Constraints Resolution in Semi-Automated Course Allocation Process for Educational Institute

Muhammad Imran[1], Riaz-ul-Amin[2], Abdul Sattar Malik[2], Bakhtiar Kasi[2], Mumraiz Kasi[2]

[1]Department of Information Technology, [2]Department of Computer Science, Balochistan University of Information Technology, Engineering and Management Sciences, Quetta, Pakistan

Abstract

In this study, the problem of biasness and unfairness in course allocation mechanism is addressed. Institutions face problem when allocating courses to the faculty members that are according to their expertise and this becomes more complex when two or more than two individuals are having expertise in same area. The work done previously in this area is extended by suggesting the mechanism that reduce the human biasness and encourages the fairness to the extent possible. Mechanism suggested uses well known process mining algorithm, the Alpha. It generates process model based on software generated event logs and comparison of original process model with process mining generated process model reveals the deviations in process model which in turn helps to detect deviations in it and is founding stone for suggestive conclusion for further improvement and modification in the software process model and the software system itself. Conclusively the suggested mechanism helps to reduce human biasness in course allocation and encourages to allow fair allocation of the resources to the right person.

Key words: Process Mining; Course Allocation; Constraints resolution; Process Modeling; Deviations detection

Corresponding author's email: *mimranit@outlook.com*

INTRODUCTION

Allocation of resources in any field is considered an important task. Whether it is allocation of a task to the workers working onsite or a student being assigned its homework and assignments. Everyone desires the resources allocated to him/ her that are distributed equally, evenly and fairly so that no one gets suffered from extra burden of work and get its fair share of incentives whether in the form of money or in the form of moral support. Conclusively the objective of the resource allocation is to maintain equilibrium between resources and needs.

Many constraints are needed to be resolved while going through resource allocation process. Biasness is one of the constraints that is probably on the top of the list without which fair allocation seems like good dream being converted into nightmare due to only a single reason.

Allocation of courses to the instructors in the educational institutions is one of these scenarios where instructors from different field are allocated courses that they have to teach in any specific academic session/ semester or year, varying depending upon the policy of the institution. Each or group of faculty members may belong to different area research or interest in which he/ she may have expertise and willing to work on and do further research. Being expert in that specific subject/ area, that faculty members desires to instruct the course that is from his the domain in which is already working. Like an instructor working on a software development project most probably would like to teach the course that is from the programming domain. So this arises needs of a system that allocates course to the instructors according to their expertise and give every instructor their fair chance to get a course of their field of expertise. While many researchers have worked previously on this problem, this paper is focused on the work of (Ullah and Amin, 2016) to be more precise to our solution.

Process mining techniques allow for extracting information from event logs (Aalst, 2016). Trails of transaction logs are used for discovering models that describe processes, organizations, and products. It can also be used discover or monitor deviations in the business process model, the result of which can be further used to improve the business process model. It is done by comparison of happened events with existing models and after analysis, some conclusion is made based on the result of analysis.

There are a number of tools available in the market to be used for process mining. Probably being top of the list in open source standing. In commercial tools, Disco (pronounced Discover) by Fluxicon Inc. It is prominent tool, the institutional license of which is used throughout this work for research purposes only.
In this paper the use of process mining in course allocation software systems is proposed to detect any biasness which compromises the standard course allocation process as incorporated in software system so that every instructor may have fair chance of competing for a specific course of his expertise in case there are multiple instructors competing for a that same specific course.

Course allocation process

Course allocation process consists of two objects, courses and faculty members and in between these two mapping process is performed which maps needs (course) to the resources (faculty members). This mechanism is contrasted in figure 2.1.

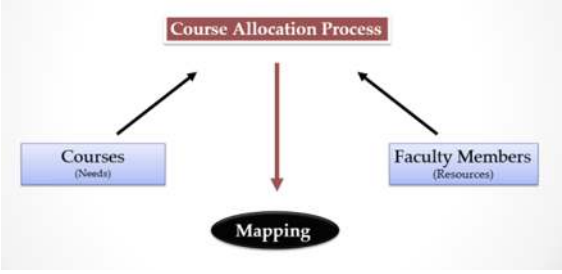

Figure 1: Course Allocation Process (CAP)

When dealing with course allocation process, there are number of constrains encountered that need to be dealt with like Course complexity, Maximum workload each faculty members is eligible for. There may be other constraints like domain restriction of faculty members which has been referred earlier. Choice or preference of the instructor being another constraint which affects the course allocation process.

MATERIALS AND METHODS

As a first step of the research, the event log data needed. As discussed in section 4, random generated data or data generated by running the simulation on the model could be taken. But to be precise and to be more accurate to the research topic, an original scenario was picked up. The system developed names CAS (Course Allocation System) for course allocation by (Ullah and Amin, 2016) was requested for the research purposes only which was granted. Now the problem was that the CAS didn't contain any mechanism for event log generation. It was needed to be incorporated. Since an SQL Server Database was being used as the data storage purposes for course allocation data in CAS, it was feasible to use that same database for event log generation on every instance of task being executed in the system. The existing database was modified to have another table names as softLog which continued the fields as LoggedBy, Time, Operation, Semester and ResExecutor. The fields LoggedBy was used as the identifier of the person executing the activity. It represented the person logged in at that specific time in the system. So any activity executed by that person was recorded in the softLog table in his name. The field's value varies by person to person logged into the system. Second, the Time field represented the timestamp of the system at which that specific activity was performed in the system. It records the read timestamp of the system. Thirdly field named Operation records the activity being performed or being executed. This records what specific task is being carried out in the system. The filed value varies depending upon the activity being carried out. The fourth field records Semester/ Session in which that task is being performed. The filed value varies depending upon the semester/ session in which the activity is being performed.
There are three basic requirements for an event log in Process Mining. It must contain a Case ID, a timestamp and resource field. In our specific scenario, the Semester field can be considered as Case ID representing different case depending upon the semester/ session. The filed Time can be considered as Timestamp which the filed LoggedBy can be considered as resource filed being as basic requirement of process mining event log data. Other fields being optional considering that if that exist in the system, they help to get more precise results. Otherwise the Case ID, Timestamp and Resource are enough for process mining analysis reasons. In the same way, the logging table, SoftLog, some extra fields were also introduced in case more precise

results be needed in case of unclear results to detect deviations. These extra fields included Course and Instructor. Course filed being used as holder of the course that is being allocated in case a manual allocation was being carried out in the system while Instructor field being holder of the name of the instructor to whom the course is being allocated in case of manual course allocation scenario. Now here arises a question that what is need of having manual course allocation process if automated system is already in place. The answer may be given as in case there are courses that do not fit the expertise of the any instructor or that are not opted by any instructor. In that specific case, it was needed to have a system in CAS to allocate that course manually providing that allocation rights to the Dean of faculty or Chairperson of the department to which instructor belongs to. Till this step, the existing system database as used in the CAS by (Ullah and Amin, 2016) was made capable of storing event logs.

Now, the logs needed to be generated by the system that will be used as base of the research. Some C# coding was done to make the front end system capable of inserting the log entries to the database table that was modified earlier. At each operations step of the CAS, the code chunk was introduced that executes database insert query as an instance of event log. Like when logging in to the system, an insert query was executed to have the database table, the name of the person being logged in to the system moreover recording the timestamp at which he or she is being logged in.

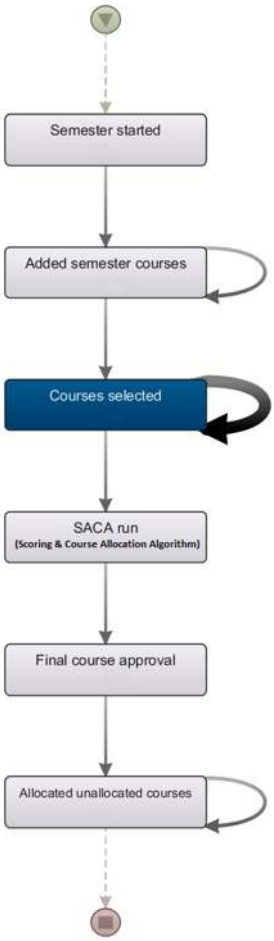

Figure 2: (a) Expected flow of Software Process Model

Taking example of the chairperson, he logs in to the system and at very first stage of the course allocation, he selects the courses offered for the current semester/ session. At each of his action in the system, as log

entry is sent to the database table softLog which holds event log containing the executor's name, time, activity being performed by him/ her and any extra fields that were kept as option. So after completing this step, the existing system by (Vanthitnen et al., 2014; Gunther et al., 2004) was made fully capable of generation of event logs.

Now to generate event logs, the CAS brought to execution. Since the system was not already implemented. So it was needed to have simulated runs of the system. Three semesters were used as test cases of our simulation named as Spring 16, Fall 16 and Spring 17. All the steps involved in course allocation system were triggered one by one for each semester. Considering one single scenario, Spring 16 in which first of all Dean starts the semester, then chairpersons select the courses for that specific semester. After that, the instructors were asked to select preferences of courses for that specific semester. After the end of specified time span for instructors to opt their courses, the system execution is handed over to the dean of the faculty again and this time he runs the CAS algorithm as developed by (Ullah and Amin, 2016). The system comes with the course allocation plan (CAP). Till this step system proposed by (Viladimir et al., 2014) works as automated. Now if any courses if not opted by any instructor or they do not fulfil the domain criterion of none of the instructor. They are kept upon as unallocated courses. Apart from that there may be instructors who could not get any courses, they are allocated courses manually by dean of the faculty or by the chairperson of the department. The same procedure is followed in the system for other two semesters, Fall 16 and Spring 17. This concludes the log generation phase in the database of CAS.

Logs generated in the system needs to be extracted in the specified format for process mining analysis to detect deviations leading towards goal of this work. Event logs were extracted from the SQL Server database into CSV form from the table SoftLog.

After extraction of the event logs, the main parameter of process mining, the event logs are ready to be analyzed by any process mining software as mentioned earlier as ProM or Fluxicon Disco.

The event logs were loaded into Fluxicon Disco. Upon loading the event logs into Disco for analysis, some configuration needs to be made to explicitly specify which filed contains case ID, which contains timestamp and which contains the recourse field being the three basic fields needed to perform process mining.

Post this step, Disco is ready to run Alpha Algorithm on the event logs to generate Process Model of the event logs loaded. In process mining terms, running alpha algorithm on event logs is referred to as Process

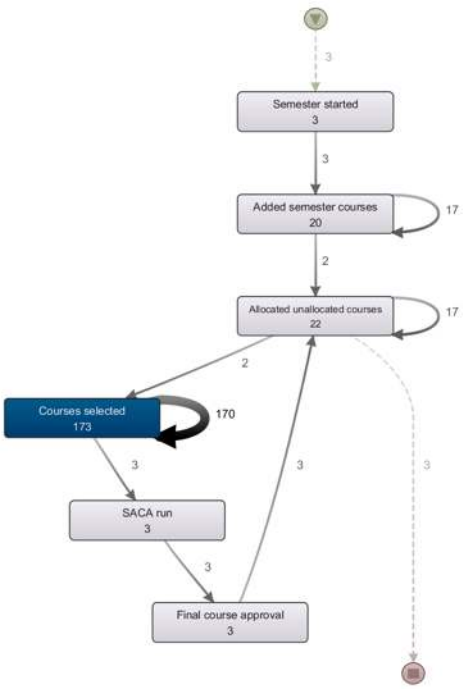

Figure 2: (b). Deviated flow of Software Process Model

Now the original process model that is expected to be followed by the software (Figure2-a) can be visually compared with the model generated by Disco using event logs generated by the software system being analyzed for biasness (Figure 2-b).

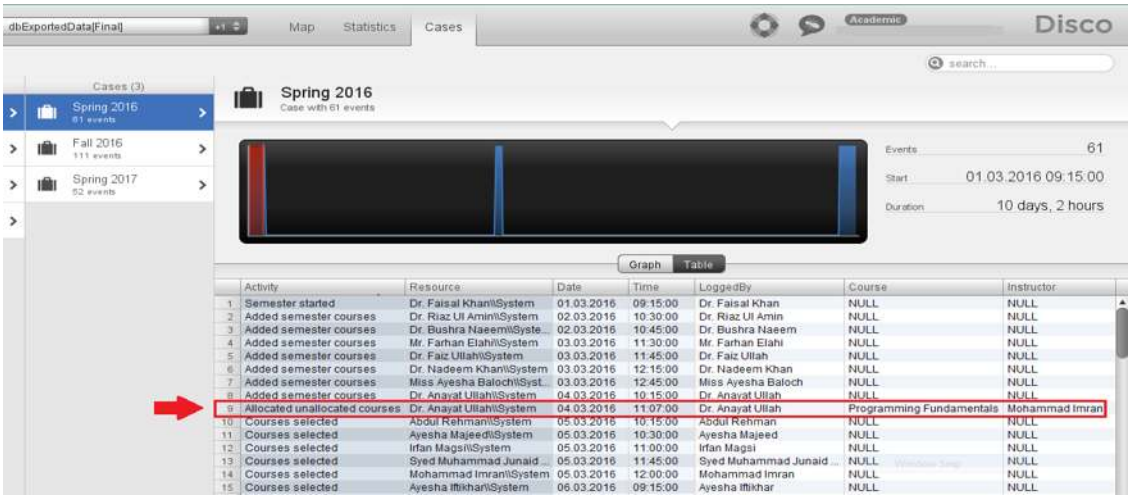

Figure 3: Detected Deviations (highlighted) Discovery so alpha algorithm is a process discovery algorithm (Aalst, 2016).

Any steps missing can be evidenced. After visually analyzing both the process models, it can be detected that if any step was missed during the course allocation or if any deviation was made from the regular process model by performing any step earlier which was needed to be performed later in the hierarchy of steps. Since both the conditions, the order of the execution of the activities and missing of steps affect the result of the course allocation system. The system is expected to exhibit human biasness that leads to the unfair allocation of the courses. Taking an example if any an instructor was allocated a course right before he/ she passes from the course allocation algorithm, it creates an unfair scenario for the rest of the instructors who pass from the course allocation algorithm by reducing their chances of getting that course considering the fact that both the course and the instructor to whom that course was allocated will not be in any competition.

RESULTS AND DISCUSSION

Upon comparing the original course allocation process model (Figure 2a) with the model generated from the Disco (Figure 2b) after running alpha algorithm on event logs of three consecutive semesters, it was noticed that one activity named “Allocate unallocated courses” which was being performed as last step of original course allocation process model, now appears to be executed before the courses are selected or being passed through Scoring and Course Allocation Algorithm (SACA), indicating the underlying improper usage of the software system or out of order execution of software system. Here it was evidenced that system under test exhibits human biasness allowing its users to bypass or to interchange the activities being performed.

To drill down further, the Cases section was analyzed that leads to visualizing the number of cases involved in that event log and activities performed in each of case. In this work, each semester was taken as case for course allocation process, making “Spring 2016”, “Fall 2016” and “Spring 2017” as our cases and multiple number of activities being involved each case. To reach the bottom of the issue where the deviated activity was performed at which time by whom, first analysis of our first case “spring 2016” was performed (Figure 3). Upon visual analysis of this, it was revealed that one instance of activity “Allocated unallocated course” was executed right before course was selected or passed through SACA. The activity revealed the name of person executing it, time at which this activity was executed and what modification done so far in the system by this activity. Moving on to the Second case of “fall 2016”, same deviation was detected in the system. The other cases revealed no such discrepancy, deviation or out of order activities.

The results that were interpreted by the analysis of the process model generated by Disco revealed that the underlying deviations that were detected in effect are impacting the fair usage of the system and also affecting the equal and fair chances of getting opted course for each faculty member. This raised notion of biasness into the system as some of the persons were getting out of the way favor leaving remaining persons as part of improper and unfair course allocation.
To validate the effectiveness of the method defined in this work, a survey was conducted. The survey was conducted at the site where this software system was expected to be used. Survey contained ten questions asking each respondent about the validity, usefulness and their satisfaction on the technique suggested in this work. 72 percent of the respondents said this work will help reduce the biasness in the system and were satisfied with the work while 17 percent of the respondents did not show their satisfaction on the system. The remaining 11 percent of the respondents remained neutral in their response.

CONCLUSION

Process mining is a fascinating field that helps organizations achieve the transparency and discrepancy in the flow of the information systems they use. Considering this work, a practical and real world scenario was taken to show in practical implementation of process mining apart from the theoretical one that helps us to understand the deviations made in the regular workflow of the software process. The detection of these deviations which raised biasness into the system can then be reported to the authorities concerned thus forcing them to revisit the existing system. Although it cannot be said that biasness from the system was completely removed but it was reduced to the extent that was focused on in this work. The use of incorporated event logs system in the course allocation software system was also suggested to reduce the future problems related to biasness into the system.

REFERENCES

- Ullah S and Amin RU. (2016). Rules driven semi-automated course allocation for education Institute.
- Aalst W. (2016). Process Mining: Data Science in Action, Springer.
- Viladimir RA, Alexy MA , Irina LA , Wil VDAP. (2014). Process Mining Can Be Applied to Software too.
- Gunther A, Alves de Medeiros AK, CW. (2004). Process Mining: Using CPN Tools to Create Test Logs for Mining Algorithms".
- Vanthienen J, Seppe KLM, Bart Baesens, Vanden Broucke (2014). Straightforward Petri netbased event log generation in ProM.