Task Allocation with BPM Tool

Romana Ferreirinha Marques
Instituto Superior Técnico
Portugal
romanaferreirinha@tecnico.ulisboa.pt

ABSTRACT

The recent evolution of business processes has given importance to the concept of resource allocation to tasks. Once understood the idea that business processes are indispensable for organizations to achieve their objectives, the need to ensure an effective allocation of resources arises. This work aims to develop a solution for an automatic allocation of tasks to resources, which considers a set of factors associated with its execution, in the context of a document management and workflow product, edoclink.

KEYWORDS

Task Allocation, Business Process, edoclink, Resource Allocation Metrics.

1 INTRODUCTION

Business processes are one of the main assets of an organization, since they have a direct impact on the attractiveness of products and services, determine tasks and responsibilities, and the revenue potential of organizations [1].

The increasing need for globalization and innovation has been driving the development of business processes. In response to this growth, a wide range of tools, techniques and methods have been developed [1].

The recent evolution of business process management has given a fair importance to resource allocation. Once the idea that business processes are indispensable for organizations to achieve their objectives was grasped, a new concern emerged: that of ensuring effective resource allocation, since it translates into a significant improvement of business processes [2].

First, it is necessary to realize that allocating human resources and material resources is different. A human resource, that is, a person, has distinct specific personal characteristics: capabilities, intelligence, experience, values, personality, and culture, which is not the case with a material resource. In order to manage human resources effectively, it is essential to realize that these factors influence how people behave at work [5].

In an organization, obstacles arise when we try to allocate an employee to a task. We do not always allocate the most suitable person for the task at hand, which means that not only is a resource being wasted, but time is being wasted and the success of the organization is being compromised [3].

The "most suitable resource" depends on the objective of the task at hand and therefore this must be taken into consideration. You may want the fastest result, the cheapest result, or the result that guarantees the best quality. The desired goal undoubtedly guides the choice of the most appropriate resource.

Thus, we can easily see that effective resource allocation is an important competitive advantage for any organization, especially for organizations where time is crucial and the amount of resources is limited [2].

The problem of this work is to develop a solution that is capable of always giving an answer, in a limited amount of time, and that this answer fulfills the SLA in question.

After the development of the solution, it is intended that it integrates the product edoclink, through the development of a stored procedure. In short, it is intended to develop a functionality that allows the automatic allocation of tasks to resources, specifying a set of criteria to be applied.

This product is an integrated document management and workflow solution from Link Consulting that supports the specification and subsequent execution of processes, as expected in a Business Process Management (BPM) solution.

2 RELATED WORK

Based on the analysis of several works related to the topic, Table 1 presents a summary of the main metrics mentioned in the works related to resource allocation.

- Time/ Performance/ Performance: duration of the execution of a given task by a given resource, that is, time that a resource is allocated to a task.
- Cost: amount of money spent for a resource to perform a given task.
- Frequency: regularity with which a resource is allocated to a task.
- Availability: indicates whether the resource is available to be allocated to a task at a given moment.
- Work volume: a resource's current amount of work.
- Experience/Specialization measures a resource's ability to perform a given task.
- Quality: measures the perception of how well a task is performed or the customer's assessment of the resource's performance of the task.

- Reliability (of each resource): the relationship between the importance of each task (through a weight) and the error rate of each resource when performing that task.

Bringing all these criteria together, it should be noted that "Reliability" is a derived metric, which can be obtained through the "Experience" and "Quality" metrics.

	Abir Ismaili-Alaoui [2]	Michael Arias [6]	Jiajie Xu [4]	Jorge Munoz-Gama [7]
Time/ Performance	X	X	X	X
Cost		X	X	X
Frequency		X		X
Availability	X			
Work volume		X		X
Experience/ Specialization		X		X
Quality		X		X
Reliability	X			

Table 1: Comparison of Used Metrics

Regarding the algorithms used, firstly, we studied Ismai-li-Alaoui's approach [2], which advocates the use of a combination of unsupervised Machine Learning Algorithms with Genetic Algorithms. Secondly, we saw the perspective of Michael Arias [6], who advocates the use of a Best Position Algorithm (BPA) for this problem. Next, we saw Jiajie Xu's proposal [4], which contemplates a set of three algorithms (Basic Allocation + Conflict Resolution + Time Constraint) that are applied depending on the objective of the organization. Finally, we analyzed Jorge Munoz-Gama's proposal [7] that, contrary to the previous ones, distinguished the concepts of Allocation and Recommendation, for which he proposed solutions based on ILP and BPA2 algorithms, respectively.

As we have seen, each of the mentioned authors uses a distinct approach to solve this problem, both in terms of metrics and algorithms. This factor leads us to believe that there is still no definitive and consensual solution to this issue.

3 PROPOSAL

3.1 Architecture Overview

The architecture elaborated for this solution is based on the The architecture elaborated for this solution is based on the development of a module that integrates the edoclink, and that is invoked by the application whenever an automatic allocation step is pending.

The development of this module includes stored procedures in SQL that incorporate all the logic of the algorithm. When we want to automatically allocate a resource to a task, edoclink calls the stored procedure DistributionStage_Allocation, which receives as input the id of a stage. This stored procedure checks if the reallocation parameter is active. If not, the stored procedure *sgc_DistributionStage_Allocate* is executed for the received task id. If it is active, this stored procedure is first allocated to all previously allocated tasks, whose execution has not yet started, and only later runs for the task you want to allocate.

In order to obtain the most suitable collaborator for the new task, the rankings of the four metrics are calculated independently, using views such as *pending_stages* or *pending_time.* A ranking is obtained for each metric, where the first ranked is, naturally, the one that best satisfies the condition. Finally, we get a final ranking, which considers the results of all metrics, as well as the weighting (%) of each one, and the result of the assignment is returned.

While the Time, Quality and Cost metrics are calculated with low volatile values that change little over the course of a day, the Availability metric relies on "current state" values. Thus, when contemplating reallocation, the allocation of each task ends up depending on the allocation of the previous task, which is why it has to be done sequentially.

In order to consider the Cost metric, it was necessary to implement a way to calculate it, since the edoclink did not allow it. In this sense, we based ourselves on the concept of Categories (Roles) and associated each one with a corresponding value (per hour). Then, we created relations between the Categories and the Users, so that it was possible to assign each user a category. These Categories are configurable in edoclink.

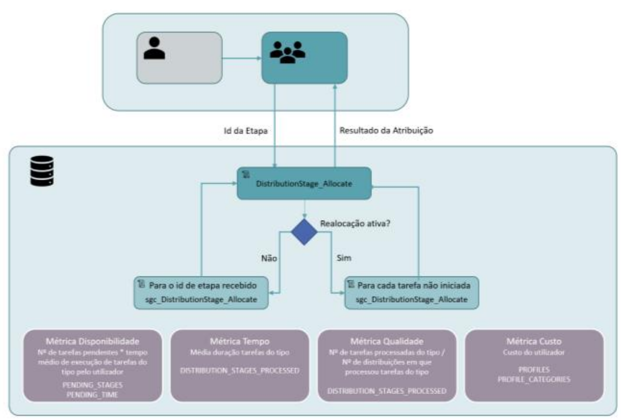

Figure 1: Overview Architecture

3.2 Algorithm Metrics

From the set of metrics analyzed in the section presented in section 2, we were able to identify the ones that made sense, given the goal we wanted to achieve.
The decision regarding the metrics to be used in the development of this resource allocation algorithm was based on the importance that they may have for the context of our problem. Naturally, we also had to take into consideration the information that the edoclink made available to us, as well as its format and restrictions.

The goal of this algorithm is to allocate resources to tasks, minimizing costs, and meeting time and quality SLAs.
To arrive at the desired algorithm, the first metric we considered was Availability. It would make no sense to be thinking about how to allocate tasks to people unavailable to execute them. So, this first concern arose to allocate tasks to the resource with the least pending work, that is, with the least number of pending tasks. However, we quickly concluded that considering only this variable would be insufficient, since not all tasks have the same degree of complexity. More complex or time-consuming tasks have naturally longer average execution times. Thus, we decided to consider the average execution time per task type per employee. Consequently, the need arose to define the task type, a concept that we will address later.

$$Availability = min \sum_{1}^{n} (average\ execution\ time\ ti * nr.pending\ tasks\ ti)$$

where ti = task type and n = number of pending tasks

In effect, to find out which employee is the most available, we considered the sum of the average execution times by task type per employee of the pending tasks. The employee who obtains the lowest value will be considered the most available. This weighting allows us to ensure not only that the employee assigned to the task is effectively available to perform it, but also seeks to maintain a better balance in task distribution, even knowing that not all tasks have the same degree of complexity.

After overcoming the first obstacle, a new concern arose, that of ensuring that tasks were successfully performed, guaranteeing quality. Parallel to this concern, another one arose: was quality necessarily related to time?

We decided to treat these two criteria separately. First, consider Time, that is, the duration of execution of a task by a given resource. This metric was intended to ensure that the resource assigned to the task is the one that performs it the fastest. In situations where we are interested in having an active, almost instantaneous response, it is necessary to ensure that the algorithm being developed can meet that expectation. To calculate this measure, we considered variables such as the average execution time per type of task per employee.

$$Time = \sum_{1}^{n} \left(\begin{matrix} average\ execution\ time\ per\ type\ of \\ task\ per\ employee\ ti \end{matrix} \right)$$

where ti = task type

Once the fastest response is guaranteed, we are able to measure Quality. This metric attributes a task to the resource that usually performs it most successfully, that is, with the lowest repetition rate. Its relevance falls on situations where we want to ensure a successful performance:

$$Quality = \min(repetition\ rate\ Cx) = \min\ ((executed\ tasks\ ti) / (\#Distributions\ where\ tasks\ executed\ ti))$$

where ti = task type, n = number of pending tasks and Cx = employee

So far, we have been able to ensure that we select the fastest resource that is available and performs the task with the greatest success. However, this evaluation is not sufficient, as it still leads to "tie" situations where we cannot choose a resource. To answer this problem, we had to refine our search to make it more complete. In this sequence of thought comes the Cost metric, which intends to assign a task to the element that ensures the cheapest execution. Therefore, the calculation of this metric is done through the value per hour that each employee costs the company. Naturally, the lowest value corresponds to the employee who performs the task the cheapest.

$$Cost = \min\ (\mathrm{cost\ Cx})$$

where Cx = employee

3.3 Algorithm

Knowing that we intended to implement a metric-based algorithm, it became necessary to map them with existing edoclink concepts. In this way, whenever a "task" is considered, it relates to the existing "step" concept. The same applies to the "collaborator" concept, commonly referred to as the "edoclink user".

Having determined which metrics to consider in this solution and clarified the edoclink concepts present in this work, it became necessary to define the functioning of the algorithm, whose objective was to improve the allocation of resources to tasks, done manually until now. Since it is an algorithm based on rankings and metrics, the possibility of considering simplex type algorithms became remote. Although this type of algorithms guarantees a lower execution complexity, it does not ensure stability in universes that are constantly changing. Since our universe is a set of pending tasks, which can change at every instant, the simplex approach undermines this fundamental requirement of our solution.

One of the ideas behind this algorithm is that it should be configured based on the metrics we want to take into account, i.e., it should be possible to prioritize the metrics we want to stand out, in order to obtain the desired goal. Again, a simplex approach would not allow us to ensure this requirement of the edoclink product.

So, we set out for a scenario in which a new task arises to be allocated. First, it was necessary to identify the type of this task, which is defined by the set of attributes (name, phase, form, intervener, route). Tasks that contain this same set of attributes are considered tasks of the same type.

In this initial phase, the main focus of this solution was to identify the users that could process the task in question, i.e., among the group for whom the task is intended, the users that have already performed that type of task before were considered. It is important to note that this solution considered edoclink's "task delegation" concept, according to which a user can delegate to another all the tasks assigned to him during a period of time. This scenario serves to prevent constraints in situations where a user is absent.

After acquiring the first sample of users, which considers those who have experience performing the desired type of task from among the group for whom the task is intended, we collect the data needed to calculate each metric separately. Using the formulas described in section 3.2 Algorithm Metrics, we calculate, for each metric, a score for each of the potential performers. This set of scores is sorted from the highest value to the lowest, thus constructing a ranking of employee scores. We obtain four totally independent rankings, where the first place goes naturally to the employee best suited to perform the task, according to each metric.

Finally, in order to obtain a global ranking, from which we extract the final result, we assign a percentage that reflects the relevance we intend to give to each metric. The sum of these percentages must be equal to 100%. This global ranking is the output of this algorithm, where the first place is occupied by the employee who will best perform the task, considering all the metrics and their percentages of importance respectively. It is important to note that the percentage of the metrics can be configured through a parameter in the edoclink Backoffice.

$$Algorithm(Cx) = p1 * RANK(M1, Cx) + p2 * RANK(M2, Cx) + \cdots + pn * RANK(Mn, Cx)$$

$$\text{where } Cx = \text{employee}, pn = \text{percentage of each metric and } Mn = \text{metric}$$

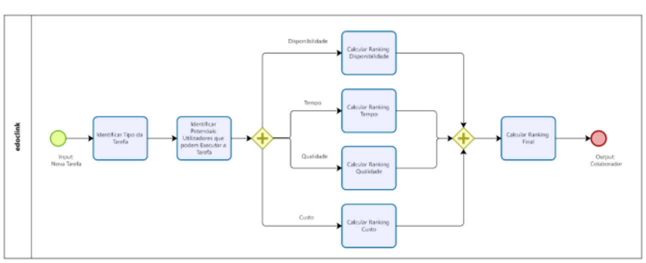

Figure 2: Algorithm Schema

When calculating each ranking, it should be noted that the history of task executions was not considered in its entirety, that is, the history since time 0 - when each employee joined the organization - was not considered. A parameter was created in edoclink's Backoffice that allows you to indicate how many months of history we want to be considered. The default value of this parameter is one year (12 months).

The entrance of new employees in the organization is another factor that we found relevant to reflect on. A new user is someone who has not yet performed tasks and therefore has no data that we can consider in the calculation of each of the metrics, in a first allocation. In the case of the Quality and Time metrics, we decided to assume that the value assigned to the new employee would be the lowest existing value (last position in the ranking). However, this would be counterbalanced by the Availability metric, where the new user would occupy the first place, by being 100% available.

4 DEVELOPMENT

As we saw earlier, a set of four metrics was defined and considered throughout this work: Availability, Time, Quality, and Cost. Naturally, there are situations in which these four criteria have the same relevance, that is, we do not intend to give special relevance to any of them. However, there may be situations in which we intend to highlight one or more factors.

In this sense, the weighting of the metrics, that is, the value attributed to each one, which is proportionally related to the importance it has in the context of an allocation, is parametrizable. Another factor that we found important to be parameterizable was the number of months of history to consider in the allocation of tasks. In fact, as we saw earlier, not all organizations evolve in the same way or at the same pace, so it is essential to be able to mark out and adapt this value.

Finally, thinking in a more macro way, we considered that there may exist situations or clients where you do not want to allocate tasks to resources automatically. In this sense, we considered it fundamental to be able to activate and deactivate the automatic task allocation module, so we also created this parameter. Along this line of thought, a parameter was also created to enable or disable task reallocation.

If the intervener of a particular stage is a group, it is possible to parameterize whether the stage of the route should consider an automatic allocation or not. Naturally, who defines this automatism is not the end user, but who configures the flow.
Thus, when a route is instantiated within a distribution, this configuration is taken into account when each step is started, so that the respective task allocation is performed. It should be noted that all allocations are auditable, in that the log of each allocation is kept, allowing a history to be recorded.

To take advantage of an organization's existing human resources and to ensure a balanced and fair division of labor, we decided to consider the possibility of reallocation. So, to consider task reallocation, the TaskReAllocationEnabled parameter of the edoclink backoffice must be active. Of course, you can only consider enabling or disabling the task reallocation parameter once the automatic allocation is active.

When task reallocation is considered, the moment an allocation arises, all tasks allocated to a user whose execution has not yet started are considered and reallocated. This feature allows tasks to be executed faster, balancing the distribution of work.

5 DEMONSTRATION

In order to demonstrate the validity of this solution, we have tried to apply it to different situations, including the implementation of several simulations and a pilot project in a real client.

The simulations were made with potential customers, in order to validate that the solution corresponds to the existing needs. Naturally, we tried to include simulations with different degrees of difficulty and complexity, to test as many different situations as possible.

More specifically, we tested the test scenario "Expense Request", in different contexts, considering the possibility of reallocating or not.

Due to bureaucratic issues, external to this work, it was not possible for us to conduct the pilot project at a client in a timely manner.

As an alternative, a mini pilot project was carried out, where edoclink's consulting team implemented several business processes. This project was based on three fundamental phases:
- Training: presentation of the presented solution to edoclink's consulting team.
- Development: Configuration and execution of tests on real cases, by each member of the edoclink consulting team.
- Evaluation: Analysis of the feedback received by each member of the consulting team, through the completion of a questionnaire.

6 EVALUATION

To evaluate the developed solution, we tried to validate this situation through simulation and through project-client.

So, to validate the efficiency of the algorithm, the first step was to try to typify the edoclink clients into small, medium and large, according to the volume of their step table. We obtained the following results:

Database Dimension		Exec (1)	Exec (2)	Exec (3)	Exec (4)	Exec (5)	Average
Small *(3400 stages)*	Without reallocation	0,046s	0,045s	0,047s	0,045s	0,048s	**0,0462s**
	With reallocation	0,472s	0,477s	0,472s	0,461s	0,469s	**0,4702s**
Medium *(Most common. 15000 stages)*	Without reallocation	0,051s	0,053s	0,051s	0,057s	0,052s	**0,0528s**
	With reallocation	0,539s	0,540s	0,538s	0,0539s	0,538s	**0,5388s**

Table 2: Simulations

The demonstration through simulation was carried out, having tested several cases, with different degrees of complexity. The results obtained were quite satisfactory, allowing us to validate the success of this work.

Since a large test database did not exist, it was not possible to test this scenario. Still, comparing the volume of the small and typical (medium) client, which is about 4 times larger, we found that there is no linear growth, which leads us to believe that the performance in a large client would also present satisfactory results.

Due to issues related to RGPD, external to the development of this work, it was not possible for us to perform the pilot project in a client in a timely manner. However, to get some feedback from users that work with edoclink on a daily basis, the consulting team implemented a set of business processes using the developed solution and later answered a questionnaire, where they gave us their feedback regarding the topic, which was quite positive.

Although the solution with reallocation presented better results, we identified a disadvantage during the execution of this work. The fact that the work queue of each employee is not fixed and, therefore, is constantly changing, promotes some instability in the time management of each employee.

7. CONCLUSION

7.1 Contributions

As a result of the work presented here, the previously established objectives were achieved, as well as theoretical and practical contributions.

From a theoretical point of view, the development of this work allowed formalizing a set of metrics for task allocation, in the context of the edoclink product, where we highlight Quality, Availability, Time and Cost. All these metrics are parameterizable, which means that a weighting can be assigned to them, depending on the importance you want to give them.

From a practical point of view, the solution developed meets the set of requirements it proposes, enabling an effective response to the needs of customers from different organizational contexts. As previously mentioned, this solution will integrate a future release of edoclink, more specifically the case tool module, a complementary module to the base product.

Through the development of this work, customers where the product edoclink is implemented will be able to better manage their resources, through an intelligent resource allocation, which is based on the characteristics of each user, as well as historical and temporal context at the time of allocation.

The possible imbalance in the distribution of tasks among users made us consider the hypothesis of task reallocation, which guarantees greater equity in each allocation. Thus, whenever a new task appeared on the list of tasks to be executed, all the assigned tasks that had not yet started would be reassigned to an employee, giving priority to the tasks that had been pending for the longest time.

7.2 Future Work

Even though the main objectives of this work were achieved, additional developments were identified that could complement and enrich our solution:

I. Partnership between edoclink and ATLAS products:
Extract from the BPMN drawing in ATLAS pertinent information that could determine the execution of the algorithm. In this way, at the time a flow was surveyed, if such characteristics were identified to be considered in the execution of the algorithm, they should be placed in the flow and subsequently read by ATLAS and implemented in the algorithm as constraints.

II. Instability of the employees' work queue
Consider reallocation only if it presented results above a decided percentage value or even study other reallocation orders.

III. Minimize the discrepancy and fairness of the positions in each ranking.

REFERENCES

[1] M. Dumas, M. La Rosa, J. Mendling, and H. A. Reijers, *Fundamentals of Business Process Management*. Springer Berlin Heidelberg, 2013. doi: 10.1007/978-3-642-33143-5.

[2] A. Ismaili-Alaoui, K. Benali, K. Baïna, J. Baïna, K. Ba, and J. Ba, “Business Process Instances Scheduling with Human Resources Based on Event Priority Determination,” pp. 118–130, 2018, doi: 10.1007/978-3-319-96292-4_10ï.

[3] R. Uahi and J. L. Pereira, “Alocação de Tarefas em Processos de Negócio Suportados por BPMS”.

[4] J. Xu, C. Liu, and X. Zhao, “Resource Allocation vs. Business Process Improvement: How They Impact on Each Other,” 2008, pp. 228–243. doi: 10.1007/978-3-540-85758-7_18.

[5] M. Armstrong and S. Taylor, *Armstrong’s Handbook of Human Resource Management Practice*, 15th ed. Kogan Page Publishers, 2014.

[6] M. Arias, E. Rojas, J. Munoz-Gama, and M. Sepúlveda, “A Framework for Recommending Resource Allocation based on Process Mining.”

[7] M. Arias, J. Munoz-Gama, M. Sepúlveda, and J. C. Miranda, “Human resource allocation or recommendation based on multi-factor criteria in on-demand and batch scenarios,” Eur. J. Ind. Eng., vol. 12, no. 3, pp. 364–404, 2018, doi: 10.1504/EJIE.2018.092009.