ISSN 1996-3343

Asian Journal of Applied Sciences

Knowledgia SCIENTIFIC
A Place to Publish Outstanding Research

http://knowledgiascientific.com

OPEN ACCESS Asian Journal of Applied Sciences

ISSN 1996-3343
DOI: 10.3923/ajaps.2016.131.142

Review Article
A Review on Software Process Mining Using Petri Nets

[1]R. Thamizharasan and [2]Kumaravel Appavoo

[1]Department of Computer Science, Bharath University, 600 073 Chennai, India
[2]School of Computing, Bharath University, Tambaram, 600 073 Chennai, India

Abstract
The theory of regions and the algorithms for synthesizing a Petri Nets model from a transition system, which are based on this theory, have motivating practical applications-in particular in the design of electronic circuits. In this study, it is discuss about the several research about software process using Petri Nets. Much study has to be done in put on the mining and synthesis algorithms to different document management systems in different application areas and making practical assessment of them both in the area of business process management and software process engineering. Since method used in this study is also pertinent to the area of mining the activity logs, in the future, we should also compare it to the existing approaches in this area. This study aims at making the first step from the well-developed theory of Petri Net synthesis to the practically relevant research domain of process mining. Here, it show that this theory can be also applied for mining the underlying process from the user interactions with a document management system. In have invented a new Petri Net model and compare with the other models.

Key words: Software engineering, software process, mining, Petri Nets, algorithms, model

Received: October 27, 2015 Accepted: January 13, 2016 Published: June 15, 2016

Citation: R. Thamizharasan and Kumaravel Appavoo, 2016. A review on software process mining using Petri Nets. Asian J. Applied Sci., 9: 131-142.

Corresponding Author: R. Thamizharasan, Department of Computer Science, Bharath University, 600 073 Chennai, India

Competing Interest: The authors have declared that no competing interest exists.

Data Availability: All relevant data are within the paper and its supporting information files.

INTRODUCTION

Today, there is a bunch of techniques that help to routinely come up with process models from a sequence of activities that are executed in an enterprise[1]. Typically, such sequences come from the log of a workflow management system or some standard software which is used for executing these processes. There are many different novel algorithms and methods that help to obtain faithful and valuable process models; some techniques come up with an initial model rather fast and the process models are incrementally improved by new interpretations[2]. All these techniques can be summarized by the term process mining. Here interest in process mining came from the area of software engineering.

Software engineering processes are often not well-documented, though good engineers have the processes in their minds. In the Capability Maturity Model (CMM), this level of maturity of a software company is called repeatable[3-7]. Therefore, it is looked for methods for automatically mining these process models from the observed study. The main source for observing the work of software engineers are the logs of the version management systems and document management systems that are used in the development process. The problem, however, is that these systems are aware of documents only and not of the underlying activities. Basically, they see the creation, modification and checking of documents, but they are not aware of the activities and to which activity these events belong to Assar *et al.*[8], Baeza-Yates[9] and Baeza-Yates *et al.*[10]. Therefore, the standard mining algorithms do not work; the activities must identify from the event logs of the document management systems before: Here it is call this activity mining (Fig. 1).

Here, it could easily obtain a transition system for the underlying processes, where the transitions are the activities of the processes. So, basically, deriving a process model from the result of the activity mining algorithm means deriving a Petri Net from a transition system, which is a well-known area of Petri Net theory called Petri Net synthesis. It was established by the seminar paper on regions and later extended and elaborated by other researchers. In this study, we show that our activity mining algorithm in combination with the tool petrify[2,11-13] can be used for faithfully mining process models from logs of document management systems and version management systems. The focus of this study is on the use of synthesis algorithm for details on the activity mining algorithms.

The practical relevance of process mining is increasing as more and more event data becomes available. Process mining techniques aim to discover, monitor and improve real processes by extracting knowledge from event logs. The two most prominent process mining tasks are: (1) Process discovery: Learning a process model from example behavior recorded in an event log and (2) Conformance checking: Diagnosing and quantifying discrepancies between observed behavior and modeled behavior[14-16].

Most of the study done in conformance checking in the literature focuses on the control-flow of the underlying process, i.e., the ordering of activities. There are various approaches to compute the fraction of events or traces in the log that can be replayed by the model[17-19].

Petri Nets are popular due to their inherent ability to express concurrency, choice and causality between events in a system, without explicit enumeration of global states.

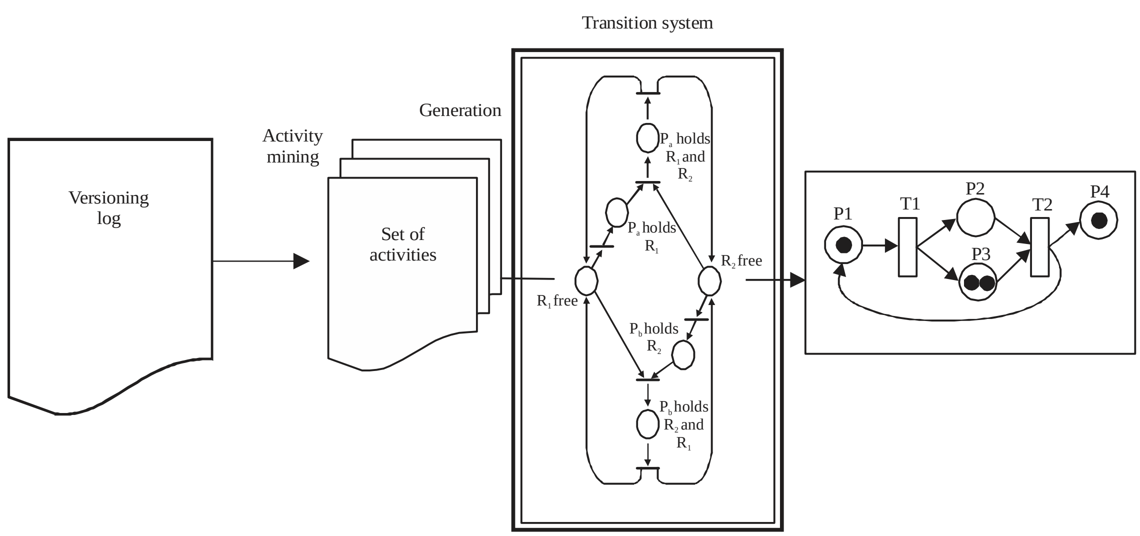

Fig. 1: Mining and synthesis schema

Although checking properties of Petri Nets could be difficult in general, for some subclasses of Petri Nets there are efficient verification algorithms[20].

The work presented in this study is related to process mining, i.e., discovering a process model based on some event log. Process mining techniques focus on discovering behavioral aspects from log data. The idea of applying process mining in the context of workflow management was first introduced by Agrawal *et al.*[21].

A similar idea was used in the context of automating the detection of process models using probabilistic and algorithmic methods by Chen and Yun[22].

Cook and Wolf described a Markov method that they developed exactly for process discovery. This model is integrally successive. Chow[23], Christie[24], Chulef *et al.*[25], Clauzel *et al.*[26] and Conklin and Begeman[27] the exploring techniques that can use basic event data captured from an on-going process to generate a formal model of process behavior. In this kind of data analysis process discovery, they describe using methods: Algorithmic grammar inference, Markov models and neural networks. Note that the results presented in[28-34] are limited to successive behavior[35].

Healthcare is another famous application domain for process mining. The applicability of process mining in healthcare was demonstrated using a real case of a gynecological oncology process in the AMC hospital in the Netherlands[36-39]. The log data contained information about a representative group of 627 gynecological oncology patients. The goalmouth of using process mining was to discover the care paths followed by individual patients and whether certain procedures are followed or not. After applying process mining techniques, many useful results became visible to the people at the hospital. For example, it was found that patients who undergo several chemotherapy sessions often need to visit the dietician. This was not immediately clear to everyone and illustrates the value of creating transparency using process mining[40-45].

The above two mentioned projects were implemented with the process mining tool named ProM[46]. The ProM contains more than 250 plug-ins that implement different process mining algorithms. However, it is not clear how to use ProM in process redesign projects. In the above two projects, the used different plug-ins were used but viewed each plug-in result alone. Although, ProM allows the results from some algorithms to be integrated in a Colored Petri Net (CPN) that support analysis and simulation, there was no guidance from ProM on how to improve the business processes. Instead, the researchers concluded the redesign ideas from viewing the simulated models. It is hard to make process redesign using process mining a repeatable service.

Process mining has been applied in a variety of organizations covering many application domains. In the process mining was used to analyze the test process in ASML. The ASML makes so-called wafer scanners that are used to manufacture processors in devices ranging from mobile phones to desktop computers. Wafer scanners are really complex machines that use a photographic process to image nanometric circuit patterns onto a silicon wafer. The testing of the manufactured wafer scanners is a time-consuming process. So, the goal of the analysis was to reduce the testing time. Each wafer scanner in the ASML factory produces a log of the software tests that are executed on it. Process mining was used to visualize the actual flow of the test process and confront this visualization with the idealized view of the tests according to engineers. It was found that as soon as one test fails, a fix is made to the scanner and all other tests are put on hold (idle time) and often after the fix is made, some tests are re-executed again. Visualizing this loop-backs caused by some tests gave the engineers a useful view on what was causing the time loss in the test process. Hence, allowed them to make changes to the test process to reduce the time (for example, execute some tests at earlier phases)[47-53].

Nabil R. Adam proposed modeling and analysis of workflows using Petri Nets" in which he has demonstrated the use of PN as an effective tool for modeling workflows at a conceptual level and then analyzing them.

Cintra and Ruggiero presented a simulation technique for performance analysis of Generic Petri Net models of computer systems" in which he presented a simulation algorithm tp observe that the simulator performed reasonably well, even on a modest machine.

Boucheneb and Hadjidj proposed model verification techniques of time Petri Nets[54-59]. They used temporal logic model checking to represent the behavior of a system.

Olivier and Roy introduced an approach to implement a distributed monitor of real-time system properties and then introduced a new formalism, adaptive Petri Nets, that allows to model such complex, distributed and real time systems[60-69].

The performance analysis of the model illustrates the behavior of the system. Falko Bause proposed the concept of stochastic Petri Nets with various examples. In Stochastic Petri Nets (SPN), random string delays are attached to the transition. The SPN is used for performance analysis of the system by Markovian techniques.

Michael K. Molloy proposed the performance analysis of the system using Stochastic Petri Nets[70]. They used Stochastic Petri Net for performance analysis of alternating bit protocol.

Bernardi proposed a structural performance evaluation methodology for Timed Petri Nets (TPNs) and their stochastic extensions[71].

The W.M.P. van der Aalst proposed a methodology to verify the business process using Petri Nets[72-76]. In which he verified the liveness and boundedness of workflow net using Petri Nets.

Boucheneb and Hadjidj proposed model verification techniques of time Petri Nets[77]. They used temporal logic model checking to represent the behavior of a system.

Other recent work has sought to measure process conformance, having obtained a sequence of events, by comparing the models obtained from observations to theoretical models. This study is in the early stages having come from the foundations of methods that simply tested conformance. Fitness and appropriateness are proposed as metrics that can be assessed by incrementally replaying the events and measuring unused features of the model, respectively[78-83]. Two further metrics precision (how many invalid steps occur) and recall (how many steps are enabled) have also been defined from a machine behaviour perspective. In this study, here it is focus on the method proposed, which represents the process model as a Petri Net[84-90]. The log of a series of events will refer as a task. When the task is executed a token is produced at the start state, then when an event is executed if an edge with the same name is enabled (that is to say there is a token on the preceding state) then the token is consumed and one produced at each connected state. The fact that an edge may lead to more than one state allows for parallel execution. If there is no enabled edge matching the event then one which has the same name is chosen at random and the same process is followed, without the consumption. The method described twofold based on two metrics fitness and appropriateness[84-89]. Fitness is a measurement of the extent to which the tasks can be fitted to a Petri Net captivating into account the number of times tokens are produced but not consumed and the number of missing tokens where they had to be introduced. Appropriateness is a measurement of over fitting, or in other words how much of the process model was not used by the tasks executed. Behaviour suitability measures if the model allows too many possible paths. This is achieved by calculating all possible orderings of events in both the model and the log and discounting those that always or never follow each other. The metric calculates the remaining set (those states which sometimes follow each other) by considering the difference between the model and log sets. The calculation is founded on a count of alternate duplicate events and unnecessary imperceptible events that could be removed without otherwise moving the behaviour of the model. The alternate duplicate events are those that never happen together in one execution sequence. This measurement is made on the model only and so will not be affected by changes to the logs.

This algorithm was proposed to rebuild the causality in the Petri Nets workflow from the existing relations in the event log. The α-algorithm takes the event logs as input, rebuilds process models by using simple XOR and splits and joins; thereby creates the workflow nets as output. The α-algorithm cannot handle noise and certain complicated routing constructs of workflow nets such as, loops and long-term dependencies, particularly during complex situations. A more robust but less precise approach was then proposed to deal with the issues of α-algorithm. To overcome this difficulty a protracted algorithm, α++ algorithm was introduced to generate new relationships between event logs to handle long-term or implicit dependencies[91-96].

Hierarchical clustering this technique separates a set of event logs for a given process into clusters and finds the dependency graph for each log[97-109]. It structures the clusters of event logs into a hierarchy tree. For each cluster, a workflow model is constructed and finally all the models are merged into a single one. Some clustering techniques use theory of regions to discover processes. The advantage of the theory of regions is that the characteristics of the resulting model can be influenced before the mining starts (e.g., the number of places in the Petri Net or the number duplicate task can be deter-mined in advance). A mining tool has been developed for discovering hierarchically structured workflow processes that need to balance splits and joins[110-118].

Processes are frequently expressed as a form of directed transition system[119-124]. These are composed by events or activities. Approaches to forming these from observations vary based on the type of events captured. Principally there are three observation approaches discussed in the literature: In the first case the developers document the process in detail as they complete it although this is maybe unreliable; secondly a research group observes the team, perhaps recomposing the process by interview[125-127]; lastly the final method involves the analysis of logs *post hoc*[128-131]. In the former instance some knowledge of the process model and the relation between the states is known, whereas in the latter two all that is available is the event series in the timeline. In the later case where the data is reconstructed, business rules must be identified in order to associate a log entry with the activity stereotype.

Genetic algorithm[132]: This technique provides process models (Petri Nets) built on causal matrix, i.e., input and output dependencies for each activity. This technique tackles problems such as, noise, incomplete data, non-free-choice constructs 4, hidden activities, concurrency and duplicate activities. Nevertheless, it requires the configuration of many parameters to deal with irrelevant data, which is a complex task.

Heuristic algorithm[133-136]: This technique is based on α-algorithm. It calculates the frequencies of relations between the tasks, e.g., causal dependency, loops, etc and construct dependency/frequency tables and dependency/frequency graphs. This technique can detect irrelevant logs. However, like the Genetic algorithm, Heuristic algorithm needs a complex configuration phase.

DEVELOPMENT OF THE TCPN SIMULATION MODEL FOR THE PATIENT CARE FLOW PROCESSES

The CPN Tools (version 4.0.1) was used in constructing a timed colored Petri Nets simulation model for the considered patient care flow processes. The proposed TCPN simulation model consists of 16 places and 12 transitions.

Simulation model, places are draw as ovals while transitions are drawn as rectangle. Places and transitions are connected with directed arcs which model the relations among the individual elements of the developed model. The arcs with their arc expressions define the flows of tokens in the net. The descriptions of the major places and transitions in the proposed TCPN simulation model are stated in Table 1 and 2, respectively. The color sets, variables, initial parameters and functions that are needed in developing the TCPN simulation model of the patient care flow processes are depicted in Fig. 2.

The color sets, variables, initial parameters and functions that are needed in developing the TCPN simulation model of the patient care flow processes are depicted in Fig. 3. The simulation of the proposed TCPN model was carried out using CPN tools. Due to the fact that the simulation model is stochastic, it is necessary to execute several simulation runs with the proposed model in order to compute mean value. Hence, several replications were run for each day and average of each was calculated. Besides, validation is important for the

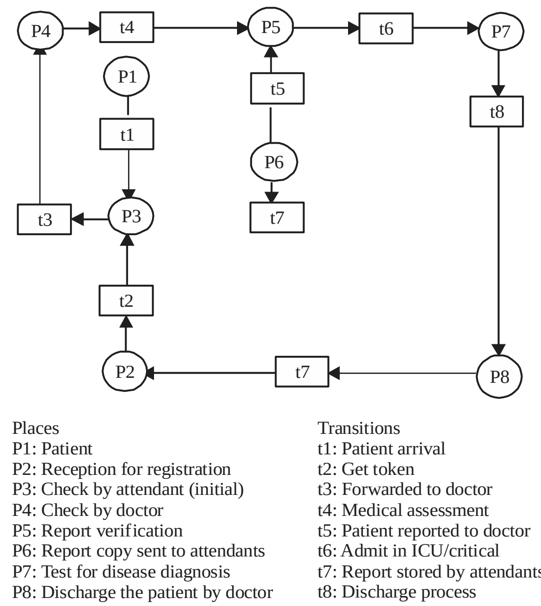

Fig. 2: Petri Net flow diagram of patient health care process

Table 1: Description of major places in the TCPN model

Places	Description
Next patient	Model entry of new patient
Waiting patients	Model list of patient waiting to be served by the medical attendants
Busy	Number of attendant(s) busy in the centre
Free	Number of attendant(s) free in the centre
pwfdasse	Patient waiting for doctor in a queue
Treatment room	Patient in the treatment room
Doctor	Number of doctor(s) available for service
End	Indicate end of treatment
Admit	Indicate inpatient (IP)
Discharge	Indicate outpatient (OP)
Attendant	Medical attendants for medical service

Table 2: Description of major transitions in the model

Transition	Description
Arrival patient	Execution of this transition models arrival of new patient
Start service	Execution of this transition models start of service by medical attendant(s)
Finished	Execution of this transition models end of service by medical attendant
Examination room	A substitution transition
Start assessing	Execution of this transition models start of service by doctor(s)
End treatment	Execution of this transition models of end of service by the doctor(s)
Critical patient	Execution of this transition models of list of critical patients
Non critical patient	Execution of this transition models of list of non-critical patients (ready for discharge)

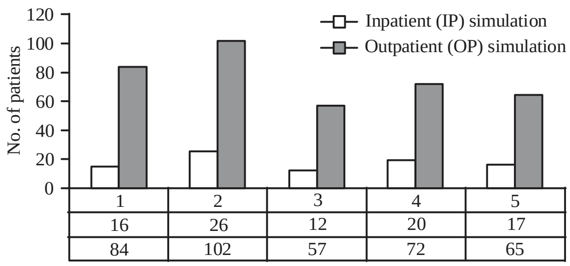

Fig. 3: Comparison in between the inpatient and outpatient's simulation of the TPCN model for 5 days

Table 3: Simulation output of the TPCN model for 5 days

Days	Inpatient (IP) simulation	Outpatient (OP) simulation
1	16	84
2	26	102
3	12	57
4	20	72
5	17	65

correctness and credibility of the model[136-139]. Validation is to determine the model which will be a representation of the real system.

Thus, the proposed timed colored Petri Net model was validated by comparing the output of the simulation model (i.e., number of inpatients and number of outpatients) for five consecutive days with the number of inpatients and of outpatients (Table 3) of the actual system.

EXPERIMENTAL ANALYSIS

Figure 2 shows the developed TCPN simulation model of patient care flow processes of the patient health centre under consideration. Figure 2 depicts the main page which models the arrival of patients and process at the medical record area of the case study. The following, is the depicts a subnet layer (Treatment sub-module) of the main model. It models operation in the examination room of the health centre under study.

From our data, entry of each patient to the health centre is modelled by a token on the place next patient. This place has the color set UNIT and the color set UNIT is defined to be equal to unit timed type as depicted in the declarations block of the developed model. The color set UNIT is used to model arrival time of patient based on the time stamp such as @++Day1AT() attached to the arc that runs from transition In it to place Next patient. Based on the evaluation of the distribution expression: ~0.5+weibull (12.3, 0.964), function Day1AT() is used to generate the arrival time of new patient into the system. From Fig. 3, color set patient type is used to represent types of patient entering the patient health centre. It is enumerated type of Critical Patient (CP) and Non-Critical Patient (NCP). The place waiting patients has the color set Patients defined to be set of list patient. The color set patients is used to model the queue of patients to be attended to. The color set patient models a patient as a record consisting of two fields. The first field denoted with PatientType is of type PatientType and represents type of the patient. Second field is denoted with the title AT is of type real and represents arrival time of a patient. The color set Attendant×Patient is a product color set defined as product Attendant×Patient timed. This color set is used to represent the attendant when he/she is busy serving patient. Also, the color set Doctor×Patient is to represent the doctor when he/she is busy treating patient. The function Day1AT() uses weibull distribution with the expression: ~0.5+weibull(12.3, 0.964) to generate arrival times of patients for day 1. This distribution is used instead of lognormal distribution because currently CPN tools (Version 4.0.1) does not support lognormal distribution. The place waiting patients and the place pwfdasse are used to model the queue of patients at the registration counter unit and examination room respectively. The single token on each of the places waiting patients and the place pwfdasse represents the queue of patients. In the initial marking the lists are empty. The places free and busy are used to represent the status of the medical attendant. A token on the place free indicates that the medical attendant is not serving a patient at that time. The parameter hospref_no_of_attendant = 5 and function fun initAttendants() = (!num_of_attendants)' attendant in Fig. 4, show that there are 4 tokens on the place free in the initial marking. A token on the place busy indicates that the medical attendant is busy attending to a patient and the value of the token indicates which patient is being processed. The initial marking of busy is empty.

The medical attendant can start attending to patient (transition start service), if the medical attendant is free and if there is at least one patient in the queue of patients (patient::patients on the arc from place waiting patients to transition start service).

COMPARATIVE STUDY OF RESEARCH WORKS

Here, compare with several research on the basis of mining loops, hidden tasks, delta analysis, visualizing, process rediscovery, duplication tasks, noise and concurrent processes (Table 4).

More recently, to deal with less structured, i.e., very diverse or flexible processes, dynamically adaptive process simplification algorithms have been proposed[67]. The approach demonstrates that for some subclasses, it is possible to

```
▽ (*------ TPCN model developed by Tamizharasan------*)
▷ Toolsbox
▷ HELP
▷ Options
▽ Newproposed.cpn
  Step: 0
  Time: 0:0
▷ Options
▷ History
▽ Standard declarations
▽ COLOR SET and VARIA
▷ colset UNIT
▷ colset INT = int;
▷ colset REAL = real;
▷ var timeotime
▽ colset PatientType = with CP[MIRA] Timotime;
▽ colset Patient = record PatientType:PatientType*AT:REAL timeotime;
▽ var Patient:Patient;
▽ colset Patients = list Patients;
▽ var Patients:Patients;
▽ colset Attendant = with Attendant;
▽ var atten = Attendant;
▽ colset Attendant×Patient = product Attendant*Patient timeotime;
▽ colset CONSU  = with CONSU;
▽ var doctor CONSU;
▽ colset Dosctor×Patient = product CONSU*Patient timeotime;
▽ (*-------Parameter Declaration-------*)
▽ PARAMETERS
▽ hospref_no_of_attendant = 5;
▽ hospref_no_of_doctor = 3;
▽ hospref_no_of_nurses = 4
▽ (*---------------Function of TPCN---------------*)
▽ funDAY1on()=~0.5+weibull(12.3, 0.964);
▽ funDAY2on()=~0.5+57.58*beta(0.52, 1.84);
▽ funDAY3on()=~0.5+42.23*beta(0.64, 1.85);
▽ funDAY4on()=~0.5+40.56*beta(0.75, 1.95);
▽ funDAY5on()=~0.5+weibull(8.94, 0.865);
▽ fun model Time()=
  Model Time.time();
  Fun newPatient()=
        {patientType = if uniform (0.0, 1.0)<= 0.25 CP else NCP,  AT = modelTime()}
      ▽ Fun initAttendants() = (!num of attendants) attendant;
      ▽ Fun initdoctors() = (!num_of_doctors)’ DOC
▽ Monitors
  My TPCN
```

Fig. 4: Declarations for the TCPN model of the CNP

Table 4: Research works dealing with process mining issues

Research works	Mining loops	Hidden tasks	Delta analysis	Non-free choice constructs	Visualizing results	Heterogeneous data source	Local/global search	Process rediscovery	Different perspectives	Duplicate tasks	Noise	Concurrent processes
Greco *et al.*[75]							✂					
Cook and Wolf[28]												✂
Golani and Pinter[70]												✂
Cook and Wolf[29]					✂						✂	
De Medeiros *et al.*[44]	✂				✂							✂
De Medeiros *et al.*[41]	✂										✂	
Gaaloul and Godart[68]											✂	✂
Agrawal *et al.*[21]										✂	✂	
De Medeiros *et al.*[121]		✂		✂							✂	
Cook and Wolf[30]												✂
Herbst[91]	✂				✂					✂		
Dustdar *et al.*[53]					✂							
Hammori *et al.*[81]										✂		
Cook *et al.*[32]											✂	✂

discover the accurate workflow model using α-algorithm. In another work, an extended version of α-algorithm is used to include the timing information.

CONCLUSION

Much study has to be done in put on the mining and synthesis algorithms to different document management systems in different application areas and making practical assessment of them both in the area of business process management and software process engineering. Since this method is also pertinent to the area of mining the activity logs, in the future, it should also compare it to the existing approaches in this area. This study aims at making the first step from the well-developed theory of Petri Net synthesis to the practically relevant research domain of process mining.

REFERENCES

1. Chen, Z., F. Lin, H. Liu, Y. Liu, W.Y. Ma and L. Wenyin, 2002. User intention modeling in web applications using data mining. World Wide Web, 5: 181-191.

2. Bass, L., P. Clements and R. Kazman, 2003. Software Architecture in Practice. 2nd Edn., Addison-Wesley Publishing Co., Boston, Massachusetts, ISBN: 0-321-15495-9.
3. Fishbein, M. and I. Ajzen, 1975. Belief, Attitude, Intention and Behavior: An Introduction to Theory and Research. 1st Edn., Addison-Wesley, Reading, MA., USA., ISBN-13: 9780201020892, Pages: 578.
4. Akkermans, H., 2006. Ontology Engineering, Scientific Method and the Research Agenda. In: Managing Knowledge in a World of Networks, Akkermans, H. and J. Gordijn (Eds.). Springer, New York, pp: 112-125.
5. Amyot, D., J. Horkoff, D. Gross and G. Mussbacher, 2009. A lightweight GRL Profile for I* Modeling. In: Advances in Conceptual Modeling-Challenging Perspectives, Heuser, C.A. and G. Pernul (Eds.). Springer, New York, USA., ISBN: 9783642049477, pp: 254-264.
6. Arbaoui, S. and F. Oquendo, 1994. Goal Oriented VS. Activity oriented Process Modelling and Enactment: Issues and Perspectives. In: Software Process Technology, Warboys, B.C. (Ed.). Springer, New York, USA., ISBN: 9783540483267, pp: 171-176.
7. Ashkan, A., C.L.A. Clarke, E. Agichtein and Q. Guo, 2009. Classifying and Characterizing Query Intent. In: Advances in Information Retrieval, Boughanem, M., C. Berrut, J. Mothe and C. Soule-Dupuy (Eds.). Springer, New York, USA., ISBN: 9783642009587, pp: 578-586.
8. Assar, S., C.B. Achour and S. Si-Said, 2000. Un modele pour la specification des processus d'analyse des systemes d'Information. In INFORSID, pp: 287-301.
9. Baeza-Yates, R., 2005. Applications of Web Query Mining. In: Advances in Information Retrieval, Losada, D.E. and J.M. Fernandez-Luna (Eds.). Springer, New York, USA., ISBN: 9783540318651, pp: 7-22.
10. Baeza-Yates, R., L. Calderon-Benavides and C. Gonzalez-Caro, 2006. The Intention Behind Web Queries. In: String Processing and Information Retrieval, Crestani, F., P. Ferragina and M. Sanderson (Eds.). Springer, New York, USA., ISBN: 9783540457756, pp: 98-109.
11. Barrios, J. and S. Nurcan, 2004. Model Driven Architectures for Enterprise Information Systems. In: Advanced Information Systems Engineering, Persson, A. and J. Stirna (Eds.). Springer, New York, USA., ISBN: 9783540259756, pp: 3-19.
12. Baum, L.E. and T. Petrie, 1966. Statistical inference for probabilistic functions of finite state Markov chains. Ann. Math. Stat., 37: 1554-1563.
13. Baum, L.E., T. Petrie, G. Soules and N. Weiss, 1970. A maximization technique occurring in the statistical analysis of probabilistic functions of Markov Chains. Ann. Math. Stat., 41: 164-171.
14. Boehm, B.W., 1988. A spiral model of software development and enhancement. Computer, 21: 61-72.
15. Bratman, M.E., 1999. Intention, Plans and Practical Reason. Cambridge University Press, Cambridge, UK., ISBN: 9781575861920, Pages: 200.
16. Bresciani, P., A. Perini, P. Giorgini, F. Giunchiglia and J. Mylopoulos, 2004. Tropos: An agent-oriented software development methodology. Autonomous Agents Multi-Agent Syst., 8: 203-236.
17. Beck, K., M. Beedle, A. van Bennekum, A. Cockburn and W. Cunningham *et al*., 2001. Manifesto for agile software development. http://agilemanifesto.org/.
18. Bengio, Y., 2009. Learning deep architectures for AI. Found. Trends Mach. Learn., 2: 1-127.
19. Biermann, A.W. and J.A. Feldman, 1972. On the synthesis of finite-state machines from samples of their behavior. IEEE Trans. Comput., C-21: 592-597.
20. Carmona, J., J. Cortadella and M. Kishinevsky, 2008. A region-based algorithm for discovering Petri nets from event logs. Proceedings of the 6th International Conference on Business Process Management, September 2-4, 2008, Milan, Italy, pp: 358-373.
21. Agrawal, R., D. Gunopulos and F. Leymann, 1998. Mining process models from workflow logs. Proceedings of the 6th International Conference on Extending Database Technology, March 23-27, 1998, Valencia, Spain, pp: 469-483.
22. Chen, K.C.W. and D.Y.Y. Yun, 2003. Discovering Process Models from Execution History by Graph Matching. In: Intelligent Data Engineering and Automated Learning, Liu, J., Y.M. Cheung and H. Yin (Eds.). Springer, New York, USA., ISBN: 9783540450801, pp: 887-892.
23. Chow, T.S., 1978. Testing software design modeled by finite-state machines. IEEE Trans. Software Eng., SE-4: 178-187.
24. Christie, B., 1981. Face to File Communication: A Psychological Approach to Information Systems. John Wiley and Sons, Inc., New York, USA., ISBN-13: 978-0471279396, Pages: 318.
25. Chulef, A.S., S.J. Read and D.A. Walsh, 2001. A hierarchical taxonomy of human goals. Motivation Emotion, 25: 191-232.
26. Clauzel, D., K. Sehaba and Y. Prie, 2009. Modelling and visualising traces for reflexivity in synchronous collaborative systems. Proceedings of the International Conference on Intelligent Networking and Collaborative Systems, November 4-6, 2009, Barcelona, Spain, pp: 16-23.
27. Conklin, J. and M.L. Begeman, 1989. gIBIS: A tool for all reasons. J. Am. Soc. Inform. Sci., 40: 200-213.
28. Cook, J.E. and A.L. Wolf, 1995. Automating process discovery through event-data analysis. Proceedings of the 17th International Conference on Software Engineering, April 24-28, 1995, Seattle, WA., USA., pp: 73-82.
29. Cook, J.E. and A.L. Wolf, 1998. Discovering models of software processes from event-based data. ACM Trans. Software Eng. Methodol., 7: 215-249.

30. Cook, J.E. and A.L. Wolf, 1998. Event-based detection of concurrency. Proceedings of the 6th ACM SIGSOFT International Symposium on Foundations of Software Engineering, Volume 23, November 3-5, 1998, Florida, USA., pp: 35-45.
31. Cook, J.E. and A.L. Wolf, 1999. Software process validation: Quantitatively measuring the correspondence of a process to a model. ACM Trans. Software Eng. Methodol., 8: 147-176.
32. Cook, J.E., Z. Du, C. Liu and A.L. Wolf, 2004. Discovering models of behavior for concurrent workflows. Comput. Ind., 53: 297-319.
33. Curtis, B., H. Krasner and N. Iscoe, 1988. A field study of the software design process for large systems. Commun. ACM, 31: 1268-1287.
34. Curtis, B., M.I. Kellner and J. Over, 1992. Process modeling. Commun. ACM, 35: 75-90.
35. Dardenne, A., A. Van Lamsweerde and S. Fickas, 1993. Goal-directed requirements acquisition. Sci. Comput. Program., 20: 3-50.
36. Das, S. and M.C. Mozer, 1994. A unified gradient-descent/clustering architecture for finite state machine induction. Neural Inform. Process. Syst., 6: 19-26.
37. Das, S. and M.C. Mozer, 1994. A Unified Gradient-Descent/Clustering Architecture for Finite State Machine Induction. In: Neural Information Processing Systems 1993, Volume 6, Cowan, J.D., G. Tesauro and J. Alspector (Eds.). Morgan Kaufmann, USA., pp: 19-26.
38. Datta, A., 1998. Automating the discovery of as-is business process models: Probabilistic and algorithmic approaches. Inform. Syst. Res., 9: 275-301.
39. Davis, F.D., R.P. Bagozzi and P.R. Warshaw, 1989. User acceptance of computer technology: A comparison of two theoretical models. Manage. Sci., 35: 982-1003.
40. De Medeiros, A.K.A., W.M.P. van der Aalst and A.J.M.M. Weijters, 2003. Workflow Mining: Current Status and Future Directions. In: On the Move to Meaningful Internet Systems 2003: CoopIS, DOA and ODBASE, Meersman, R., Z. Tari and D.C. Schmidt (Eds.). Springer, New York, USA., ISBN: 9783540399643, pp: 389-406.
41. De Medeiros, A.K.A., B.F. van Dongen, W.M.P. van der Aalst and A.J.M.M. Weijters, 2004. Process mining: Extending the α-algorithm to mine short loops. BETA Working Paper Series No. WP 113, Eindhoven University of Technology, Eindhoven.
42. De Medeiros, A.K.A. and C.W. Gunther, 2005. Process mining: Using CPN tools to create test logs for mining algorithms. Proceedings of the 6th Workshop and Tutorial on Practical Use of Coloured Petri Nets and the CPN Tools, October 24-26, 2005, Aarhus, Denmark.
43. Van Der Aalst, W.M., A.A. de Medeiros and A.J.M.M. Weijters, 2005. Genetic Process Mining. In: Applications and Theory of Petri Nets, Gianfranco, C. and D. Philippe (Eds.). Springer Science and Business Media, New York, pp: 48-69.
44. De Medeiros, A.K.A., B.F. van Dongen, W.M.P. van der Aalst and A.J.M.M. Weijters, 2005. Process Mining for Ubiquitous Mobile Systems: An Overview and a Concrete Algorithm. In: Ubiquitous Mobile Information and Collaboration Systems, Baresi, L., S. Dustdar, H.C. Gall and M. Matera (Eds.). Springer, New York, USA., ISBN: 9783540301882, pp: 151-165.
45. De Medeiros, A.K.A., A.J.M.M. Weijters and W.M.P.van der Aalst, 2006. Genetic Process Mining: A Basic Approach and Its Challenges. In: Business Process Management Workshops, Bussler, C.J. and A. Haller (Eds.). Springer, Berlin Heidelberg, ISBN: 978-3-540-32595-6, pp: 203-215.
46. De Medeiros, A.K.A., C. Pedrinaci, W.M.P. van der Aalst, J. Domingue and M. Song *et al*., 2007. An Outlook on Semantic Business Process Mining and Monitoring. In: On the Move to Meaningful Internet Systems 2007: OTM 2007 Workshops, Meersman, R., Z. Tari and P. Herrero (Eds.). Springer, New York, USA., ISBN: 9783540768906, pp: 1244-1255.
47. Delorenzi, M. and T. Speed, 2002. An HMM model for coiled-coil domains and a comparison with PSSM-based predictions. Bioinformatics, 18: 617-625.
48. Dempster, A.P., N.M. Laird and D.B. Rubin, 1977. Maximum likelihood from incomplete data via the EM algorithm. J. R. Stat. Soc. Ser. B (Methodol.), 39: 1-38.
49. Deneckere, R. and E. Kornyshova, 2010. Process Line Configuration: An Indicator-Based Guidance of the Intentional Model MAP. In: Enterprise, Business-Process and Information Systems Modeling, Bider, I., T. Halpin, J. Krogstie, S. Nurcan, E. Proper, R. Schmidt and R. Ukor (Eds.). Springer, New York, ISBN-13: 9783642130519, pp: 327-339.
50. Desel, J., 1995. Free-Choice Petri Nets. Vol. 40, Cambridge University Press, Cambridge.
51. Dietz, J.L.G. and A. Albani, 2005. Basic notions regarding business processes and supporting information systems. Requirements Eng., 10: 175-183.
52. Dik, S.C., 1989. The Theory of Functional Grammar: The Structure of the Clause. Volume 1, Foris Publications, USA.
53. Dustdar, S., T. Hoffmann and W. van der Aalst, 2005. Mining of ad-hoc business processes with TeamLog. Data Knowledge Eng., 55: 129-158.
54. Eclipse, 2013. Filtered UDC data 2013. The Eclipse Foundation, Canada.
55. Eder, J.E., G.E. Olivotto and W. Gruber, 2002. A data warehouse for workflow logs. Proceedings of the Engineering and Deployment of Cooperative Information Systems, September 17-20, 2002 Springer, New York, Pages: 1-15.
56. Etien, A., 2006. Ingenierie de l'alignement: Concepts, modeles et processus: La methode ACEM pour l'alignement d'un systeme d'information aux processus d'entreprise. Ph.D. Thesis, Universite Pantheon-Sorbonne, Paris.

57. EUT., 2013. Process mining. Eindhoven University of Technology, Netherlands.
58. Feather, M.S., 1987. Language support for the specification and development of composite systems. ACM Trans. Program. Lang. Syst., 9: 198-234.
59. Feiler, P.H. and W.S. Humphrey, 1993. Software process development and enactment: Concepts and definitions. Proceedings of the 2nd International Conference on Continuous Software Process Improvement, February 25-26, 1993, Berlin, Germany, Pages: 28-40.
60. Fernstrom, C. and L. Ohlsson, 1991. Integration needs in process enacted environments. Proceedings of the 1st International Conference on the Software Process, October 21-26, 1991, England, Pages: 142-158.
61. Fickas, S. and B.R. Helm, 1992. Knowledge representation and reasoning in the design of composite systems. IEEE Trans. Software Eng., 18: 470-482.
62. Fillmore, C.J., 1967. The case for case. Proceedings of the Texas Symposium, on Language Universals, April 13-15, 1967, Holt, Rinehart and Winston.
63. Fischer, M., 2008. ARIS process performance manager. Proceedings of the 14th GI/ITG Conference on Measurement, Modelling and Evaluation of Computer and Communication Systems, March 31-April 2, 2008, Dortmund, Germany, pp: 1-3.
64. Forney, G.D., 1973. The viterbi algorithm. Proc. IEEE, 61: 268-278.
65. Fowler, M., 1999. Refactoring: Improving the Design of Existing Code. Addison-Wesley, New York, USA., ISBN-13: 9780201485677, Pages: 431.
66. Friedman, N., D. Geiger and M. Goldszmidt, 1997. Bayesian network classifiers. Mach. Learn., 29: 131-163.
67. Gunther, C.W. and W.M.P. Van der Aalst, 2007. Fuzzy mining-Adaptive process simplification based on multi-perspective metrics. Proceedings of the 5th International Conference on Business Process Management, September 24-28, 2007, Brisbane, Australia, pp: 328-343.
68. Gaaloul, W. and C. Godart, 2005. Mining workflow recovery from event based logs. Proceedings of the 3rd International Conference on Business Process Management, September 5-8, 2005, Nancy, France, pp: 169-185.
69. Gales, M.J.F., 1998. Maximum likelihood linear transformations for HMM-based speech recognition. Comput. Speech Lang., 12: 75-98.
70. Golani, M. and S.S. Pinter, 2003. Generating a Process Model from a Process Audit Log. In: Business Process Management, Van Der Aalst, W.M.P. and M. Weske (Eds.). Springer, New York, USA., ISBM: 9783540448952, pp: 136-151.
71. Gilks, W.R., S. Richardson and D.J. Spiegelhalter, 1996. Markov Chain Monte Carlo in Practice. Chapman and Hall, London, UK., ISBN-13: 9780412055515, Pages: 486.
72. Goutte, C. and E. Gaussier, 2005. A Probabilistic Interpretation of Precision, Recall and F-Score, with Implication for Evaluation. In: Advances in Information Retrieval, Losada, D.E. and J.M. Fernandez-Luna (Eds.). Springer, New York, USA., ISBN: 9783540318651, pp: 345-359.
73. Gove, P.B., 1981. Webster's Third New International Dictionary of the English Language, Unabridged. Merriam-Webster, New York, USA.
74. Gray, W.D., B.E. John and M.E. Atwood, 1992. The precis of project ernestine or an overview of a validation of GOMS. Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, May 3-7, 1992, Monterey, CA, USA., pp: 307-312.
75. Greco, G., A. Guzzo, L. Pontieri and D. Sacca, 2004. Mining Expressive Process Models by Clustering Workflow Traces. In: Advances in Knowledge Discovery and Data Mining, Dai, H., R. Srikant and C. Zhang (Eds.). Springer, New York, USA., ISBN: 9783540247753, pp: 52-62.
76. Greco, G., A. Guzzo, G. Manco and D. Sacca, 2005. Mining and reasoning on workflows. IEEE Trans. Knowledge Data Eng., 17: 519-534.
77. OMG., 2011. Business Process Model and Notation (BPMN), version 2.0. OMG Document No. formal/2011-01-03, Object Management Group, January, 2011.
78. OMG., 2013. Business process model and notation. Version 2.0.2. December 2013, http://www.omg.org/spec/BPMN/2.0.2/.
79. Gruber, T.R., 1995. Toward principles for the design of ontologies used for knowledge sharing? Int. J. Hum.-Comput. Stud., 43: 907-928.
80. Hammori, M., J. Herbst and N. Kleiner, 2004. Interactive Workflow Mining. In: Business Process Management, Desel, J., B. Pernici and M. Weske (Eds.). Springer, New York, USA., ISBN: 9783540259701, pp: 211-216.
81. Harel, D., 1987. Statecharts: A visual formalism for complex systems. Sci. Comput. Program., 8: 231-274.
82. Hartigan, J.A. and M.A. Wong, 1979. Algorithm AS 136: A K-means clustering algorithm. J. R. Stat. Soc. Ser. C (Applied Stat.), 28: 100-108.
83. Hashemi, R., A. Bahrami, J. LaPlant and K. Thurber, 2008. Discovery of intent through the analysis of visited sites. Proceedings of the 2008 International Conference on Information and Knowledge Engineering, July 14-17, 2008, Las Vegas, Nevada, pp: 417-422.
84. Hassine, I., D. Rieu, F. Bounaas and O. Seghrouchni, 2002. Symphony: A conceptual model based on business components. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 3, October 6-9, 2002, Tunisia.
85. Hayashi, M., 2003. Hidden markov models to identify pilot instrument scanning and attention patterns. Proceedings of the IEEE International Conference on Systems, Man and Cybernetics, Volume 3, October 5-8, 2003, Cambridge, MA., USA., pp: 2889-2896.

86. Henderson-Sellers, B. and J.M. Edwards, 1990. The object-oriented systems life cycle. ACM Commun., 33: 142-159.
87. Herbst, J. and D. Karagiannis, 1998. Integrating machine learning and workflow management to support acquisition and adaptation of workflow models. Proceedings of the 9th International Workshop on Database and Expert Systems Applications, August 25-28, 1998, Vienna, Austria, pp: 745-752.
88. Herbst, J. and D. Karagiannis, 1999. An inductive approach to the acquisition and adaptation of workflow models. Proceedings of the International Joint Conference on Artificial Intelligence, Volume 99, August 1999, Stockholm, Sweden, pp: 52-57.
89. Herbst, J., 2000. Dealing with concurrency in workflow induction. Proceedings of the 7th European Concurrent Engineering Conference, April 17-19, 2000, Leicester, UK.
90. Herbst, J., 2000. A machine learning approach to workflow management. Proceedings of the 11th European Conference on Machine Learning, May 31-June 2, 2000, Barcelona, Catalonia, Spain, pp: 183-194.
91. Herbst, J., 2004. Ein Induktiver Ansatz zur Akquisition und Adaption von Workflow-Modellen. Tenea Verlag Ltd., Berlin, Germany.
92. Herbst, J. and D. Karagiannis, 2004. Workflow mining with InWoLvE. Comput. Ind., 53: 245-264.
93. Hoey, J. and J.J. Little, 2007. Value-directed human behavior analysis from video using partially observable Markov decision processes. IEEE Trans. Pattern Anal. Mach. Intell., 29: 1118-1132.
94. Jensen, F.V., 1996. An Introduction To Bayesian Networks. Vol. 210, Taylor and Francis, London, ISBN: 9781857283327, Pages: 188.
95. Jensen, K., 1996. Coloured Petri Nets: Basic Concepts, Analysis Methods and Practical Use. Vol. 1, Springer, New York, USA.
96. Jensen, K., L.M. Kristensen and L. Wells, 2007. Coloured petri nets and CPN tools for modelling and validation of concurrent systems. Int. J. Software Tools Technol. Transfer, 9: 213-254.
97. Jethava, V., L. Calderon-Benavides, R. Baeza-Yates, C. Bhattacharyya and D. Dubhashi, 2011. Scalable multi-dimensional user intent identification using tree structured distributions. Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information Retrieval, July 24-28, 2011, Beijing, China, pp: 395-404.
98. Juang, B.H. and L.R. Rabiner, 1991. Hidden Markov models for speech recognition. Technometrics, 33: 251-272.
99. Kaabi, R.S. and C. Souveyet, 2007. Capturing intentional services with business process maps. Proceedings of the 1st International Conference on Research Challenges in Information Science, April 23-26, 2007, Ouarzazate, Morocco, pp: 309-318.
100. Kathuria, A., B.J. Jansen, C. Hafernik and A. Spink, 2010. Classifying the user intent of web queries using k-means clustering. Internet Res., 20: 563-581.
101. Kelley, R., A. Tavakkoli, C. King, M. Nicolescu, M. Nicolescu and G. Bebis, 2008. Understanding human intentions via hidden markov models in autonomous mobile robots. Proceedings of the 3rd ACM/IEEE International Conference on Human Robot Interaction, March 12-15, 2008, Netherlands, pp: 367-374.
102. Khodabandelou, G., C. Hug, R. Deneckere and C. Salinesi, 2013. Supervised intentional process models discovery using hidden Markov models. Proceedings of the IEEE 7th International Conference on Research Challenges in Information Science, May 29-31, 2013, Paris, pp: 1-11.
103. Kil, D.H. and F.B. Shin, 1997. Pattern Recognition and Prediction with Applications to Signal Processing. Vol. 16, American Inst. of Physics, USA., Page: 418.
104. Kruchten, P., 2004. The Rational Unified Process: An Introduction. Addison-Wesley Professional, USA., ISBN: 9780321197702, Pages: 310.
105. Kumar, N. and I. Benbasat, 2006. Research note: The influence of recommendations and consumer reviews on evaluations of websites. Inform. Syst. Res., 17: 425-439.
106. Kunz, W. and H.W.J. Rittel, 1970. Issues as Elements of Information Systems. Vol. 131, Institute of Urban and Regional Development, California, Pages: 16.
107. Laflaquiere, J., L.S. Settouti, Y. Prie and A. Mille, 2006. Trace-Based Framework for Experience Management and Engineering. In: Knowledge-Based Intelligent Information and Engineering Systems, Gabrys, B., R.J. Howlett and L.C. Jain (Eds.). Springer, New York, pp: 1171-1178.
108. Lee, J., 1991. Extending the Potts and Bruns model for recording design rationale. Proceedings of the 13th International Conference on Software Engineering, May 13-16, 1991, Austin, TX., pp: 114-125.
109. Lee, S., 2012. A generic graph-based multidimensional recommendation framework and its implementations. Proceedings of the 21st International Conference Companion on World Wide Web, April 16-20, 2012, Lyon, France, pp: 161-166.
110. Li, C. and G. Biswas, 1999. Finding behavior patterns from temporal data using hidden markov model based unsupervised classification. Proceedings of the Computational Intelligence Methods and Applications, June 22-25, 1999, Rochester, NY., pp: 266-272.
111. Liu, H. and P. Singh, 2004. ConceptNet-a practical commonsense reasoning tool-kit. BT Technol. J., 22: 211-226.
112. McDermid, J. and K. Ripken, 1984. Life Cycle Support in the Ada Environment. CUP Archive, USA., ISBN: 9780521260428, Pages: 247.
113. Malcolm, N., 1967. Explaining behavior. Philos. Rev., 76: 97-104.

114. Mannila, H., H. Toivonen and A.I. Verkamo, 1997. Discovery of frequent episodes in event sequences. Data Mining Knowledge Discov., 1: 259-289.
115. Mannila, H. and D. Rusakov, 2001. Decomposition of event sequences into independent components. Proceedings of the 1st SIAM International Conference on Data Mining, April 5-7, 2001, Chicago, IL., pp: 1-17.
116. Martelli, P.L., P. Fariselli, A. Krogh and R. Casadio, 2002. A sequence-profile-based HMM for predicting and discriminating β barrel membrane proteins. Bioinformatics, 18: S46-S53.
117. Martin, J.J., 1967. Bayesian Decision Problems and Markov Chains. Wiley, New York, Pages: 202.
118. Martin, J., 1991. Rapid Application Development. Macmillan Publishing Co., USA., ISBN: 9780023767753, Pages: 788.
119. Maruster, L., W.M.P. Van der Aalst, A.J.M.M. Weijters, A. van den Bosch and W. Daelemans, 2001. Automated discovery of workflow models from hospital data. Proceedings of the 13th Belgium-Netherlands Conference on Artificial Intelligence, October 25-26, 2001, Amsterdam, pp: 183-190.
120. Maruster, L., A.J.M.M. Weijters, W.M.P. van der Aalst and A. van den Bosch, 2002. Process mining: Discovering direct successors in process logs. Proceedings of the 5th International Conference on Discovery Science, November 24-26, 2002, Lubeck, Germany, pp: 364-373.
121. De Medeiros, A.K.A., A.J.M.M. Weijters and W.M.P. van der Aalst, 2005. Using genetic algorithms to mine process models: Representation, operators and results. The Beta Research School for Operations Management and Logistics, September 5, 2005.
122. Mei, T., X.S. Hua and H.Q. Zhou, 2005. Tracking users' capture intention: A novel complementary view for home video content analysis. Proceedings of the 13th Annual ACM International Conference on Multimedia, November 6-12, 2005, Singapore, pp: 531-534.
123. Miller, G.A., 1995. WordNet: A lexical database for English. Commun. ACM, 38: 39-41.
124. Mirbel, I. and J. Ralyte, 2006. Situational method engineering: Combining assembly-based and roadmap-driven approaches. Requirements Eng., 11: 58-78.
125. Mobasher, B., R. Cooley and J. Srivastava, 2000. Automatic personalization based on web usage mining. Commun. ACM., 43: 142-151.
126. Mostow, J., 1985. Toward better models of the design process. AI Magazine, 6: 44-56.
127. Mulyar, N., M. Pesic, W.M.P. van der Aalst and M. Peleg, 2008. Declarative and Procedural Approaches for Modelling Clinical Guidelines: Addressing Flexibility Issues. In: Business Process Management Workshops, Ter Hofstede, A., B. Benatallah and H.Y. Paik (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-78237-7, pp: 335-346.
128. Murphy, K.P., 2002. Dynamic Bayesian networks: Representation, inference and learning. Ph.D. Thesis, University of California, USA.
129. Murphy-Hill, E. and A. Black, 2008. Breaking the barriers to successful refactoring. Proceedings of the ACM/IEEE 30th International Conference on Software Engineering, May 10-18, 2008, Leipzig, Germany, pp: 421-430.
130. Mylopoulos, J., L. Chung and B. Nixon, 1992. Representing and using nonfunctional requirements: A process-oriented approach. IEEE Trans. Software Eng., 18: 483-497.
131. Myung, I.J., 2003. Tutorial on maximum likelihood estimation. J. Math. Psychol., 47: 90-100.
132. Najar, S., M. Kirsch-Pinheiro and C. Souveyet, 2011. Towards semantic modeling of intentional pervasive information systems. Proceedings of the 6th International Workshop on Enhanced Web Service Technologies, September 14, 2011, Lugano, Switzerland, pp: 30-34.
133. Nurcan, S., A. Etien, R. Kaabi, I. Zoukar and C. Rolland, 2005. A strategy driven business process modelling approach. Bus. Process Manage. J., 11: 628-649.
134. OMG., 2004. Object management group. http://www.omg.org.
135. Olle, T.W., 1988. Information Systems Methodologies: A Framework for Understanding. Addison-Wesley, USA., ISBN-13: 978-0201416107, Pages: 364.
136. Outmazgin, N. and P. Soffer, 2013. Business Process Workarounds: What Can and Cannot Be Detected by Process Mining. In: Enterprise, Business-Process and Information Systems Modeling, Nurcan, S., H.A. Proper, P. Soffer, J. Krogstie, R. Schmidt, T. Halpin and I. Bider (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-642-38483-7, pp: 48-62.
137. Park, K., T. Lee, S. Jung, H. Lim and S. Nam, 2010. Extracting search intentions from web search logs. Proceedings of the 2nd International Conference on Information Technology Convergence and Services, August 11-13, 2011, Cebu, Philippine, pp: 1-6.
138. Pesic, M. and W.M.P. van der Aalst, 2006. A Declarative Approach for Flexible Business Processes Management. In: Business Process Management Workshops, Eder, J. and S. Dustdar (Eds.). Springer, Berlin, Heidelberg, ISBN: 978-3-540-38444-1, pp: 169-180.
139. Peterson, J., 1981. Petri Net Theory and the Modeling of Systems. 1st Edn., Prentice Hall, Englewood Cliffs, New Jersey, ISBN: 0136619835, pp: 288.