Modeling and Implementation of Sequencing of Tasks to Maximize the Processing in an Organization using the UML and Web Modeling

Amina Elkebbaj, Abdelwahed Namir, Youssef Amlal Bajja
Department of Mathematics and Computer Science, Hassan2-Mohammedia University, Casablanca, Morocco

Abstract—The main objective of this work is to develop a practical approach to improve customer satisfaction, which is generally regarded as the pillar of customer loyalty to a company. Today, customer satisfaction is a major challenge. In fact, listening to the customer, anticipating, properly managing his claims and serving him for the shortest deadlines are the keys and fundamental values for a company. The wait time before being served is systematically bad noted by the customers and can cause their dissatisfaction. For this reason, it is necessary to find a way to decrease the customer dissatisfaction by improving the service. This project describes a simple and practical approach for modeling and implementation of a sequence of tasks to maximize the serving and eliminates any manner of loss of time. The problem will be mathematically modeled by an integer linear programming and solved numerically by respecting priority rules. To illustrate this discussion, our project will be modeled by using UML and will create a web application for our problem designed to be used by people with little background in computer programming...

Keywords—Maximization, Modeling, Queueing, Satisfaction, UML, Web.

I. INTRODUCTION

It was noticed by specialists of analysis that a person spends at least the tenth of his wait time. So, this issue always gives a negative impression of services quality and a low customer satisfaction. To avoid these situations, companies have an interest in establishing continuous improvement process which ultimately goal elimination of all forms of wait time.

This project designed in a context which consists of maximizing the processing customer request within companies, in order to eliminate any manner of wasted time inside an organization.

We defined the framework of reflection of this work. The following is composed of three paragraphs:

- In the 2nd paragraph, we model mathematically the problem posed by an integer linear programming (already published in a previous article)[1].
- In the 3rd paragraph, we model the problem by using Unified Modeling Language.
- In the 4th paragraph, we model the problem by creating application web using SQL Server and PHP.

The conclusion looks back over the main the main focus point of this study and our contribution. It also shows the diverse uses of time and the possible applications of this work.

II. SPECIFICATION SHEET

A. Modeling of the Problem

We have a variety of tasks (phone, fax, email…) treated by the organization, we note:

- L: Group of tasks to be treated periodically by the organization;
- N: Maximum number of tasks which the organization has to handle within a period (Car(L) ≤ N);
- K: Number of types of tasks to be periodically handled by the organization (Ki for 1 ≤ i ≤ K);
- Pi: Minimal proportion of tasks of type Ki being necessarily and periodically in L, with 1 ≤ i ≤ K

For each period Pn, we have:

- Ci = Number of tasks of type Ki which are in the organization in the period Pn, with 1 ≤ i ≤ K (Ki capacity).
- Xi = Number of tasks of type Ki to be treated by the organization in the period Pn, with 1 ≤ i ≤ K (flot Ki).

Our problem consists, for a period Pn, to maximize the number of tasks to be treated by the organization. The problem can be mathematically modeled by the following model:

\[
\begin{align*}
\text{Max} & \quad \sum_{i=1}^{K} x_i \\
\text{S.C} & \quad x_i \leq C_i \text{ pour } 1 \leq i \leq K \\
& \quad x_i \geq \text{Min}\{C_i, p_i N\} \text{ pour } 1 \leq i \leq K \\
& \quad \sum_{i=1}^{K} x_i \leq N
\end{align*}
\]
The number of tasks which the organization can treat at the most during a period $P_n$ is:

$$\textbf{Max} \ (Z) = \text{Max} \left\{ \sum_{i=1}^{K} x_i \right\} = \text{Min} \left\{ \sum_{i=1}^{K} C_i, N \right\}$$

### B. The Optimal Solution of the Problem

In the majority of service systems, the rule of prioritizing the treatment of tasks is the “First-In-First-Out” (FIFO). However, in several situations, this rule is not applicable; because of the cost or the consequences which result are not satisfied (for example the urgent tasks must be treated at first).

In our work we will assign to the tasks that arise according to the priority rules. By priority rule, the tasks are classified by categories according to the rule of priority which is attributed to them.

The proposed solution consists of calculating the number of tasks of type $K_i$ which should be treated by order of priority. And if the first item with the highest priority has fewer tasks then we give the rest to the second priority, and so on...

If we organize the types of tasks in ascending order by giving the priority to the first type $K_1$ then the second type $K_2$ then the third type $K_3$ and so on, we have the optimal solution of (S) below:

$$x_1 = \text{Min}\{C_1, N - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\}\}$$

$$x_i = \text{Min}\left\{ C_i, N - \sum_{j=1}^{i-1} x_j - \sum_{j=i+1}^{K} \text{Min}\{C_j, P_jN\} \right\} \text{ for } 2 \leq i \leq K$$

$$x_K = \text{Min}\{ C_K, N - \sum_{j=1}^{K-1} x_j \}$$

**Proof:**

- **First case:** $C = \sum_{i=1}^{K} C_i \leq N$
  
  We put $x_i = C_i$ for $1 \leq i \leq K \implies \text{Max}(Z) = \sum_{i=1}^{K} C_i$.

- **2nd case:** $C = \sum_{i=1}^{K} C_i > N$

  We put in L the tasks to treat, $x_i = \text{Min}\{C_i, P_iN\}$ tasks of every type $K_i$ for $1 \leq i \leq K$. It remains:

  $N - \sum_{i=1}^{K} \text{Min}\{C_i, P_iN\}$ tasks to choose.

  As $K_1$ type have the first priority, we changed, if necessary, the value of $x_1$ by:

  $$x_1 = \text{Min}\left\{ C_1, \text{Min}\{C_1, P_1N\} + N - \sum_{i=1}^{K} \text{Min}\{C_i, P_iN\} \right\}$$

  It means that:

  $$x_1 = \text{Min}\left\{ C_1, N - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\} \right\}$$

  $x_1$ Verify the constraints:

  $$\text{Min}\{C_1, P_1N\} \leq x_1 \leq C_1$$

  Because

  $$\text{Min}\{C_1, P_1N\} \leq P_1N = N - \sum_{i=2}^{K} P_iN \leq N - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\}$$

  It remains $N - x_1 - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\}$ tasks to choose.

- **Then, as $K_2$ type** have the second priority, we modify, if necessary, the value of $x_2$ by:

  $$x_2 = \text{Min}\left\{ C_2, \text{Min}\{C_2, P_2N\} + N - x_1 - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\} \right\}$$

  It means that: $x_2 = \text{Min}\{C_1, N - x_1 - \sum_{i=3}^{K} \text{Min}\{C_i, P_iN\}\}$

  $x_2$ Verify the constraints:

  $$\text{Min}\{C_2, P_2N\} \leq x_2 \leq C_2$$

  Because

  $$x_1 \leq N - \sum_{i=2}^{K} \text{Min}\{C_i, P_iN\} = N - \text{Min}\{C_2, P_2N\} - \sum_{i=3}^{K} \text{Min}\{C_i, P_iN\}$$

  It means that:
\[
\text{Min}(C_2, P_2 N) \leq N - x_1 - \sum_{i=3}^{K} \text{Min}(C_i, P_i N)
\]

It remains \(N - x_1 - x_2 - \sum_{i=3}^{K} \text{Min}(C_i, P_i N)\) tasks to choose.

- And so on ……..

Such as \(C = \sum_{i=1}^{i=K} C_i > N\) at \(\sum_{i=1}^{i=K} P_i N = N\), so there is \(K' \in \{1, 2, \ldots, K - 1\}\):

\[
N - \sum_{i=1}^{K'} x_i - \sum_{i=K+1}^{K} \text{Min}(C_i, P_i N) = 0
\]

It means that, from \(K' + 1\), \(x_i\) can’t be changed and consequently:

\[
x_i = \text{Min}(C_i, P_i N) \text{ for } K' + 1 \leq i \leq K
\]

So \(\sum_{i=1}^{K} x_i = N \Rightarrow Z = N\)

This justifies that the solution given by equations (1) is an optimal solution of the system (S).

III. UML MODELING

Like any type of project, a software project requires an analysis phase, followed by a design phase. For that, we propose in this work the unified modeling language (UML) based on graphs. It is used in software development in order to understand and describe accurately the needs of users or customers. What do they want to do with the software? What features do they want? For what purpose? How should the action work? This is called "needs analysis". After validation of our understanding of the need, we imagine the solution. This is the analysis part of the solution.

This part will be illustrated by two UML diagrams giving a clear vision for non-IT professionals.

A. Class Diagram

This diagram is considered the most important diagram in the development [3]. It represents the conceptual architecture of the system. It describes the classes that the system uses, the inheritance links, or aggregation…

![Fig.1 Class diagram](image)

B. Use Case Diagram

However, lay people need a simple way to express their needs. This is precisely the role of “use cases” diagrams that collect, analyze, and organize the needs. It captures the behavior of a system, a subsystem, a class or a component such as a user sees the outside. It consists of two elements:

- The actor is the agent that interacts with the system.
- The Use case models a service provided by the system.
IV. WEB MODELING

A. the Used Technologies

To illustrate the area of wait time in customer satisfaction service, we want to model our problem as a web application using PHP which is a free programming language mainly used to produce dynamic web pages via an HTTP server and a database MySQL.

The objective of our application is to translate our mathematical model to SQL queries to simplify the creation of the web application. We begin by specifying the types of tasks in the queue and the maximum tasks that can be treated by the organization in the period $P_n$ as shown in the following figure.

Fig. 3 First window in the application

Then we will fill in the tasks types, customer requirements and the number of items in the queue as shown in the figure below:

Fig 4. Fill in the queue data
After filling the data, click on the button “calculate” and then the following table is obtained:

Let \( n = 100 \)

We have no Fax in the system so its percentage will be added into the phone as it has the highest priority:

\[
X_1 = \min (n \cdot P_1, C_1) = \min (75.70) = 70
\]

so we will have 5% of the phone that is not used, we will add it to the 2nd highest priority type (the message), so instead of 15% it will be 20% so we can handle 20 tasks and 10 remain in the queue. And so on…as shown on the following figure:

**Fig 5. The obtained result of the first example**

### B. Examples

Below we have 5 examples of exceptional cases of our application, they are real cases based on a Moroccan operator called Morocco Telecom:

**Fig 6. Second example**
C. Source Code
For modeling the problem on PHP, we used SQL query in order to simplify the creation of our application:

Blow is a piece of code that contains the calculations of the mathematical model using PHP:

```php
<?php
  error_reporting(E_ALL ^ E_DEPRECATED);
  include("Connection.php");

  $sql99="SELECT  `Id_element`,  `Type_element`,  `Pourcentage_element`,  `Valeur_element`,  `Element_traite`,  `Element_Queue`  FROM elements where nombre_max=0";
  $req99=mysql_query($sql99) or die(mysql_error());
  while ($ligne = mysql_fetch_array($req99))
    $reponses[] = $ligne;
  for ($cpt = 0; $cpt< count($reponses); $cpt++)
    {
      $x=$reponses[$cpt]["Valeur_element"];
      if($x==0)
        {
        // we will make the sum of the percentage which have the value equal to zero
          $sql="SELECT SUM(`Pourcentage_element`) FROM elements WHERE `Valeur_element`=0";
          $req=mysql_query($sql)or die('Erreur SQL !'.$sql.'<br>'.mysql_error());
          $data=mysql_fetch_array($req, MYSQL_NUM);
          // we will add the sum of the percentage which have the value equal to zero at the highest priority percentage different from zero
          $sql2= "select max(elements.Pourcentage_element)+ "$data[0]" from elements where `Valeur_element` <>0 ";
          $req2=mysql_query($sql2)or die ('Erreur SQL !'.$sql2.'<br>'.mysql_error());
          $data2=mysql_fetch_array($req2, MYSQL_NUM);
          //echo "the max percentage: ",data2[0];
          $sql3="SELECT max(elements.Pourcentage_element) FROM elements where `Valeur_element` <>0";
          $req3=mysql_query($sql3)or die('Erreur SQL !'.$sql3.'<br>'.mysql_error());
          $data3=mysql_fetch_array($req3, MYSQL_NUM);
          $sql4="UPDATE elements SET Pourcentage_element=$data2[0] where Pourcentage_element= $data3[0] and `Valeur_element` <>0 ";
          $req4=mysql_query($sql4) or die(mysql_error());
          $sql= "select max(`nombre_max`) from elements ";
          $req=mysql_query($sql) or die('Erreur SQL !'.$sql.'<br>'.mysql_error());
          $data=mysql_fetch_array($req, MYSQL_NUM);
          $sql5="SELECT  `Id_element`,  `Type_element`,  `Pourcentage_element`,  `Valeur_element`,  `Element_traite`,  `Element_Queue`  FROM elements where nombre_max=0";
          $req5=mysql_query($sql5) or die(mysql_error());
          while ($ligne = mysql_fetch_array($req5))
            $reponses[] = $ligne;
          for ($cpt = 0; $cpt< count($reponses); $cpt++)
            {
            //the calculation of treated elements
              $reponses[$cpt]["Element_traite"] = min((data[0]*$reponses[$cpt]["Pourcentage_element"]/100,$reponses[$cpt]["Valeur_element"]);
            //the difference in unexploited percentage: for example, we have 40% and just 30% was used so the rest is 10%, we will give it to the second highest percentage
```
V. CONCLUSIONS

The work done consists of designing a practical and pragmatic approach of implementation of a task scheduling to maximize treatment in an organization for a given period Pn. Therefore, it has been proposed a method of mathematical modeling and numerical solution based on a web application realized with the PHP programming language and SQL database.

This work opens the way for our view to a variety of research perspectives, it would be interesting first time:
Connect this approach to governance of information systems;
Pilot a decision-making system of examining the options and compare them to choose an action assisting in the decision making.

REFERENCES