Detection of Bottleneck and Social Network in Business Process of Agile Development

Nemesio Raitubu
Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
Nemesioroko15@gmail.com

Kelly Rossa Sungkono
Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
kelly@its.ac.id

Riyanarto Sarno
Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
riyanarto@if.its.ac.id

Cahyaningtyas Sekar Wahyuni
Informatics
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
cahyaningtyas.18051@mhs.its.ac.id

***Abstract*— Many software organizations are moving away from traditional methods to adopt Agile development methodologies with the increasing popularity of Agile. It was noted that Agile is more adaptive and people focused. It was created for small to medium and collaborative team that work closely together. When Agile is implemented, team size a common factor in turn constrained by people factor which are often overlooked. Many organizations and industries are migrant of Agile methodologies because of its popularity in today's technology. One of the main assets of any organization are the employees because they bring out the important success of the organization to make the best use of the resources available in the organization. for this, it will be easy to measure the job performance evaluation of an organization. The aims of this study are to investigate the use of Agile methodology for the performance analysis evaluation of software development. it is difficult to find the bottleneck and dependencies between members of the team if we do observe the process manually. This paper proposes an agile process model that can generate an event log and to analyzed the event log by using alpha ++ and social network analysis to find the bottle neck and dependencies of the member.**

Keywords: Agile, Bottleneck, Performance analysis, Social network

I. INTRODUCTION

In making software development like Agile [1], Developer had some major influence in way that it should be done. It is because the software needs to accommodate some changes that was plan to be created in the process of their work, in ways how the customer and stakeholder can communicate with software and also in the nature of the sizes of the organization including small, medium or large.

In creating Agile software their methods taken to be used that are based on the development of interactive and incremental. Within the process there are characteristic of the method used in Agile including: planning to use powerful, easy to use and low cost packages, building a software which is composed of many iteration in sequence and self-contained, teams focus and continue assess their performance to a high level perspective and promoting communication between the software and the user.

In obeying the process [2]–[7] principles of the software development, the software should be light and sufficient enough, members of the organization get along with software and a communication centered perspective. Lightweight process known as small project are suitable in this type of development. Agile Software Development team always starting from simple then then continue in incrementing the idea of the requirements throughout the whole process of development.

In all the stages of production activity including design, coding and testing require further refines so that the developed software is correct and be useful as the final software itself.

In contras of classical software development process [6] and Agile development software, classic software all the requirements are completed in the process and no changes can be done in modifying improvement whereas in Agile software development it accepts changes and improvement frequently when changes arise.

II. RELATED WORK

A. Agile softwre development

Agile [1] can be define on its flexibility or moving quickly from other software development programs. It also creates an approach in dealing with inspection, self-organization and monitoring, the fast in delivering of the quality software and the approaches to take for the company goals to be achieved. Agile software development is described as iterative and incremental because all phases are revised throughout the life cycle.

The approach of agile software development has been defined in the Agile manifesto. Twelve principles have been captured in the philosophies in 2000 where members set up an Agile software development. These twelve principles including:

- customer satisfactions
- changeable - allowing changes at any point of time
- Time frame that are used in the process.
- Customer and business people oriented
- motivated individual and good working environment

978-1-7281-3832-9/19/$31.00 ©2019 IEEE

- Face-to-face conversation helps in the development within members
- progression of the software
- development pace to be maintain
- continuous attention to be given in a good design
- maximizing the work done
- Self-organizing teams
- effective team work.

B. Agile Process Modelling

Incremental and Iterative model are both combination of the Agile process model and customer involvement at this stage of development process is highly needed to accommodate the basic purpose in all activities. Fig 1 shows the agile process model which including the process of Requirements, Design, Development, Testing, Implementation and maintenance.

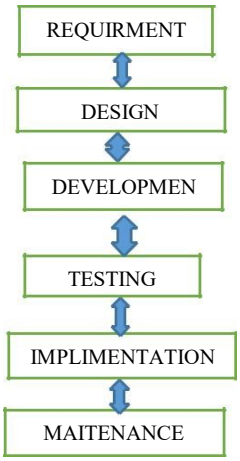

Fig.1 Agile model

Advantages of Agile Model in software development

- It Support customer satisfaction and involvement
- It provides strong communication oriented with the customer.
- Focus on user friendly
- Fast development
- Changes can be done at any time
- Low cost software
- Focus on team works within organization
- Rapid delivery
- Less planning required in the process

Disadvantages of Agile Model in software development

- Large projects are not their choice
- Customer confusion in the process time can create problems if they want to change the requirements.
- One of the problems faced by many organizations is the lack of documentation which leading to changing of software that need to be maintain.
- Full support for documentation and design are not accommodated.

C. Social Network Analysis

Social network analysis (SNA) [8]–[11] is the process of investigating social structure through the use of networks and graph theory. The degree in social network analysis attempts to investigate between the relationships and interactions of people, team, departments, ministries within the organization. SNA also improve on the flow of communication within the organization and also giving the Heads of the organizations to have access in the work done and to see the improvement of the development of software. To provide a transparent communication process and proper tools to make it better is the goal of Social network analysis. The graphic presentation of Social Network Analysis is one of the important aspects because all its techniques rely upon it. Each social network is represented by graph in which each node is a person and each link between two nodes (people) are in a connected relationship.

D. Social Network Mining

The Social Network Mining reads through a process log and generate social network that usually Social Network used it as a starting point. Analysis. Several techniques can be applied to analyze the social network, for example, in finding the interaction patterns, and evaluate the role of an individual in an organization. Social network is based on human interactions. There are two major strategies for data mining tasks for social networks:

- structure-based
- content-based.

There are two kind of data instances collected in the social network:

- static
- dynamic.

III. RESEARCH METHOD

a. Creating a Process Model

Fig 2 represent the Agile process model including Requirements, Design, Development, Testing, Implementation and Maintenance.

Requirements define and document the product requirements and get them approved by the customer.

Design is where the best architecture product to be develop by uploading of forms related to design.

Development is where the actual development product is built where programs upload files and descriptions of each requirement made.

Software testing is the stage where the product is to be tested.

Implementation its where it can be seen uploading the user manual/ upload photos and video as a proof of implementation.

Maintenance is the progress form from the program whether or not a bug is fixed and also the feedback of the program.

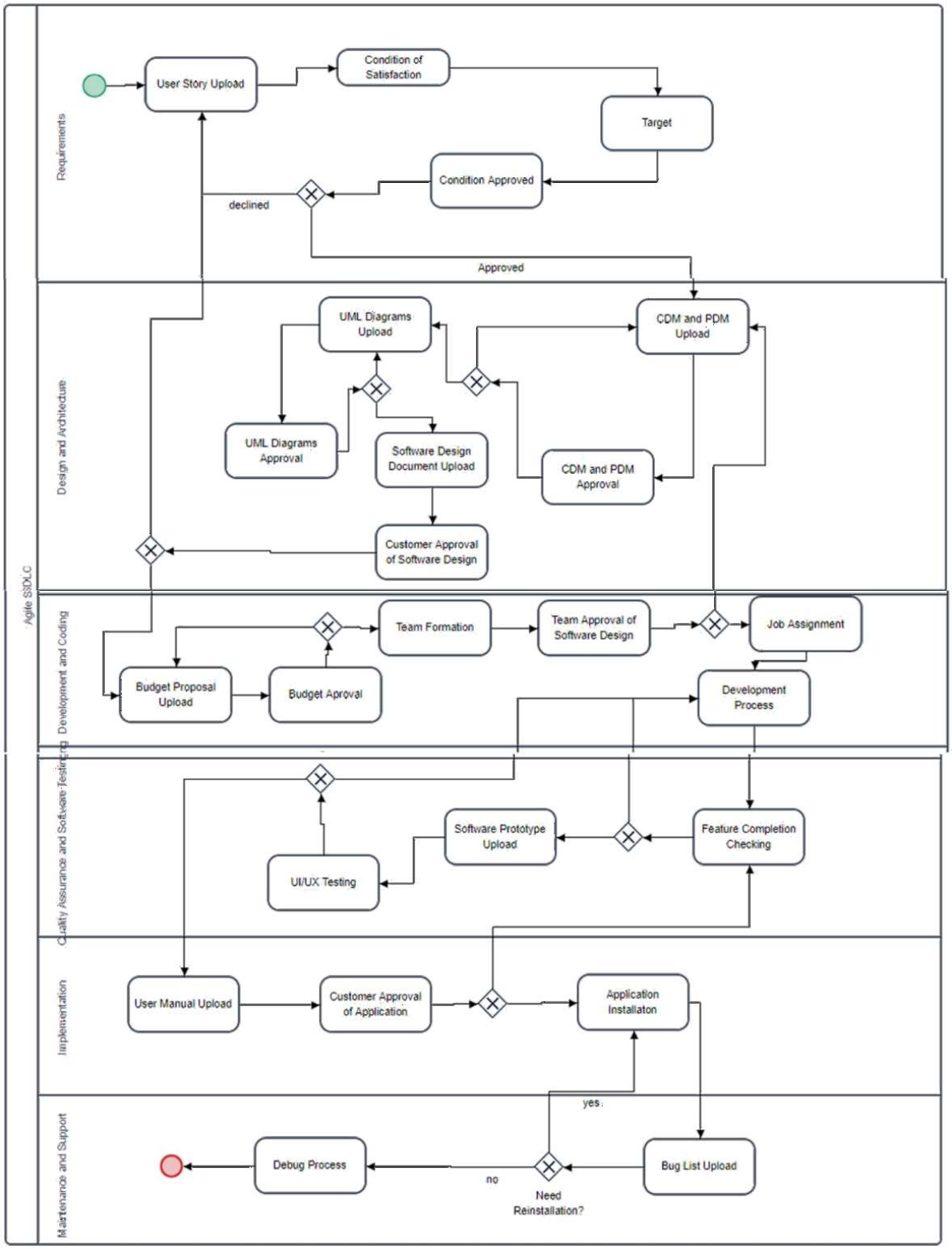

Fig.2 Process Model

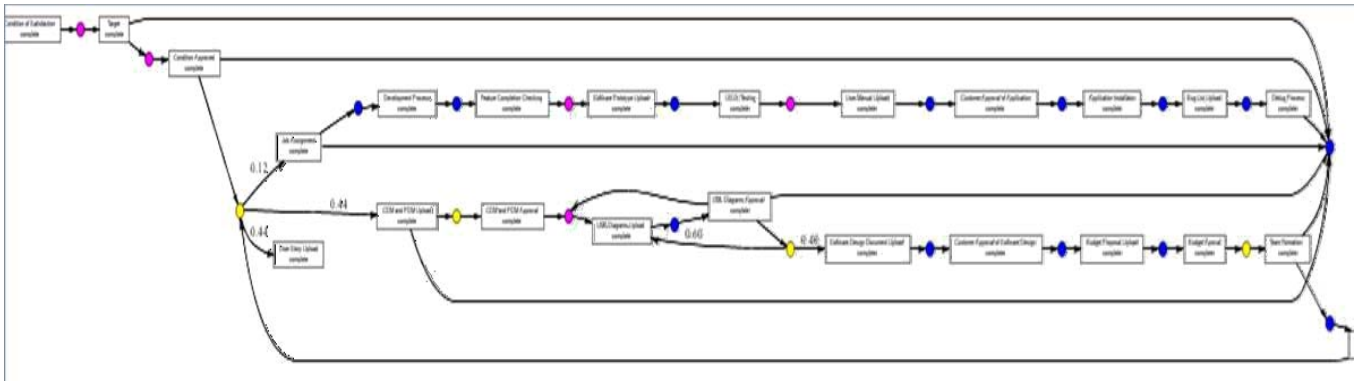

Fig.3 alpha ++ algorithm – PROM

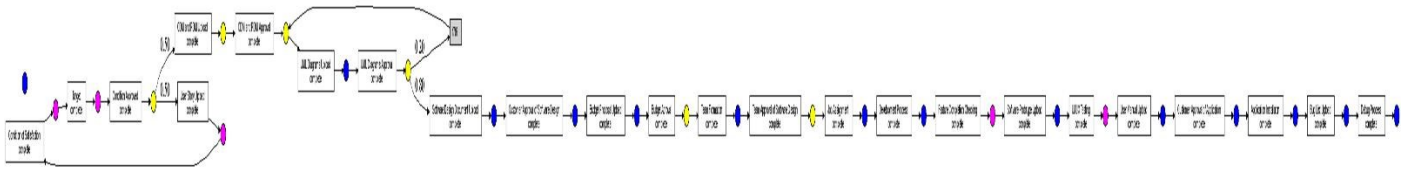

Fig. 4 Heuristic algorithm – PROM

b. Alpha++ and Heuristic model

The two algorithms that are used in this project are alpha ++ and heuristic miner algorithms. After getting the result from these two algorithms we run the data using ProM to get the bottleneck [2] as a result. The models are shown in Fig. 3 and Fig. 4. These two models show that the flow of the business process model goes right up to the final stage.

There is a total of nine (9) cases were selected and their arrival rate is 2.46 cases per second. Process information are as follow:

- Average – 918786.67
- Minimum – 300,
- Maximum– 2319000,
- Standard deviation – 1083267.84,
- Fast – 600,
- Slow = 2318940,
- Normal - 726000

IV. RESULT AND ANALYSIS

The aim of the project is to design a software using agile method. In order to find the result of time taken for the software to be build and other steps in carrying out the building of this project. Two data mining tools was used for this project are Disco and Prom and their result will be discussed

Disco uses the csv excel file that was created from the Process Maker database. All the tasks and cases created in the process Maker application are being stored in the PhpMyAdmin. Csv excel consisting of Caseid, Name Activity, Time Stamp and Resources was generated from it and used by the Disco to find the frequency and to get the mxml file to be used my PROM.

TABLE I. excel file – ProcessMaker consisting of caseid, Nameactivity, timestamp and resources.

Caseid	NameActivity	TimeStamp	Resources
1043361135ca467161	Condition of Satisfaction	03/04/2019 07:59	mirry
1043361135ca467161	Target	03/04/2019 08:01	Emosi
1043361135ca467161	Condition Approved	03/04/2019 08:02	Emosi
1043361135ca467161	User Story Upload	03/04/2019 08:03	nem
.	.	.	.
.	.	.	.
.	.	.	.
1043361135ca467161	CDM and PDM Upload	29/04/2019 15:43	mirry
1043361135ca467161	CDM and PDM Approval	29/04/2019 16:08	kevin

Table1 is part of data will be used in this project. This data contained case id and activity, timestamp and the resources. Case id is a path to, trace activity to complete the process. Name activity is the activity in the project, time stamp is the time the activity took place and the resources is the person who is in charge of the activity.

The mxml file that was generated from the disco are used by PROM for analysis to find the waiting time of each task. There are nine (9) cases altogether from this project with their different throughput time.

Alpha++ algorithm was used to reconstructing causality and generate the activity of each cases.

Fig. 3 show the business process model that was the product of alpha++ algorithm with the location of the software. The pictures in fig.3 show the 3 different magenta-colored tasks, that gives the average waiting time.

- Pink color - high waiting time then
- Yellow color - medium waiting time and the
- Blue color - low wait time.

Out of the 24 Activities in the model, High and medium waiting time have 5 each and low waiting time have 14 activities.

Activity with waiting time are Target, Condition approved, User story upload, Software prototype upload, User Manual upload. Activity with medium waiting time are CDM PDM upload and approval, Software Design Upload, Team formation, Job Assignment and activity with low waiting time are Condition Satisfactory, UML Diagram Upload, Customer Approval, Budget Proposal, Budget Approved Team Approval of software design, Development prose, Features Completion, UI-UX Testing, Customer Approval of application, Application Installation, Bug list upload, and Debug process.

The bottleneck will be different for each algorithm in each case. In each case the three different colors will be given where the occurrence of bottleneck making decision in the task of fast processing. In the calculation of bottleneck, it only gives the time difference required between the connected transaction and the place in the case and it where the average waiting time between each place will be finalized.

Decision making process occurs in the long waiting time which change the decision making based on the quality of orders in the within the process. Thus, it will have an effect on the delivery whether it will take the quick delivery or requires long waiting until the quota of order is all fulfilled. This is where the decision-making process occurs.

From each case that will be analyzed, bottleneck determination will be highly noted. This is done because the possibility of bottleneck going high in the process. Petri net in ProM was used to viewed the bottleneck analysis performance and main purpose of this analysis is to provide a way to assess the process performance of event logs. It is possible to see the places that are jams by using performance analysis Petri net.

From the test result, it is shown that the algorithm has an important role in a precision bottleneck. We get many accurate bottleneck results from a more precise business process. The alpha++ don't shows the frequency of modeling business process whereas in Disco the frequency of the process model [12]–[15] was calculated and also detect any noise in the data.

The results also revealed that the blue color bottleneck got low waiting time but have a fast throughput time because the task is generated in order from the process model. All the task has been approved from the start until the final steps. Whereas, in other two colors, yellow and pink, they have high and medium waiting time because the process to execute a task take much because in the process some of the task are incomplete and its need to go back and re redo the process so that it can be approved. The total throughput time that determine bottleneck of each cases are shown in fig.4.

Table 2 Social Network Mining result table consisting the number of dependencies each member has.

	Admin	Emosi	kevin	mirry	nem
Admin	0.0	0.0	0.0	0.0	0.0
Emosi	0.0	0.0	0.0	0.0	0.02
kevin	0.0	0.0	0.04	0.01	0.01
mirry	0.0	0.02	0.0	0.0	0.0
nem	0.0	0.0	0.0	0.0	0.18

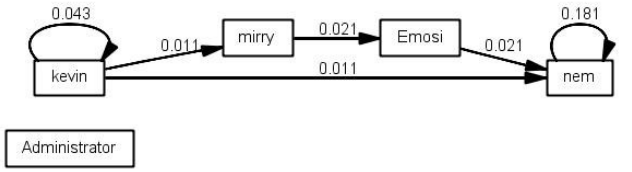

Fig. 5 Social Network Mining Result

From TABLE II and graph in Fig. 5 show the result in SNM from the project in PROM. It shows how each member of the organization dependent to each other to complete a task.

The highest frequency from the graph and table is 0.181 (nem -nem), 0.0423 (kevin – kevin) 0.021 (mirry – moss and mirry – nem), 0.011 (kevin – mirry and kevin – nem). The first two highest frequency are from individual member which they are dependent for themselves.

V. CONCLUSION

By using the two algorithms, Alpha++ algorithms and Heuristic miner the result shows bottlenecks and fraud in a business process. Four conclusions can be derived based on the evaluation above.

Firstly, are the exact details of business process depended on the explicit of its calculation of bottleneck The higher the accuracy the more precise the business process. In the algorithm used, Alpha++ algorithm do not examine the frequency whereas in Disco it considers the frequency of the business modeling process so that any noise in the data can be detected.

Secondly, is the existence of the waiting time. This enable to make decision in the process base on the model used. It is where the result is identified until the order is fulfilled from the fast or slow process of time taken between the decision-making and the bottleneck.

Thirdly, is the achievement of using Disco and alpha++ algorithm. We can say that Disco has a accomplish more than alpha++ algorithm because it considers to calculate the frequency whereas alpha++ algorithm cannot calculate frequency.

And the last conclusion is the dependency of individual within the organization. PROM also show the frequency of how each member of the organization are dependent to each other in the completion of the software from the highest to the lowest frequency.

REFERENCES

[1] S. Sharma, D. Sarkar, and D. Gupta, "Agile Processes and Methodologies: A Conceptual Study," *International Journal on Computer Science and Engineering*, vol. 4, no. 5, 2012.

[2] Y. Caesarita, R. Sarno, and K. R. Sungkono, "Identifying bottlenecks and fraud of business process using alpha ++ and heuristic miner algorithms (Case study: CV. Wicaksana Artha)," *Proceedings of the 11th International Conference on Information and Communication Technology and System, ICTS 2017*, vol. 2018-Janua, pp. 143–148, 2018. http://doi.org/10.1109/ICTS.2017.8265660.

[3] R. Sarno, K. R. Sungkono, R. Johanes, and D. Sunaryono, "Graph-Based Algorithms for Discovering a Process Model Containing Invisible Tasks," *International Journal of Intelligent Engineering and Systems (IJIES)*, vol. 12, no. 2, pp. 85–94, 2019.

[4] L. Hakim, R. Sarno, and K. R. Sungkono, "Modified Alpha++ Algorithm for Discovering the Hybrid of Non-Free Choice and Invisible Task of Business Processes," *International Journal of Intelligent Engineering and Systems (IJIES)*, vol. 12, no. 3, pp. 31–40, 2019.

[5] K. R. Sungkono and R. Sarno, "Constructing Control-Flow Patterns Containing Invisible Task and Non-Free Choice Based on Declarative Model," *International Journal of Innovative Computing, Information and Control (IJICIC)*, vol. 14, no. 4, 2018.

[6] R. Sarno and K. R. Sungkono, "Coupled Hidden Markov Model for Process Mining of Invisible Prime Tasks," *International Review on Computers and Software (IRECOS)*, vol. 11, no. 6, pp. 539–547, 2016. http://doi.org/10.15866/irecos.v11i6.9555.

[7] K. R. Sungkono, R. Sarno, and N. F. Ariyani, "Refining business process ontology model with invisible prime tasks using SWRL rules," in *2017 11th International Conference on Information Communication Technology and System (ICTS)*, 2017, pp. 215–220. http://doi.org/10.1109/ICTS.2017.8265673.

[8] M. Song and W. M. P. van der Aalst, "Towards comprehensive support for organizational mining," *Decision Support Systems*, vol. 46, no. 1, pp. 300–317, 2008. http://doi.org/10.1016/j.dss.2008.07.002.

[9] F. Bonchi, C. Castillo, A. Gionis, and A. Jaimes, "Social network analysis and mining for business applications," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 2, no. 3, p. 22, 2011. http://doi.org/10.1145/1961189.1961194.

[10] V. Raghavan, G. ver Steeg, A. Galstyan, and A. G. Tartakovsky, "Coupled hidden markov models for user activity in social networks," 2013. http://doi.org/https://doi.org/10.1109/icmew.2013.6618397.

[11] X. Ding, T. Liu, J. Duan, and J.-Y. Nie, "Mining User Consumption Intention from Social Media Using Domain Adaptive Convolutional Neural Network," in *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 2389–2395.

[12] W. M. P. Van Der Aalst and A. H. M. Hofstede, "YAWL : yet another workflow language," vol. 30, pp. 245–275, 2005. http://doi.org/10.1016/j.is.2004.02.002.

[13] A. Rozinat, "Disco User 's Guide," pp. 1–155, 2012.

[14] a K. a De Medeiros, W. M. P. Van Der Aalst, and A. Weijters, "Workflow mining: Current status and future directions," *The Move to Meaningful Internet Systems 2003 CoopIS DOA and ODBASE*, vol. 2888, no. i, pp. 389–406, 2003. http://doi.org/10.1007/978-3-540-39964-3_25.

[15] F. M. Maggi, M. Montali, M. Westergaard, and W. M. P. Van Der Aalst, "Monitoring business constraints with linear temporal logic: An approach based on colored automata," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 6896 LNCS, pp. 132–147, 2011. http://doi.org/10.1007/978-3-642-23059-2_13.