  Average TST time: 25%
  Average time others: 10%

Regarding quality it will account for each site probabilities of state *Bug fix (BF)* divided by the probability of state *Development (DEV)*. Regarding time it will account for each site the sum of probabilities of all states. Regarding cost it will apply an average dollar rate hour value per site multiplied by the number of project hours. It is clear that there are more aspects to be discussed about time, cost and quality, but they will not considered in this study.

Based on the sense that this research has some exploratory aspects, *two model instances examples were created to calibrate and to evaluate the model due the lack of practical FTS cases available in literature*. An average dollar value per hour was set based on average values from *payscale.com*[3] to exercise the model due the lack of literature with detailed information about real costs. The cost calculation is given by the sum of the average time spent on each model state in each site multiplied by the average dollar cost *per* hour for a given site.

This research uses heuristics to split different amount of hours on the scenarios creation. Those heuristics basically divide a given number of hours by the activities performed by each site, such as: requirements gathering, development, testing, bug fix and handoff. Section 5 provides details about heuristics used and the scenarios creation.

Besides previous mentioned references, the model was also calibrated with the following information from literature: Patil et al. (2011), Raffo and Setamanit (2005), Dafoulas et al. (2009), O'Leary and Cummings (2007), Sooraj and Mohapatra (2008), Treinen and Miller-Frost (2006), Carmel (1999), Czekster et al. (2011), Jalote and Jain (2004). Next section presents the results and demonstrates the model instantiation and the performance evaluation.

5. Performance evaluation results

There are different practices to measure software development projects and it is difficult to have an agreement about what should be measured and how to analyze the results. In this sense, the SAN model and performance evaluation results can be used as a tool to enhance the decision making on Follow-The-Sun projects.

Software metrics can help project management planning. In this phase it is possible to identify the amount of required effort and an estimated cost to complete a certain project. As mentioned in previous sections, this model focus on scenarios to measure time, cost and quality for Follow-The-Sun software development projects.

5.1. Running the model

To illustrate and to exercise the functionality and the usefulness of the Follow-The-Sun model, we created two example projects. Due to the fact of been using stochastic automata networks for performance evaluation, the values generated by the model execution are by nature stochastically precise.

5.2. Samples settings

These examples are abstracting the model defined phases: *Requirements gathering (RG)*, *Design and coding (DEV)*, *Testing (TST)* and *Bug fix / Rework (BF)*. Besides this model considers *Start Handoff from previous shift (SH)*, *Finish handoff to the next shift (FH)* and *When the site is offline (OFF)*. Examples are based on *UTC+12-12* and the average time used on each model event rate is based on the following literature: Javed et al. (2004), Basili et al. (1995), Carmel et al. (2009).

5.2.1. Sample 1

This project has three development sites: *New Zealand (UTC+12), Russia (UTC+4) and Bolivia (UTC-4)* with the following respective average hour dollar rate: 26.62, 9.32 and 11.46. Fig. 8 presents this sample country distribution.

It considers a day of 24 h of work divided by three sites. Assumptions and basis for calculations are the following:

- Workdays per week: 5
- Work hours per day: 8
- Non project work: 8 h per month. These hours are spent in regular activities, time away and staff meetings.
- Project activities includes all tasks related to software development including coordination time and project meetings.

Each sample scenario was created to illustrate the use of model instances. This example instance used the estimates described in Table 8.

---

[3] Annual number of hours based on 40 working hours per week = 1920 h per year. Average annual salary converted from local currency to US dollar. Russia DEV US $ 17,910.57 per year: US $ 9.32 per h. USA DEV US $ 67,685 per year: US $ 35.95 per h. UK DEV US $ 50,272.32 per year: US $ 26.18 per h. New Zealand DEV US $ 51,124.95 per year: US $ 26.62 per h. Due to the lack of software developer salary information in Bolivia, it was made an analogy to Brazil DEV US $ 22,007 per year: US $ 11.46 per h.

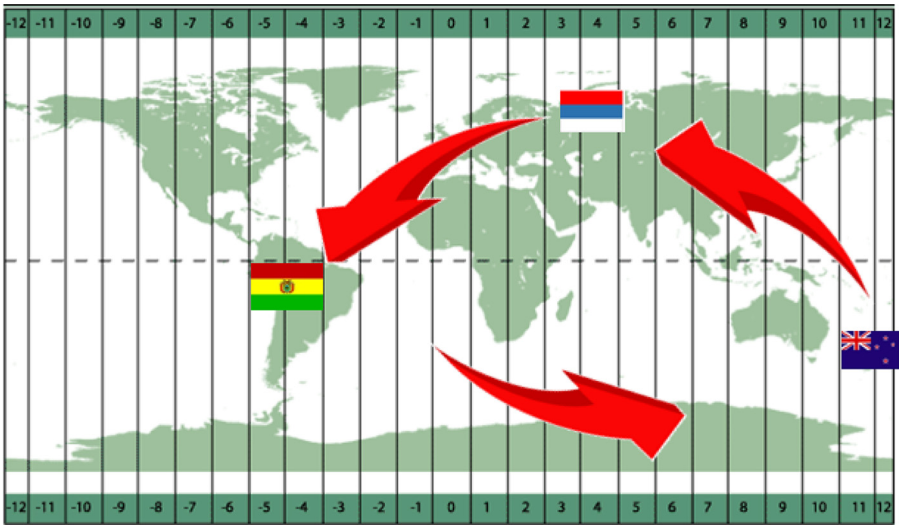

Fig. 8. Country distribution – sample 1.

Table 8
Project experts estimation for sample 1.

State	Hours	Dollar cost avg.
Requirements gathering	1000	15,800.00
Development	2500	39,500.00
Testing	1200	18,960.00
Bug fix and rework	600	9480.00
Operational handoffs	150	2370.00
Total	5450	86,110.00

Table 9
Parameters (in minutes) – sample 1.

State	New Zealand	Russia	Bolivia
OFF	960.00	960.00	960.00
SH	8.00	3.82	8.00
RG	85.00	88.00	91.22
DEV	220.55	230.00	210.00
TST	105.00	105.06	107.00
BF	53.88	52.12	52.53
FH	7.57	1.00	11.25
Total	1440.00	1440.00	1440.00

For each model instance, heuristics were build to create the estimation scenario. In this sense this model instance estimates are based on the following heuristics:

- Requirements gathering = 18.3% of a workday.
- Development = 45.84% of a workday.
- Testing = 22.11% of a workday.
- Bug fix = 11% of a workday.
- Handoff = 2.75% of a workday.

Estimates of performance evaluation - sample 1

- Expected duration in months: 4
- Average number of hours per month: 1362.50 (5450 h ÷ 4 months)
- Average number of hours per day: 61.93 (1362.50 h ÷ 22 workdays)
- Average number of hours per site per day: 20.64 (61.93 h ÷ 3 sites)
- Average number of resources per site: 2.58 (20.64 h ÷ 8 h)
- Average monthly cost: US $ 21,527.50 (US $ 86,110.00 ÷ 4 months)
- Quality: 23.99% = (6.81 BF hours daily (11%) ÷ 28.39 DEV hours daily (45.84%)) × 100

According to the sample project estimates, the main idea is to assign durations to every state in the model, *i.e.*, frequencies at every connection among states. Table 9 demonstrates this sample parameterization, *i.e.*, average time probability in minutes are based on a day of 24 h, (*i.e.*, 1440 min), considering each site working 8 h per day.

The parameterization completes the model, which can now be subjected to specialized tools for numerical solution, in this case, SAN Lite-Solver (Sales, 2012), in order to extract performance indices and measures. After the model execution the resultant probabilities rates are multiplied back by the base parameters value for a day of work. In this sense each probability output is multiplied by 1440 min representing a day of 24 h of work. A model solved means that it was found a steady-state solution.

Performance evaluation results - sample 1

- Average number of resources per site: 3.82 (4 resources working on average 95.5% of time on project activities[4])
- Average number of hours per site per day: 30.56 (8 h × 3.82 resources)
- Average number of hours per day: 91.64 (30.56 h per site per day × 3 sites)
- Average number of hours per month: 2015.99 (91.64 h per day × 22 days per month)
- Duration in months: 2.70 (5450 h ÷ 2015.99 h per month)
- Average monthly cost: US $ 31,878.22 = (813.77 + 284.91 + 350.33) × 22 days per month
  - New Zealand daily cost: US $ 813.77 (30.56 h × 26.62 dollars)
  - Russia daily cost: US $ 284.91 (30.56 h × 9.32 dollars)
  - Bolivia daily cost: US $ 350.33 (30.56 h × 11.46 dollars)
- Quality: 23.99% = (10.09 BF hours daily (3.74% + 3.63% + 3.64%) ÷ 42.06 DEV hours daily (15.31% + 15.92% + 14.63%)) × 100

---

[4] This percentage is the sum of the probabilities of automata *Team Site* remain in state *Avail*.

Table 10
State probabilities – sample 1.

Site	State	Time (min)	Probability
New Zealand	OFF	960.05	66.67%
	SH	7.92	0.55%
	RG	85.25	5.92%
	DEV	220.32	15.30%
	TST	104.98	7.29%
	BF	53.86	3.74%
	FH	7.63	0.53%
Russia	OFF	960.05	66.67%
	SH	3.74	0.26%
	RG	87.84	6.10%
	DEV	229.82	15.96%
	TST	105.26	7.31%
	BF	52.27	3.63%
	FH	1.01	0.07%
Bolivia	OFF	960.05	66.67%
	SH	7.92	0.55%
	RG	91.15	6.33%
	DEV	210.67	14.63%
	TST	106.85	7.42%
	BF	52.42	3.64%
	FH	10.94	0.76%

The time associated to state *OFF* means that the site will remain 960.05 min (around 16 h) on offline state. After the model execution the probability of a site stays on this state is 66.67% (16 h from 24 h). The probabilities of other states also resulted on some variation based on the average estimated time due to the model events synchronization and due to the model dynamics. However, these are small variations, confirming that the model parameterization is correct. For instance, site *New Zealand* had an estimation of 8 daily minutes to start handoff (SH) and the model execution resulted in 7.92 daily minutes in this state.

Table 10 demonstrates this scenario sample average results for a day of work obtained from the usage of SAN Lite-Solver software tool (Sales, 2012).

This first sample performance evaluation generated the following conclusions:

- With four team members this project could be executed using 67% (2.7 ÷ 4 months) of the estimated project time.
- The monthly budget would increase 10,350.72 dollars (31,878.22 – 21,527.50).
- The estimated quality result and the performance quality result match based on the model parameters set.

5.2.2. *Sample 2*

This sample explores the increase of sites working per day. In this sample, we have now four development sites: *New Zealand* (UTC+12), *Russia* (UTC+6), *UK* (UTC 0) and *USA* (UTC-6) with the following respective average hour dollar rate: 26.62, 9.32, 26.18 and 35.95 (Fig. 9).

We are considering a day of 24 h of work divided by four sites. This sample assumes that four team members are working on each shift. Assumptions and basis for calculations are the following:

- Workdays per week: 5
- Work hours per day: 6
- Non project work: 6 h per month. These hours are spent in regular activities, time away, staff meetings.
- Project activities include all tasks related to software development including coordination time and project meetings.

The scenario estimates applied to the second sample can be seen in Table 11.

Table 11
Project experts estimation for sample 2.

State	Hours	Dollar cost avg.
Requirements gathering	690	16,917.08
Development	3430	84,095.03
Testing	690	16,917.08
Bug fix and rework	600	14,710.50
Operational handoffs	670	16,426.73
Total	6080	149,066.42

For each model instance, heuristics were build according to the literature in order to create the estimation scenario. In this sense this model instance estimates are based on the following heuristics:

- Requirements gathering = 11.35% of a workday.
- Development = 56.41% of a workday.
- Testing = 11.35% of a workday.
- Bug fix = 9.87% of a workday.
- Handoff = 11.02% of a workday.

Estimates of performance evaluation - sample 2

- Expected duration in months: 5
- Average number of hours per month: 1216 (6080 h ÷ 5 months)
- Average number of hours per day: 55.27 (1216 h ÷ 22 workdays)
- Average number of hours per site per day: 13.82 (55.27 h ÷ 4 sites)
- Average number of resources per site: 2.30 (13.82 h ÷ 6 h)
- Average monthly cost: US $ 29,813.28 (US $ 149,066.42 ÷ 5 months)
- Quality: 17.51% = (5.46 BF hours daily (9.87%) ÷ 31.18 DEV hours daily (56.41%)) × 100

Table 12 demonstrates this sample parameterization, *i.e.*, average time probability in minutes are based on a day of 24 h, (*i.e.*, 1440 min), considering each site working 6 h per day. After the model execution the resultant probabilities rates are multiplied back by the base parameters value for a day of work. In this sense each probability output is multiplied by 1440 min representing a day of 24 h of work. It is important to remark that this second sample is based on four sites working six hours per day each, covering 24 h.

Performance evaluation results - sample 2

- Average number of resources per site: 3.82 (4 resources working on average 95.5% of time on project activities[5])
- Average number of hours per site per day: 22.92 (6 h × 3.82 resources)
- Average number of hours per day: 91.68 (22.92 h per site per day × 4 sites)
- Average number of hours per month: 2016.96 (91.68 daily hours × 22 days per month)
- Duration in months: 3.01 (6080 h ÷ 2016.96)
- Average monthly cost: US $ 49,450.72 = (610.13 + 213.61 + 600.05 + 823.97) × 22 days per month
  - New Zealand daily cost: US $ 610.13 (22.92 h × 26.62 dollars)
  - Russia daily cost: US $ 213.61 (22.92 h × 9.32 dollars)
  - UK daily cost: US $ 600.05 (22.92 h × 26.18 dollars)
  - USA daily cost: US $ 823.97 (22.92 h × 35.95 dollars)

---

[5] This percentage is the sum of the probabilities of automata *Team site* remain in state *Avail*.

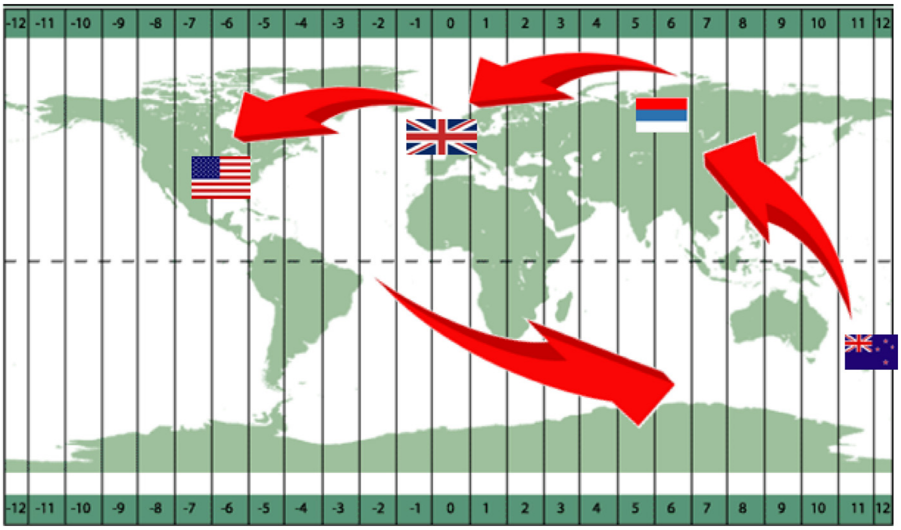

Fig. 9. Country distribution – sample 2.

Table 12
Parameters (in minutes) – sample 2.

State	New Zealand	Russia	UK	USA
OFF	1080.00	1080.00	1080.00	1080.00
SH	10.00	10.00	10.00	20.00
RG	50.00	80.00	20.00	100.00
DEV	100.00	170.00	230.00	150.00
TST	160.00	10.00	50.00	30.00
BF	20.00	80.00	40.00	45.00
FH	20.00	10.00	10.00	15.00
Total	1440.00	1440.00	1440.00	1440.00

- Quality: 28.71% = (11.97 BF hours daily (1.36% + 5.65% + 2.85% + 3.20%) ÷ 41.70 DEV hours daily (7.19% + 11.79% + 15.88% + 10.62%)) × 100

Table 13 demonstrates this scenario execution average results for a day of work obtained from the usage of SAN Lite-Solver software tool.

The rate for the OFF model state means that the site will be 1080 min (18 h) on offline state. After the model execution the probability of a site stays on this state is 75.00% (18 h from 24 h). This sample model demonstrated that the (OFF) state is 75% for each site which is tied to eighteen non working hours in a day of 24 h. Probabilities of other states also resulted on some variation based on the average estimated time due to the model events synchronization and due to the model dynamics. However, these are small variations, confirming that the model parameterization is correct. For example, site UK had an estimation of 40 daily minutes on bug fix (BF) and the model execution found 41.04 daily minutes on this state.

Regarding those results, note that the increasing of the number of sites can reduce project cycle and project cost. An interesting fact is that using the same average number of resources on a three sites configuration model and the average number of working hours per day is really close to the average number in a four sites configuration model results, because even with four resources working in four different sites the working capacity of each site will be six hours a day instead of eight hours a day.

The second example provided an interesting conclusion: the more sites are working, the more costs compositions are available. It is possible to exercise different costs compositions. Surely

Table 13
State probabilities – sample 2.

Site	State	Time (min	Probability
New Zealand	OFF	1080.00	75.00%
	SH	9.79	0.68%
	RG	53.28	3.70%
	DEV	103.54	7.19%
	TST	153.79	10.68%
	BF	19.58	1.36%
	FH	20.02	1.39%
Russia	OFF	1080.00	75.00%
	SH	9.79	0.68%
	RG	78.48	5.45%
	DEV	169.78	11.79%
	TST	10.66	0.74%
	BF	81.36	5.65%
	FH	9.93	0.69%
UK	OFF	1080.00	75.00%
	SH	9.79	0.68%
	RG	20.74	1.44%
	DEV	228.67	15.88%
	TST	49.82	3.46%
	BF	41.04	2.85%
	FH	9.94	0.69%
USA	OFF	1080.00	75.00%
	SH	19.00	1.32%
	RG	96.19	6.68%
	DEV	152.93	10.62%
	TST	30.82	2.14%
	BF	46.08	3.20%
	FH	14.98	1.04%

there are studies that point a major effort on coordination (Carmel et al., 2009; van Solingen and Valkema, 2010), however it is possible to discuss that the use of more sites and different costs composition possibilities may compensate an increasing cost on project coordination.

Based on the literature review and on these two samples model instances presented, it is possible to observe that the number of sites makes the difference on project execution time as well as on project costs composition. The proposed model allows to exercise different projects compositions depending on the number of team members allocated on each site and on the average resource cost from a given site.

Through provided samples it was observed indicatives that the use of the SAN formalism can help the decision making process of Follow-The-Sun projects. It is important to reinforce that models can be easily changed to capture other dimensions of FTS projects which, however, were not part of this work scope.

The examples previously provided are hypothetical samples to illustrate how the FTS model can be used to enhance projects planning. They were calibrated with literature information previously presented in Section 4.2. Since it cannot be created hypotheses based on available data, these results cannot be classified as findings but they are indicatives of how SAN can be used to improve Follow-The-Sun decision making process.

6. Discussion

The literature review pointed that FTS shares many challenges as a GSD approach. However, many FTS variants are available, as the possibility of sites overlap, techniques to cope communication and coordination challenges, to cite a few. The results provided in previous section are hypothetical and they intend to illustrate the functionality and usefulness of the FTS model. Despite the fact that our results are based on simplified models from the literature (Patil et al., 2011; Raffo and Setamanit, 2005; Dafoulas et al., 2009; O'Leary and Cummings, 2007; Sooraj and Mohapatra, 2008; Treinen and Miller-Frost, 2006; Carmel, 1999; Czekster et al., 2011; Jalote and Jain, 2004), it is quite clear that the success of FTS-based projects relies on different factors and this research aims to help FTS projects decision making. Specifically, we are interested in how a stochastic modeling tool, the Stochastic Automata Networks formalism, can be used as a prediction tool to estimate the costs involved in choosing among FTS variants.

On Sample 1 we performed an evaluation of a three sites scenario where we were able to capture decrease between project duration estimates and project execution from 4 to 2.7 months. This could be achieved increasing the monthly cost in US $ 10,350.72, and having 4 resources working on average 95.5% of the time on project activities. The numerical results of our simulations show that using the same group of sites it is possible to reduce the project execution time, which is probably the more important claim for FTS projects.

On Sample 2 we performed an evaluation of a scenario with four sites configuration, where the total cost could be US $ 148.8K (3.01 × 49,450.72) instead of US $ 149K, representing a small difference of less than 1% in cost between the expert estimation and the simulation results. Such small difference indicates at the same time the reliability of our simulation results, as well as the dependency of the overall project configuration. Specifically, the fact of considering the same country-based teams results in similar behavior, and only a different choice of groups' sites would represent a real choice of the best scenario to choose.

In terms of quality, based on our model abstraction we have not found any significant difference between three and four sites samples. It is important to observe that our model does not focus on the quality of the product, since our abstraction level remains on the number of working hours, and on its impact of project duration and cost. Therefore, we believe that an eventual difference of quality would be expected changing from Sample 1 to Sample 2, but it is not the goal of our model to estimate such difference. Nevertheless, our model results remains useful to Follow-The-Sun projects decision making. For instance, giving similar costs, it seems to be natural to choose a three sites FTS configuration over a four sites FTS configuration.

This information can also help to improve project performance evaluation in order to create different sites allocation strategies. For example, it is possible to search for an optimal cost composition considering the hiring cost of professional in each possible country, creating flexibility on coordination cost management options.

In our model we were able to capture important aspects such as handoff and development stages, (*e.g.*, requirements gathering, development and testing). An interesting conclusion when comparing Samples 1 and 2 is that the inclusion of more sites is easy to evaluate, and consequently the decision making process can pay more attention to other aspects not included in the model. Specifically, our model allows to exercise different costs compositions as a supporting tool do define Follow-The-Sun sites configuration. There are studies pointing a major effort on coordination (Carmel et al., 2009; van Solingen and Valkema, 2010), but our approach allows to rely on numerical predictions of the impact of real life configurations. Such powerful analysis may show that some different costs compositions may compensate an increasing cost on project coordination. 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-h model (Jalote and Jain, 2004). This model could be extended to include other practices such as sites overlap to evaluate different strategies combination in order to find the best alternative according to the company/institution goals.

7. Final remarks

In this paper, we presented models using the Stochastic Automata Networks (SAN) formalism and a through result analysis of these models. We observed indicatives that the use of SAN can help the decision making process of Follow-The-Sun (FTS) projects. Evidently, the stochastic modeling presented in this study is not a complete description of all FTS aspects, but it already provides insights on the cost of some decisions to specific FTS project configurations.

Regarding the modeling process, within the abstraction level to create a SAN model, there was no previous data categorization prepared. Therefore, we had to define model parameters values by ourselves, but once the model was instantiated generating the output, it was possible to categorize the information to be evaluated. For any project decision maker, more important than the tool itself is the degree of confidence generated by the output results. It is important to stress that this work focus was not on measuring the quality level of the development product, instead it focuses on the degree of credibility from the results generated by models executions.

A challenge found during this research was the lack of a methodology to simulate software engineering projects using SAN models. There is an opportunity as future work to create a methodology for such demand in order to elaborate a robust method to apply SAN on software engineering problems.

From this point of view, this work presented how SAN can be used to enhance decision making on FTS projects mostly by time and cost prediction. It was also shown that the increasing number of sites may increase the coordination costs, but it also opens different costs compositions that may compensate the coordination overhead. It is important to remark that the model herein presented could be extended/modified and applied on analysis of FTS projects using different levels of abstraction.

FTS is not a common industry practice yet, and despite FTS being the focus of our work, the model proposed here could be applied to other methodologies where timezone difference is an important aspect. In such way, the main contribution of this study is focused on the modeling process of FTS projects, instead of the practical application on the industry.

Jorgensen and Shepperd (2007) conducted a study that reviewed software cost estimation papers published in journals and tried to support other software cost estimation researchers through a library of estimation classified papers. However it has not provided an average cost value per hour and per country. It has been difficult to find public material about software development cost distributions in different countries and the study presented here used *payscale.com* average cost rates for cost simulation and analysis.

An important limitation to the model herein presented is the lack of shifts overlap. However, it can be considered a small overlap for synchronous handoff model, declaring teams overlap at the beginning/end of a shift to allow synchronous coordination. Naturally, some extensions to the current FTS model can be studied to include such features, but this future work demands a deeper analysis of practical aspects of the FTS recommendations for shifts overlap.

Additionally, this model does not consider how different cultures may impact on the performance, specially in sight of tasks complexity and workload burden. It assumes on the abstraction that the workload is equally distributed among the members. Also, it does not consider different levels of complexity for each task.

Possible model extensions and future works can be done creating a new modeling exercise aiming the generation of an evaluation toolkit to present a mathematical framework for FTS projects. This conceptual modeling workflow could instantiate different models depending on the configuration for handoff and the desired analysis (or evaluation) to be done.

This work focus on one instance of timezone gradation, the FTS one. However, Espinosa et al. (2007) conducted a study about different timezone gradations. It is possible to create a stochastic model to evaluate different timezone gradations where FTS would be just one instance of this model as an extreme timezone difference case. As a future work users can extend the existing model to numerically analyze other FTS projects dimensions as well as the impact of timezone gradations on GSD projects.

For all those reasons, we believe that this paper contribution resides more on the proposal of a different way of dealing with project estimation, rather than solely on the prediction of FTS project behavior. Despite of that, our initial numerical results may be already helpful to FTS practitioners since the simulation results were fairly accurate in comparison with the project expert estimations.

References

Avram, G., 2007. Knowledge work practices in global software development. Electron. J. Knowl. Manage. 5 (4), 347–356.

Avritzer, A., Lima, A., 2009. 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, pp. 341–346.

Basili, V., Zelkowitz, M., McGarry, F., Page, J., Waligora, S., Pajerski, R., 1995. SEL's software process improvement program. IEEE Softw. 12 (6), 83–87.

Bertolini, C., Farina, A.G., Fernandes, P., Oliveira, F.M., 2004. 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, pp. 251–260.

Brenner, L., Fernandes, P., Plateau, B., Sbeity, I., 2007. PEPS 2007 - Stochastic Automata Networks Software Tool. In: Proceedings of the 4th International Conference on Quantitative Evaluation of Systems (QEST 2007), IEEE Computer Society, Edinburgh, Scotland, pp. 163–164.

Brenner, L., Fernandes, P., Sales, A., 2003. MQNA - Markovian queueing networks analyser. In: 11th IEEE/ACM International Symposium on Modelling, Analysis and Simulation on Computer and Telecommunication Systems (MASCOTS'03), IEEE Computer Society, Orlando, FL, USA, pp. 194–199.

Brenner, L., Fernandes, P., Sales, A., 2005. The need for and the advantages of generalized tensor algebra for Kronecker structured representations. Int. J. Simul. Syst. Sci. Technol. 6 (3–4), 52–60.

Buragga, K.A., 2006. An analytical model for requirements activities and software product quality. J. Comput. Methods Sci. Eng. 6, 205–215.

Carmel, E., 1999. Global Software Teams: Collaborating Across Borders and Time Zones. Prentice Hall PTR, Upper Saddle River, NJ, USA.

Carmel, E., 2006. Building your information systems from the other side of the world: how Infosys manages time zone differences. MIS Q. Exec. 5 (1).

Carmel, E., Dubinsky, Y., Espinosa, J., 2009. Follow The Sun software development: new perspectives, conceptual foundation, and exploratory field study. In: 42nd Hawaii International Conference on System Sciences (HICSS '09), pp. 1–9.

Carmel, E., Dubinsky, Y., Espinosa, J., 2010. Follow The Sun workflow in global software development. J. Manage. Inf. Syst. 27 (1), 17–38.

Carmel, E., Espinosa, J., 2011. I'm Working While They're Sleeping: Time Zone Separation Challenges and Solutions. Nedder Stream Press, USA.

Clear, T., MacDonell, S.G., 2011. Understanding technology use in global virtual teams: research methodologies and methods. Inf. Softw. Technol. 53 (9), 994–1011.

Cummings, J., Espinosa, J., Pickering, C., 2009. Crossing spatial and temporal boundaries in globally distributed projects: a relational model of coordination delay. Inf. Syst. Res. 20 (3), 420–439.

Czekster, R., Fernandes, P., Prikladnicki, R., Sales, A., Santos, A., Webber, T., 2011. Follow-The-Sun methodology in a stochastic modeling perspective. In: 6th IEEE International Conference on Global Software Engineering (ICGSE): Methods and Tools for Project/Architecture/Risk Management in Globally Distributed Software Development Projects (PARIS). Helsinki, Finland, pp. 54–59.

Czekster, R., Fernandes, P., Sales, A., Webber, T., 2010. Analytical modeling of software development teams in globally distributed projects. In: International Conference on Global Software Engineering (ICGSE'10). IEEE Computer Society, Princeton, NJ, USA, pp. 287–296.

Dafoulas, G., Swigger, K., Brazile, R., Alpaslan, F., Cabrera, V., Serce, F., 2009. Global teams: futuristic models of collaborative work for today's software development industry. In: 42nd Hawaii International Conference on System Sciences (HICSS '09), pp. 1–10.

Denny, N., Crk, I., Sheshu, R., 2008. Agile software processes for the 24-h knowledge factory environment. J. Inf. Technol. Res. 1 (1), 57–71.

Djavanshir, G., 2005. Surveying the risks and benefits of it outsourcing. IT Prof. 7 (6), 32–37.

Espinosa, J., Cummings, J., Wilson, J., Pearce, B., 2003. Team boundary issues across multiple global firms. J. Manage. Inf. Syst. 19 (4), 157–190.

Espinosa, J., Nan, N., Carmel, E., 2007. Do gradations of time zone separation make a difference in performance? A first laboratory study. In: Proceedings of the International Conference on Global Software Engineering (ICGSE '07), IEEE Computer Society, Munich, Germany, pp. 12–22.

Faraj, S., Sproull, L., 2000. Coordinating expertise in software development teams. Manage. Sci. 46 (12), 1554–1568.

Farina, A.G., Fernandes, P., Oliveira, F.M., 2002. 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, pp. 401–407.

Fernandes, P., Plateau, B., Stewart, W., 1998. Efficient descriptor-vector multiplications in stochastic automata networks. J. ACM 45, 381–414.

Fernandes, P., Sales, A., Santos, A., Webber, T., 2011. Performance evaluation of software development teams: a practical case study. Electron. Notes Theor. Comput. Sci. 275, 73–92.

Gorton, I., Motwani, S., 1996. Issues in co-operative software engineering using globally distributed teams. Inf. Softw. Technol. 38 (10), 647–655.

Gupta, A., Mattarelli, E., Seshasai, S., Broschak, J., 2009. Use of collaborative technologies and knowledge sharing in co-located and distributed teams: towards the 24-h knowledge factory. J. Strateg. Inf. Syst. 18, 147–161.

Houston, D., Mackulak, G., Collofello, J., 2001. Stochastic simulation of risk factor potential effects for software development risk management. J. Syst. Softw. 59 (3), 247–257.

Jalote, P., Jain, G., 2004. Assigning tasks in a 24-h software development model. In: Asia-Pacific Software Engineering Conference, pp. 309–315.

Javed, T., Maqsood, M., Durrani, Q., 2004. A study to investigate the impact of requirements instability on software defects. ACM SIGSOFT Softw. Eng. Notes 29 (3), 1–7.

Jorgensen, M., Shepperd, M., 2007. A systematic review of software development cost estimation studies. IEEE Trans. Softw. Eng. 33 (1), 33–53.

Kroll, J., Hashmi, S., Richardson, I., Audy, J., Aug 2013. A systematic literature review of best practices and challenges in follow-the-sun software development. In: IEEE 8th International Conference on Global Software Engineering Workshops (ICGSEW), pp. 18–23.

Kroll, J., Prikladnicki, R., Audy, J.L.N., Carmel, E., Fernandez, J., 2013b. A feasibility study of follow-the-sun software development for gsd projects. In: International Conference on Software Engineering (SEKE), Boston, USA, pp. 196–199.

Kroll, J., Santos, A., Prikladnicki, R., Hess, E., Glanzner, R., Sales, A., Audy, J., Fernandes, P., 2012. Follow-the-Sun software development: a controlled experiment to evaluate the benefits of adaptive and prescriptive approaches. In: Proceedings of the 24th International Conference on Software Engineering and Knowledge Engineering (SEKE 2012), San Francisco Bay, USA, pp. 551–556.

Laurent, P., Mader, P., Cleland-Huang, J., Steele, A., 2010. A taxonomy and visual notation for modeling globally distributed requirements engineering projects. In: Proceedings of the 2010 5th IEEE International Conference on Global Software Engineering. ICGSE '10, IEEE Computer Society, Washington, DC, USA, pp. 35–44.

Lehman, M., Lehman, M., Ramil, J., 1998. The impact of feedback in the global software process. J. Syst. Softw. 46, 123–134.

Nguyen, T., Wolf, T., Damian, D., 2008. Global software development and delay: Does distance still matter? In: IEEE International Conference on Global Software Engineering, ICGSE 2008, pp. 45–54.

Nicholson, B., Sahay, S., 2004. Embedded knowledge and offshore software development. Inf. Organ. 14 (4), 329–365.

O'Leary, M.B., Cummings, J.N., 2007. The spatial, temporal, and configurational characteristics of geographic dispersion in teams. MIS Q. 31 (3), 433–452.

Padberg, F., 2002. A discrete simulation model for assessing software project scheduling policies. J. Softw. Process Improv. Pract. 7, 127–139.

Patil, S., Kobsa, A., Ajita, J., Seligmann, D., 2011. Methodological reflections on a field study of a globally distributed software project. Inf. Softw. Technol. 53 (9), 969–980.

Plateau, B., 1985. On the stochastic structure of parallelism and synchronization models for distributed algorithms. ACM SIGMETRICS Perform. Eval. Rev. 13 (2), 147–154.

Prikladnicki, R., Audy, J., 2010. Process models in the practice of distributed software development: a systematic review of the literature. Inf. Softw. Technol. 52 (8), 779–791.

Prikladnicki, R., Audy, J., Shull, F., 2010. Patterns in effective distributed software development. IEEE Softw. 27 (2), 12–15.

Prikladnicki, R., Carmel, E., 2013, May. Is time-zone proximity an advantage for software development? The case of the Brazilian it industry. In: 35th International Conference on Software Engineering (ICSE), pp. 973–981.

Raffo, D., Setamanit, S., 2005. A simulation model for global software development project. Development, 1–7.

Raj, J., 1991. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons Inc, New York, NY, USA.

Ramasubbu, N., Cataldo, M., Balan, R., Herbsleb, J., 2011. Configuring global software teams: a multi-company analysis of project productivity, quality, and profits. In: 33rd International Conference on Software Engineering (ICSE), pp. 261–270.

Sales, A., 2012. SAN lite-solver: a user-friendly software tool to solve SAN models. In: Spring Simulation Multi-conference (SpringSim'12): SCS/ACM Theory of Modeling and Simulation: DEVS Integrative M&S Symposium (DEVS 2012), SCS/ACM, Orlando, FL, USA, pp. 9–16, pp. 44.

Sangwan, R., Bass, M., Mullick, N., Paulish, D., Kazmeier, J., 2006. Global Software Development Handbook (Auerbach Series on Applied Software Engineering Series). Auerbach Publications, Boston, MA, USA.

Sarker, S., Sahay, S., 2002. Understanding virtual team development: an interpretive study. J. Assoc. Inf. Syst. 3, 247–285.

Sere, F.C., Swigger, K., Alpaslan, F.N., Brazile, R., Dafoulas, G., Lopez, V., 2011. Online collaboration: collaborative behavior patterns and factors affecting globally distributed team performance. Comput. Hum. Behav. 27 (1), 490–503.

Setamanit, S., Wakeland, W., Raffo, D., 2007. Using simulation to evaluate global software development task allocation strategies: Research sections. Softw. Process Improv. Pract. 12, 491–503.

Sooraj, P., Mohapatra, P., 2008. Modeling the 24-h software development process. Strateg. Outsourcing Int. J. 1 (2), 122–141.

Stewart, W.J., 1994. Introduction to the numerical solution of Markov chains. Princeton University Press, Princeton, NJ, USA.

Taweel, A., Brereton, P., 2006. Modelling software development across time zones. Inf. Softw. Technol. 48 (1), 1–11.

Treinen, J.J., Miller-Frost, S.L., 2006. Following The sun: case studies in global software development. IBM Syst. J. 45, 773–783.

Urdangarin, R., Fernandes, P., Avritzer, A., Paulish, D.,2008. Experiences with Agile practices in the global studio project. In: Proceedings of the IEEE International Conference on Global Software Engineering (ICGSE 2008). IEEE Computer Society, pp. 77–86.

van Solingen, R., Valkema, M.,2010. 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: 5th IEEE International Conference on Global Software Engineering (ICGSE 2010). IEEE Computer Society, Princeton, NJ, USA, pp. 165–174.

Visser, C., Solingen, R.V., 2009. Selecting locations for Follow-The-Sun software development: towards a routing model. In: 4th IEEE International Conference on Global Software Engineering (ICGSE 2009), IEEE Computer Society, Limerick, Ireland, pp. 185–194.

Yap, M.,2005. Follow the Sun: distributed extreme programming development. In: Proceedings of the Agile Development Conference (ADC '05). IEEE Computer Society, Denver, CO, USA, pp. 218–224.

Alan R. Santos is Ph.D. student in Computer Science at Pontifical Catholic University of Rio Grande do Sul, Brazil. Alan got his M.Sc. degree (2012) in Computer Science from Pontifical Catholic University of Rio Grande do Sul, Brazil. His current research topics include practical applications of performance and reliability modeling applied to software engineering, agile methods and mobile application development.

Afonso Sales is university lecturer in Computer Science at Pontifical Catholic University of Rio Grande do Sul, Brazil. He got his Ph.D. degree (2009) in Computer Science from Grenoble Institute of Technology, France. He has wide knowledge about state space generation techniques using decision diagrams and numerical solution methods based on structured descriptions of Markovian models. His research interests include stochastic modeling and simulation, continuous and discrete time modeling, structured Markovian formalisms, structured and Kronecker based approaches for Markov analysis, as well as performance evaluation of systems applied to software engineering, and parallel and distributed computing.

Paulo Fernandes is Professor of Computer Science and research leader of the Performance Evaluation Group and PaleoProspec project at Pontifical Catholic University of Rio Grande do Sul, Brazil. He got his Ph.D. degree in Computer Science from Institut National Polytechnique de Grenoble, France (1998), M.Sc. and B.Sc. degrees also in Computer Science from Federal University of Rio Grande do Sul, Brazil (1987 and 1990). His research interests include numerical solution of stochastic modeling formalisms for large Markovian models. His current research topics include structured Markovian formalisms, as well as practical applications of performance and reliability modeling applied to software engineering.