JVLC

Journal of Visual Language and Computing

Volume 2021, Number 2

DOI: 10.18293/JVLC2021-N2

Journal preparation, editing and printing are sponsored by KSI Research Inc.

Journal of
Visual Language and Computing

An Open Access Journal published by

KSI Research Inc.

156 Park Square Lane, Pittsburgh, PA 15238 USA

Journal of
Visual Language and Computing

Volume 2021, Number 2

December 2021

Table of Contents

Regular Paper

Research Notes

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Ceiling-Vision-Based Mobile Object Self-Localization: A Composite Framework

Alfredo Cuzzocrea[a,*], Luca Camilotti[b] and Enzo Mumolo[b]

[a] iDEA Lab, University of Calabria, Rende, Italy & LORIA, Nancy, France
[b] University of Trieste, Trieste, Italy

ARTICLE INFO

Article History:
Submitted
Revised 6.1.2021
Second Revision 8.1.2021
Accepted 9.7.2021

Keywords:
Localization
Ceiling Landmarks
Mobile Objects
Two-Dimensional Dynamic Programming

ABSTRACT

A composite framework for supporting mobile object self-localization based on ceiling vision is presented in this paper. In particular, we consider the estimation - of the position of a mobile object using ceiling landmark images acquired by a low resolution camera placed on a mobile object, being light given by electric lampps ith circular holders. The pixeles of the images of the light holders on the ceiling are mapped on the image plane of the camera by means of a two dimesional dynamic programming algorithm (2D-DPA). The proposed algorithm estimates the distance from the camera lens to the center of the landmarks using only ceiling vision.Other intelligent tools and solutions further improve the accuracy of our self-localization task. Experiments confirms the benefits of our work.

1. Introduction

Self-localization of mobile objects is a fundamental requirement for autonomy. Mobile objects can be for example a mobile service robot, a motorized wheelchair, a mobile cart for transporting tasks or similar. Self-localization represents as well a necessary feature to develop systems able to perform autonomous movements such as navigation tasks. Self-localization is based upon reliable information coming from sensor devices situated on the mobile objects. There are many sensors available for that purpose. The early devices for positioning are rotary encoders. If the encoders are connected to wheels or legs movement actuators, relative movements of the mobile object during its path [3] can be measured. Then, mobile object positioning can be obtained with dead-reckoning approaches. Dead reckoning [3] is still widely used for mobile robot positioning estimation. It is also true that dead-reckoning is quite unreliable for long navigation tasks, because of accumulated error problems. Other popular sensor for self-localization are laser or sonar based range finders and inertial measurement devices. In outside scenarios the most popular approaches are based on Global Positioning System (GPS). Due to the importance of self-localization, many other solutions for indoor environment have been proposed so far with different cost and accuracy characteristics. For example the Ultra Wide Band radio signal indoor localization systems [15], or the Bluetooth-based angle of arrival radio devices [19], or a combination of them. However these systems have serious limitations in cost and reliability, respectively. Another important type of sensors which may be used for cost effective self-localization are the CCD cameras, which require computer vision algorithms such as for example visual odometry, [16]. Mobile objects vision based self-localization is currently an open research field [29] and an increasing number of new methods are continuously proposed. As a matter of fact we have to consider that self-localization of mobile objects requires centimeter-level accuracy and Computer Vision is one of the most cost-effective techniques able to reach that accuracy. Consequently, some surveys of Computer Vision based self-localization techniques appeared recently in the literature, [24].

In this paper we describes a novel Computer Vision

*This research has been made in the context of the Excellence Chair in Computer Engineering – Big Data Management and Analytics at LORIA, Nancy, France

alfredo.cuzzocrea@unical.it (Alfredo Cuzzocrea); camilotti@units.it (L. Camilotti); mumolo@units.it (E. Mumolo)
ORCID(s):

DOI reference number: 10-18293/JVLC2021-N2-019

based algorithm for estimating the distance from the camera lens to the center of ceiling landmarks with circular shape using a monocular low cost webcam. From that distance, mobile object localization approaches can be easily developed and a simple example is provided in this paper. The images of the ceiling landmarks are projected on the image plane of the camera. The projection can be analytically described, but the projections distortions, which may arise especially when low cost devices are used, may affect the results. To take into account the projection distortions in order to obtain a better precision of the results, we use an approximation of the two-dimensional dynamic programming (2D-DPA) algorithm [13] which finds a sub-optimal mapping between the image pixels of the ceiling landmarks and the image pixels of the landmarks projected on the camera plane. Since optimum 2D-DPA is NP-complete, however, many approximations have been developed. For example, the 2D-DPA technique described by Levin and Pieraccini in [22] has an exponential complexity in the image size, while Uchida and Sakoe describe in [30] a Dynamic Planar Warping technique with a complexity equal to $O(N^3 9^N)$, N being the image size. Lei and Govindaraju propose in [21] a Dynamic Planar Warping approximation with a complexity of $O(N^6)$. However each approximation has some limitation in terms of continuity of the mapping. In this paper we use a approximation of the optimum 2D-DPA with a complexity of $O(N^4)$ [12] which is implemented on a GPU to obtain real-time performance. When the landmark is far from the camera or if the environments has low lighting, an high quantization noise may arise in acquired images. However the algorithm we describe in this paper is particularly robust against noise due especially to the use of two-dimension DPA.

This paper is organized as follows. Section 2 reports on related work. Section 3 the localization problem is described, and in Section 4 the projection distortion is geometrically described, while in Section 5 the two-dimensional Dynamic Programming approximation is described. In Section 6 the proposed algorithm is sketched and in Section 7 the computer vision algorithms for the detection of landmarks on the image plane are reported. Section 8 sketches a possible global localization approach of the mobile object. In Section 9 we report some experimental comparison of the proposed algorithm with state of the art algorithm. Finally, Section 10 concludes the paper with concluding remarks and a suggestion of future works.

2. Related Work

Many papers on vision-based mobile robot self-localization appeared recently in the literature (e.g., [28, 14, 27, 31]).

Moreover, Avgeris *et al.* describe in [1] a self-localization algorithm for mobile robots that uses cylindrical landmarks resting on the floor and a single pivotal camera with an horizontal angle of view of 30-degree. Each cylindrical landmark has a different color in order to be easily detected by the robot. However, frontal vision could be occluded by objects or people. Such interference can be avoided by placing the landmarks on the ceiling, so that the camera is tilted toward the ceiling. Ceiling vision has been used by many authors to perform mobile robot localization. One of the early proposals is described in [25] and is based upon a digital mark pattern and a CCD camera. The camera is tilted, so the horizontal distance from the ceiling mark pattern is obtained measuring the ratio between the length and the width of the pattern picture.

Kim and Park [18] acquire ceiling images in a small area with a fish-eye lens camera. Ceiling outlines are detected by means of adaptive binarization and segmentation. Robot pose is obtained after identification of the ceiling region and the determination of the center and the momentum of the region. Lan *et al.* describe in [20] a mobile robot positioning algorithm based on artificial passive landmarks placed on the ceiling and infrared sensors. The landmarks are made of reflective film 2D structures containing dots assigned to unique ID's. The infrared sensors consist of an infrared camera and an infrared LED array. A similar approach is described in [32] where artificial passive reflective landmarks are placed on the ceiling and an infrared camera plus an infrared LED source are used to capture the reflection the IR light on the landmark for estimating the robot pose.

Wang *et al.* describe in [34] a vision control system which captures ceiling RGB images with a camera placed on the robot, converts the image to HSV color space and uses V channel images to reduce the effect of illumination lamps. The common objects and the straight lines on the ceiling are detected by template matching and used to estimate the robot orientation.

Other Computer Vision based approaches are based on the Free Space Density concept. For example, A. Ribacki *et al.* use an upward facing camera to detect the ceiling boundaries and to estimate the ceiling space density from the current image [26]. Other authors, for example [5, 6] use the ceiling depth images for robot localization. In these approaches self-localization is obtained from Principal Component Analysis of ceiling depth images. Ceiling vision is used by many other authors to perform self-localization of mobile robots. For example Lin *et al.* describe in [23] a visual odometry algorithm based on a monocular camera which points to the ceiling. The algorithm uses several local features detectors for matching the features between two sequential frames of the ceiling.

In addition, it should be considered the emerging integration of these topics with the innovative *big data trend* (e.g., [8, 11, 7, 10, 2]). Here, the main research perspective is to take into account the well-known *3V model* of big data, including *volume*, *velocity* and *variety*.

3. Problem Description

We show in Figure 1 a mobile object in an indoor environment. The movable object is equipped with a camera tilted towards the ceiling at an angle φ. We call h the distance between the camera and the ceiling. Moreover in Fig-

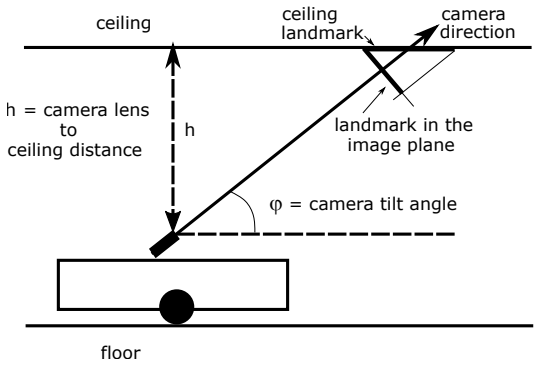


Figure 1: A mobile object with a camera on it, tilted toward the ceiling.

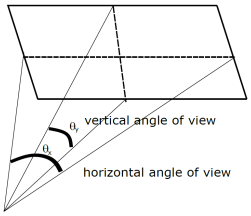


Figure 2: The horizontal and vertical angles of view of the camera.

Figure 3: An example of the circular lamp holder used in this paper.

ure 2 the horizontal and vertical angles of view of the camera, called θ_x, and θ_y respectively, are highlighted.

The direction towards which the camera is oriented is shown with the 'Camera Direction' arrow. The ceiling landmark is shown in Figure 1 with a segment with a greater thickness and the image plane of the camera is shown with a segment orthogonal to the camera direction. The ceiling landmark is projected to the landmark on the image plane. The visual landmarks positioned on the ceiling used in this approach are the lighting holders as that shown in Figure 3.

We choose landmarks with isotropic shapes because in this way the distortion components due to image rotation can be eliminated. The simplest bidimensional isotropic shape is the circle. As shown in Figure 3, the lines of pixels on the image plane are all parallel to the reference abscissa on the ceiling plane regardless of the angle of the camera with respect to the landmark. It is important to remark that each landmark must be distinguishable from the others and its co-

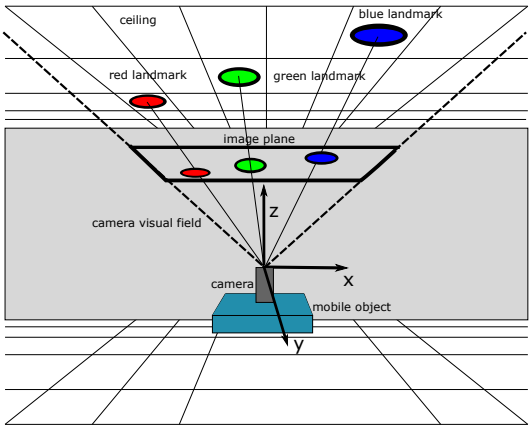


Figure 4: Schematic representation of orthonormal reference system, landmarks and image plane.

ordinate in the global reference system must be known. A schematic representation of a mobile object and some landmarks with the orthonormal reference system centered on the camera lens is shown in Figure 4.

The reference abscissa changes dynamically in relation to the direction of the focal axis. The reference abscissa, in fact, is always normal to the focal axis and at the same time it is parallel to the horizon.

The landmarks must be distinguishable from each other. There are many possible solutions for making the landmarks distinct. A simple possibility is to paint each holder with a different color. More recently, the characteristic frequency of fluorescent lights has been used, for instance in [35]. In this paper we used the simplest solution, namely we painted adjacent lamp holders with different colors.

For this reason the landmarks in Figure 4 are represented with different colors, where for simplicity the three circular landmarks positioned on the ceiling are colored in red, blue and green. Figure 4 shows that the landmarks which fall within the visual field of the camera are projected onto the image plane of the camera. Of course we know in advance the physical position of each landmark in the global reference system. On the other hand the landmark colours can be detected using well known computer vision techniques.

4. Projective Transformations

The projective transformation is the linear transformation of coordinates reported in (1).

$$p' = Tp \tag{1}$$

where p represents a generic point in space expressed in homogeneous coordinates, relative to the orthonormal reference system S described by the quadruple $(O, \hat{\imath}, \hat{\jmath}, \hat{k})$.

The projected point p' is expressed in coordinates relative to the reference system S' described by the quadruple $(O', \hat{\imath}', \hat{\jmath}', \hat{k}')$, where $\hat{\imath}' = \hat{\imath}$, $\hat{\jmath}'$ has the direction of the segment $\overline{MQ}$ and $\hat{k}'$ has the direction of the normal to the segment $\overline{MQ}$.

Since p is expressed with the three components (x_p, y_p, z_p) and p' has the three components $(x_{p'}, y_{p'}, z_{p'})$, eq. (1) can be also written as follows:

$$\begin{pmatrix} x_{p'} \\ y_{p'} \\ z_{p'} \end{pmatrix} = T \begin{pmatrix} x_p \\ y_p \\ z_p \end{pmatrix} \tag{2}$$

Such a transformation maintains the properties of collinearity, that is, the points which in S belong to a line, are aligned in a line also in S'. However, projective transformation may not be defined for every point of S, in the sense that some points could be mapped in S' at infinity.

Let us focus on Figure 4 from the left side, that is the $y - z$ plane of the orthonormal reference system which has its origin coincident with the center of the camera lens. This plane is highlighted in Figure 5, where the ceiling is at $z = h$, and the field of view of the camera is shown with points M and E. Let us assume that a landmark falls within the vertical angle of view. Then, the center of the landmark is the point C. On the other hand, if we view Figure 4 from the front side, that is the $x - z$ plane, we obtain the system shown in Figure 6. Of course the camera image plane, which is the plane normal to the focal axis in Figure 4, is shown with the segment $M - Q$ in Figure 5 and segment $G - I$ in Figure 6.

Suppose we fix a point P on the ceiling. If the point falls within the field of view of the camera it is shown as P in Figure 5. Let (p_x, p_y, p_z), with $p_z = h$, be the coordinates of P. The point P is projected to the image plane of the camera to the point P', which has coordinates $(x_{p'}, y_{p'}, z_{p'})$. Also the center C of the landmark in Figure 5 is projected to the point C' and the segment $M - E$ is projected to the segment $M - Q$ in the image plane.

In this model, the focal distance of the device or other characteristic parameters are not taken into account. It is in fact a purely ideal model, which has the only purpose of deriving the relations that define the projective transformation from the orthonormal system whose origin coincides with the center of the camera lens to the image plane system. The latter is chosen independently of the characteristics of the camera. With reference to the Figures 5 and 6, we introduce the following geometric variables characteristic of the problem.

- $$\Phi = \varphi + \frac{\theta_y}{2} - \frac{\pi}{2} \tag{3}$$
- The distance a from the origin to the barycenter of the landmark projected on the image plane:

$$\begin{aligned} a =& \overline{OC'} = \\ =& \frac{h}{\sin(\varphi)} - h(\tan(\varphi) + \frac{1}{\tan(\varphi)})\cos(\varphi) \end{aligned} \tag{4}$$

- The abscissa of the point P' on the image plane:

$$\begin{aligned} \frac{b}{2} =& \overline{MC'} = \overline{C'Q} = \\ =& h(\tan(\varphi) + \frac{1}{\tan(\varphi)}\sin(\varphi)) \end{aligned} \tag{5}$$

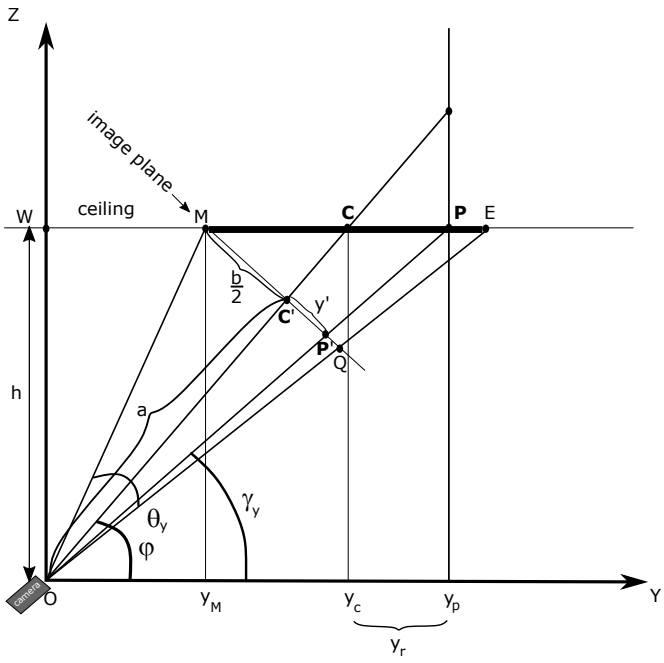


Figure 5: Plane $y - z$ in orthonormal reference system.

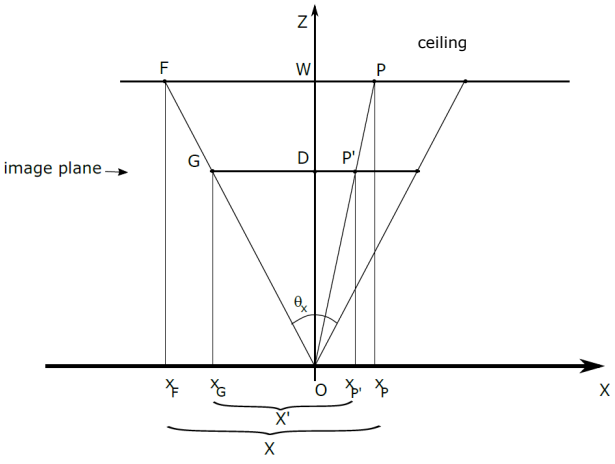


Figure 6: Plane $x - z$ in orthonormal reference system.

Equations (4) and (5) are developed in **Appendix A**. Moreover, we define the following two variables:

$$G = -h\,(\tan\Phi \tan\varphi + 1) \tag{6}$$

and:

$$F = h\,(\tan\varphi - h\tan\Phi) \tag{7}$$

We remark that the following considerations are based on three coordinate systems, namely an orthonormal reference system centered on the camera lens, shown in Figure 4, an orthonormal reference system on the image plane and a system on the ceiling plane which is simply translated by h

with respect to that centered on the camera lens. In general, points on the systems centered on the camera lens and on the ceiling are denoted with a capital letter, such as **P**, while that on the image plane of the camera are denoted with a capital letter plus an apex such as **P'**. In this case, **P'** is the **P** point projected on the image plane. If we look at the landmark seen from the orthonormal reference system centered on the camera lens, its barycenter is located at (x_c, y_c). A generic point on the ceiling has coordinate (x, y) and the same point projected on the image plane is (x', y'). The coordinates of a generic point on the landmark is given relative to its barycenter: $(x = x_c + x_r)$ and $(y = y_c + y_r)$. According to Figures 6 and 5 the offsets x_r, y_r are projected to the image plane in x', y'.

Assume now we have an optimum mapping between images. In other words, assume that, having two images A and B, $A = \{a(i,j),\ i,j = 1,...,N\}$ and $B = \{b(u,v),\ u,v = 1,...,M\}$, we can estimate the following mapping function:

$$F(i,j) = \begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} x(i,j) \\ y(i,j) \end{bmatrix} \tag{8}$$

which maps each pixel (i, j) of one image to the pixel (u, v) of the other image such that the difference between the two images is minimized, as shown in (9).

$$min \sum \sum \|a(i,i) - b(u,v)\| \tag{9}$$

where $u = x(i,j)$ and $v = y(i,j)$. Such mapping is performed through a two dimensional Dynamic Programming operation [30]. 2D-DPA is the base of image matching algorithms called Elastic Image Matching. Unfortunately, the Elastic Image Matching operation is NP-complete [17]. For this reason we devise an approximation which reduces the 2D-DPA operation complexity to $O(N^4)$, as described below in Section 5. The barycenter of the landmarks, (x_c, y_c), are estimated using the following Proposition.

By measuring the abscissa and ordinate (x', y') of a generic point on the landmark projected on the image plane we can estimate the coordinate (x_c, y_c) of the ceiling landmark using the following equations:

$$x_c = \frac{h\cos(\varphi - \gamma_y)(x' - g)}{a\sin(\gamma_y)} + g - x_r \tag{10}$$

$$y_c = \frac{aG + ay_r\tan(\varphi) - (y' - \frac{b}{2})(y_r + F)}{y' - \frac{b}{2} - a\tan(\varphi)} \tag{11}$$

In **Appendix B** we give a sketch of the derivations.

A similar estimation of the coordinates of the landmark barycenter is obtained for other points inside the landmarks. Therefore a sequence of barycenter coordinates x_c, y_c is thus obtained, of which we compute the expected value. The algorithm thus estimates $\mathbf{E}(x_c)$ and $\mathbf{E}(y_c)$ by measuring the values y' and x' of the distorted image on the image plane.

The distance from the camera lens and the landmark in the ceiling reference system is thus the following:

$$d = \sqrt{\mathbf{E}(x_c)^2 + \mathbf{E}(y_c)^2)} \tag{12}$$

with reference to Figures 6 and 5, where $C = (x_c, y_c, z_c)$ is the barycenter of the landmark in the reference system (O, i, j, k). We obtain the sub-optimal correspondence, pixel by pixel, between a reference image and a distorted image by means of approximated two dimensional dynamic programming, . Our algorithm therefore uses the deformation of the image to derive the distance of the landmark, i.e. it is intended to determine how the perspective has distorted the image.

The coordinates of the barycenter of the ceiling landmarks are obtained using the coordinate x' measured on the image plane and x_r using the mapping function, and in terms of y' and y_r. Clearly (x_r, y_r) and (x', y') are both known because they are derived from the coordinates of the pixels in the pattern and in the test images respectively. What associates the two pixels is the mapping relationship described in (8) obtained by 2D-DPA.

The characteristic that differentiates the algorithms present in the literature from the one developed in this paper is the statistical character of the obtained estimate. The algorithm based on dynamic programming is able to calculate a position estimate for each single pair of associated pixels from the mapping. The advantage is that a large number of points are used, which contribute to the calculation of the average distance value. This makes the estimate more truthful, especially when the landmark is very distant, which results in a smaller image and a greater quantization error.

5. 2D Dynamic Programming Based Image Mapping Technique (2D-DPA)

For the sake of coherence, we repeat now the mapping considerations summarized above (equations 8 and 9) about images A and B using instead images X and Y. Given two images, $X = \{x(i,j)\}$ and $Y = \{y(u,v)\}$, the mapping of one image to the other is represented by the following operation:

$$D(X,Y) = \min \sum_{i=1}^{N} \sum_{j=1}^{N} \|x(i,j) - y(u,v)\| \tag{13}$$

where $u = x(i,j)$, $v = y(i,j)$ is the mapping function between the pixels of X and Y. The quantity $D(X,Y)$ gives a distance between the image X and the optimally deformed Y, the optimal warping function $x(i,j)$, $y(i,j)$ gives an interpretation of the image X according to the generation model Y.

Given the $i-$th row of the X image and the $j-$th row of the Y images, namely $Y_j = (y_{j,1}, y_{j,2}, \ldots, y_{j,N})$, $X_i = (x_{i,1}, x_{i,2}, \ldots, x_{i,N})$ respectively, the distance between the two rows is obtained by applying a 1D-DPA [33] for finding a warping among the two rows as described in (14). Here the map M' is, say, over (n, m) coordinates, so that

$M'_l = ((i_l, n_l), (j_l, m_l))$.

$$d(X_i, Y_j) = \frac{\min_{M'} \sum_{l=1}^{M'} d(M'_l)}{M'} = \frac{\min_{M'} \sum_{l=1}^{M'} \left\| x_{i_l,n_l} - y_{j_l,m_l} \right\|}{2N} \tag{14}$$

Finally, the distance between the two images is obtained by (15). In this case the map $\overline{M'}$ is between all the rows of X and Y. As before, $|\overline{M'}|$ is the length of the path.

$$D(X,Y) = \frac{\min_{\overline{M'}} \sum_k d(\overline{M'}_k)}{|\overline{M'}|} = \frac{\min_{\overline{M'}} \sum_k d(X_{i_k}, Y_{j_k})}{|\overline{M'}|} = \frac{\min_{\overline{M'}} \sum_k \frac{\min_{M'} \sum_{l=1}^{|M'|} d(M'_l)}{2N}}{2N} = \frac{\min_{\overline{M'}} \{ \sum_k \min_{M'} \sum_{l=1}^{|M'|} \left\| x_{i,n_l} - y_{j,m_l} \right\| \}}{4N^2} \tag{15}$$

Let us assume that the images are of equal size, that is $N \times N$ pixels. Then the length of the optimum path between the two images is equal to $2N$. The local distances in each point of this path is obtained with other 1D-DPA with paths of length $2N$. The total length is the sum of $2N$ along the $2N$ long path, giving $4N^2$ at the denominator. The complexity of the described operation is $O(N^2 N^2) = O(N^4)$ where N is the image size.

6. Proposed Algorithm

The algorithm described in this paper is summarized as follows. The inputs of the algorithm are the two gray-scale images A and B which are the landmarks on the image plane and on the ceiling respectively. We perform the 2D-DPA algorithm on these two images to obtain the mapping function as result. The mapping function is represented with a linked list where each node is the map related to the two pixels. The function $get()$ gives as result the value of the pixel on the image indicated as input and is used to get the values of the two pixels linked by the map on the two landmark images. To decide if the pixel is a landmark pixel or not, we consider their gray levels. The landmarks have a lower values with respect to the environment and thus if the pixel values is less then a threshold, the pixel is a landmark pixel.

7. Computer Vision Approach for Extracting Landmark Images

We briefly summarize in this Section the computer vision operations we did on the image acquired from the ceiling. The problem is to detect from the image plane the isotropic images which represent the landmark. Another operation, which is not reported here, is the identification of

Input: A, B
Output: $distance$

```
img=Detect(A);   ▷ get the landmark in the image plane
id=identify(img);                    ▷ identify the landmark
head=2D-DPA(A, B);
ptr=head;          ▷ head is the list of mapping function
repeat
    pixA = get(A, ptr);                          ▷ pixel of A
    pixB = get(B, ptr);                          ▷ pixel of B
    if (pixA ≤ L)&&(pixB ≤ L) then ▷ if the pixels are in the landmark
        Compute x_c, y_c with (10) and (11)
        sum_y+ = y_c;
        sum_x+ = x_c;
        counter++;
    ptr = ptr → next;
until ptr == NULL
y_c = sum_y/counter;
x_c = sum_x/counter;
distance = sqrt(x_c^2 + y_c^2);
return distance
```

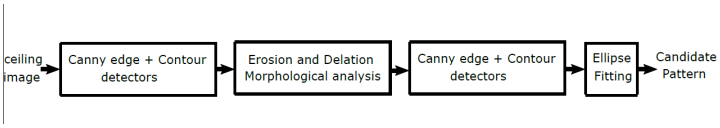


Figure 7: Block diagram of the Computer Vision algorithms.

the landmark. The simplest way is to draw the landmarks with different colors, since the computer vision operations to identify the colors are very simple. There are however many other ways which can be used for the identification, typically based on some type of code drawn inside the landmark. Of course the computer vision operations are slightly more complex than using different colors. More importantly, the computer vision operations to decode drawn codes could need greater camera resolution.

We report in Figure 7 the Computer Vision algorithms we applied on the original image for extraction of isotropic images.

The algorithms are described as follows:

- The acquired image is first transformed in gray-scale, and then its edges are obtained via the Canny's operator, obtaining the Edge image.
- From the Edge image, its contours are extracted, obtaining the Contour1 image.
- The Contour1 image is processed via morphological analysis. More precisely the opening operation with circular structuring element, is applied to Contour1 image in order to eliminate the little Side Dishes. The edges are then extracted again with the Canny opera-

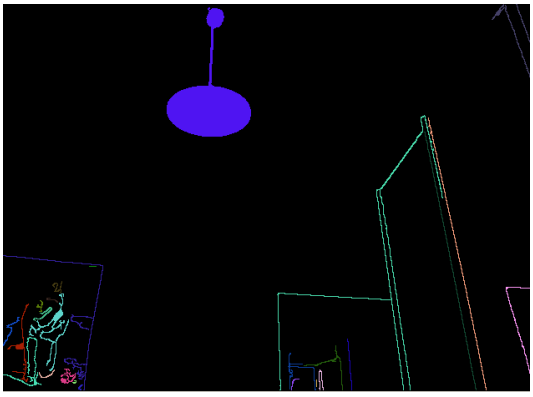


Figure 8: Processed results, with reference to Figure 3.

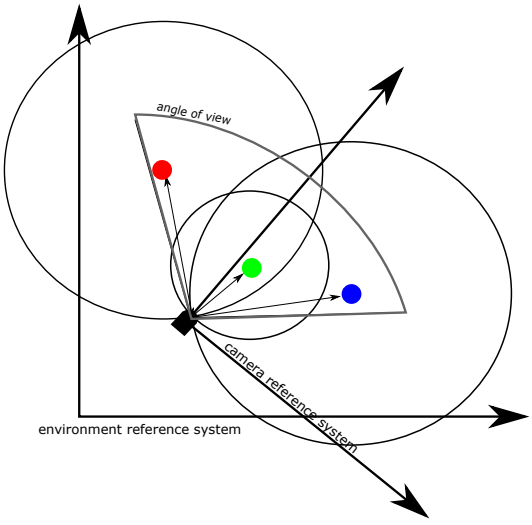


Figure 9: Sketch of a possible localization by trilateration.

tor, and then the contours are extracted again, finally obtaining the Contour2 image.

- Ellipse fitting is applied to Contours2 image. Based on the position and size of the found ellipses, square portions are cut out from original image. Most likely, the landmarks are contained in one of the extracted portions.

The results are shown in Figure 8. These results refer to the input image shown in Figure 3.

8. Localization

The localization of the mobile object is an issue we leave open as starting from distance estimation several possible solution can be developed. However, just to point out a possible simple idea based on trilateration, we report Figure 9.

This figure shows a global reference system which is related to the indoor environment is shown. Another reference system which is rotated and translated with respect to the first one. The origin of second reference system is centered on the camera lens of the mobile object. Note that the $x - y$ planes shown in Figure 9 correspond to the ceiling plane. The mobile object identify the landmarks and knows in advance their location coordinate in the global reference system. Our algorithm estimates the distance from the mobile object and the detected landmarks. Therefore, we can think to draw a circle with center on the landmark and radius equal to the estimated distance. If at least three different landmarks are detected, the mobile object can be localized in global reference system.

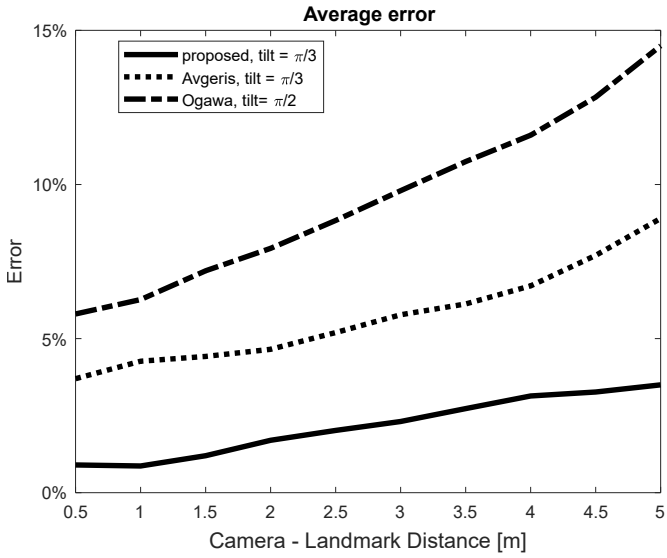


Figure 10: Average errors of the estimated distance.

9. Experimental Results

The experiments has been made using an Intel I7 CPU with 8cores running at 3.07GHz and a memory of 24GB. Then, the two dimensional DPA algorithm has been written in the CUDA framework and executed on a NVidia Kepler TM GK110 device. A low cost 640×480 webcam is used for image acquisition. In Figure 10 we report the average error of the estimation distance from the camera lens and the barycenter of the landmarks.

As a general consideration regarding these results, if the camera tilt-angle is high (i.e. if the inclination of the optical axis is close to the perpendicular to the ceiling) the error is quite small, but the field of view turns out to be very limited. To take advantage of wider fields of view, higher tilt-angles must be used. In this case, however, the error is higher. Furthermore, if the light in the environment become worse, the average error increases. Our algorithm, however, is quite robust against noise. The curve drawn in Figure 10 with solid line is obtained by the algorithm described in this paper. The curve in the middle is related to the approach developed in 2019 by Avgeris *et al.* and described in [1]. Finally, the higher curve is related to the work proposed by Ogawa *et al.* in [25]. Despite being quite old we include this result because its setting is very similar to this paper (the camera is directed towards the ceiling with a tilt angle equal to 30 de-

grees). The errors are in any cases well above that obtained by all the other algorithms.

10. Final Remarks and Future Work

In this paper we present an algorithm to measure the distance of a mobile object to the lightning lamps used as ceiling landmarks in indoor environment. The algorithm has many attractive features, mainly the accuracy, which is better than many other visual-based algorithms. Also, the distance measurements algorithm is robust against noise. Quantization noise can be high in low lighting condition of the environment and if the distance from landmarks and camera is high. The negative outcome of the algorithm is the high complexity of 2D-DPA which, even if polynomial, can lead to high computational times. In [12], however, we show how the 2D-DPA when implemented on a NVidia Kepler TM GK110 device leads to computation time less then 100 ms, for image size of 100×100 pixels.

This paper naturally opens to the development of localization algorithms based on our distance estimation algorithm. The global localization is in fact under development. Another open important issue is the landmark placement. Finally, the estimation of the orientation of the mobile object is another fundamental problem not addressed in this paper. The use of the characteristic frequencies of fluorescent lamp is an interesting method to identify the landmarks. In this case, adaptive and artificial intelligence metaphors, perhaps inherited by different scientific context (e.g., [4, 9]), may be considered. Future works will be focused on these open points.

Acknowledgements

This research has been partially supported by the French PIA project “Lorraine Université d’Excellence", reference ANR-15-IDEX-04-LUE.

Appendix A

Referring to Figure 5, we derive below the geometric variables reported in Section 4.

Consider first Eq. (4),

$$a = \overline{OC'} = \overline{OC} - \overline{C'C} \tag{16}$$

From the right triangle $\triangle OCC_y$, we have:

$$\overline{OC} = \frac{h}{\sin(\varphi)} \tag{17}$$

Moreover, from $\triangle MC'C$, we have:

$$\overline{C'C} = \overline{MC}\cos\varphi = (\overline{WC} - \overline{WM})\cos\varphi \tag{18}$$

i.e.:

$$\overline{C'C} = (\frac{h}{\tan\varphi} + h\tan\Phi)\cos\varphi \tag{19}$$

Therefore:

$$a = \frac{h}{\sin(\varphi)} - h\left(\frac{1}{\tan\varphi} + \tan\Phi\right)\cos\varphi \tag{20}$$

Considering Eq. (5), we have:

$$\frac{b}{2} = \overline{MC}\sin\varphi = h\left(\frac{1}{\tan\varphi} + \tan\Phi\right)\sin\varphi \tag{21}$$

Appendix B

We now report a sketch of the derivation of the two propositions reported in Section 4.

Let us start with Eq. (7). Regarding Figure 5, the angle formed by segments $\overline{OR}$ and $\overline{OP}$ is equal to $(\Phi - \gamma_y)$, so:

$$\begin{aligned}\tan(\Phi - \gamma_y) =& \frac{\overline{RP}}{\overline{OR}} = \\ =& \frac{[y\tan\varphi - h(\tan\Phi\tan\varphi + 1)]\cos\varphi}{h(\tan\varphi + y - h\tan\Phi)\cos\varphi}\end{aligned} \tag{22}$$

In addition, to simplifying the $\cos\varphi$, we use the definition of F and G reported above.

$$G = -h(\tan\Phi\tan\varphi + 1) \tag{23}$$

$$F = h(\tan\varphi - h\tan\Phi) \tag{24}$$

Then, we have:

$$\tan(\Phi - \gamma_y) = \frac{y\tan\varphi + G}{y + F} \tag{25}$$

We conclude that:

$$y' = \frac{b}{2} + a\frac{y\tan\varphi + G}{y + F} \tag{26}$$

By setting $y = y_r + y_c$ we obtain the landmark coordinate y_c reported in Eq. (11).

Going now back to Eq. (7), let us consider Figure 6. For lack of space we only state that, according to considerations very similar to that just described, we can conclude that:

$$x' = g + \frac{a\sin\gamma_y(x - g)}{h\cos\varphi - \gamma_y} \tag{27}$$

where $g = \overline{GD} = \overline{DI} = a\tan\frac{\theta_x}{2}$. As we did previously, we substitute $x = x_r + x_c$ in 27 and thus we can obtain x_c, described in Eq. (10).

References

[1] Avgeris, M., Spatharakis, D., Athanasopoulos, N., Dechouniotis, D., Papavassiliou, S., 2019. Single vision-based self-localization for autonomous robotic agents, in: 2019 7th International Conference on Future Internet of Things and Cloud Workshops (FiCloudW), IEEE. pp. 123–129.

[2] Campan, A., Cuzzocrea, A., Truta, T.M., 2017. Fighting fake news spread in online social networks: Actual trends and future research directions, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE. pp. 4453–4457.

[3] Campbell, S., O'Mahony, N., Carvalho, A., Krpalkova, L., Riordan, D., Walsh, J., 2020. Where am i? localization techniques for mobile robots a review, in: 2020 6th International Conference on Mechatronics and Robotics Engineering (ICMRE), IEEE. pp. 43–47.

[4] Cannataro, M., Cuzzocrea, A., Pugliese, A., 2002. Xahm: an adaptive hypermedia model based on xml, in: Proceedings of the 14th international conference on Software engineering and knowledge engineering, pp. 627–634.

[5] Carreira, F., Calado, J., Cardeira, C., Oliveira, P., 2018. Navigation system for mobile robots using pca-based localization from ceiling depth images: Experimental validation, in: 2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO), IEEE. pp. 159–164.

[6] Carreira, F., Calado, J.M., Cardeira, C., Oliveira, P., 2015. Enhanced pca-based localization using depth maps with missing data. Journal of Intelligent & Robotic Systems 77, 341–360.

[7] Ceci, M., Cuzzocrea, A., Malerba, D., 2015. Effectively and efficiently supporting roll-up and drill-down olap operations over continuous dimensions via hierarchical clustering. Journal of Intelligent Information Systems 44, 309–333.

[8] Chatzimilioudis, G., Cuzzocrea, A., Gunopulos, D., Mamoulis, N., 2013. A novel distributed framework for optimizing query routing trees in wireless sensor networks via optimal operator placement. Journal of Computer and System Sciences 79, 349–368.

[9] Cuzzocrea, A., 2006. Combining multidimensional user models and knowledge representation and management techniques for making web services knowledge-aware. Web Intelligence and Agent Systems: An international journal 4, 289–312.

[10] Cuzzocrea, A., De Maio, C., Fenza, G., Loia, V., Parente, M., 2016a. Olap analysis of multidimensional tweet streams for supporting advanced analytics, in: Proceedings of the 31st Annual ACM Symposium on Applied Computing, pp. 992–999.

[11] Cuzzocrea, A., Mansmann, S., 2009. Olap visualization: models, issues, and techniques, in: Encyclopedia of Data Warehousing and Mining, Second Edition. IGI Global, pp. 1439–1446.

[12] Cuzzocrea, A., Mumolo, E., Pirró, D., Vercelli, G., 2016b. An efficient cuda-based approximate two-dimensional dynamic programming algorithm for advanced computer vision applications, in: 2016 IEEE International Conference on Systems, Man, and Cybernetics (SMC), IEEE. pp. 004251–004258.

[13] Glasbey, C.A., 2009. Two-dimensional generalisations of dynamic programming for image analysis. Statistics and Computing 19, 49.

[14] Gross, H.M., Koenig, A., Boehme, H.J., Schroeter, C., 2002. Vision-based monte carlo self-localization for a mobile service robot acting as shopping assistant in a home store, in: IEEE/RSJ International Conference on Intelligent Robots and Systems, IEEE. pp. 256–262.

[15] He, C., Xia, Y., Yu, C., Jiang, C., 2020a. A multi-hop distributed indoor localization algorithm for ultra-wide-band sensor network, in: 2020 16th International Conference on Control, Automation, Robotics and Vision (ICARCV), IEEE. pp. 1335–1340.

[16] He, M., Zhu, C., Huang, Q., Ren, B., Liu, J., 2020b. A review of monocular visual odometry. The Visual Computer 36, 1053–1065.

[17] Keysers, D., Unger, W., 2003. Elastic image matching is np-complete. Pattern Recognition Letters 24, 445–453.

[18] Kim, Y.G., Park, T.H., 2016. Localization of mobile robots from full detection of ceiling outlines, in: 2016 IEEE International Conference on Information and Automation (ICIA), IEEE. pp. 1515–1520.

[19] Kumar, G., Gupta, V., Tank, R., 2020. Phase-based angle estimation approach in indoor localization system using bluetooth low energy, in: 2020 International Conference on Smart Electronics and Communication (ICOSEC), IEEE. pp. 904–912.

[20] Lan, G., Wang, J., Chen, W., 2016. An improved indoor localization system for mobile robots based on landmarks on the ceiling, in: 2016 IEEE International Conference on Robotics and Biomimetics (ROBIO), IEEE. pp. 1395–1400.

[21] Lei, H., Govindaraju, V., 2004. Direct image matching by dynamic warping, in: 2004 Conference on Computer Vision and Pattern Recognition Workshop, IEEE. pp. 76–76.

[22] Levin, E., Pieraccini, R., 1992. Dynamic planar warping for optical character recognition, in: Acoustics, Speech, and Signal Processing, IEEE International Conference on, IEEE Computer Society. pp. 149–152.

[23] Lin, Q., Liu, X., Zhang, Z., 2019. Mobile robot self-localizationusing visual odometry based on ceiling vision, in: 2019 IEEE Symposium Series on Computational Intelligence (SSCI), IEEE. pp. 1435–1439.

[24] Morar, A., Moldoveanu, A., Mocanu, I., Moldoveanu, F., Radoi, I.E., Asavei, V., Gradinaru, A., Butean, A., 2020. A comprehensive survey of indoor localization methods based on computer vision. Sensors 20, 2641.

[25] Ogawa, Y., Lee, J.H., Mori, S., Takagi, A., Kasuga, C., Hashimoto, H., 1999. The positioning system using the digital mark pattern-the method of measurement of a horizontal distance, in: IEEE SMC'99 Conference Proceedings. 1999 IEEE International Conference on Systems, Man, and Cybernetics (Cat. No. 99CH37028), IEEE. pp. 731–741.

[26] Ribacki, A., Jorge, V.A., Mantelli, M., Maffei, R., Prestes, E., 2018. Vision-based global localization using ceiling space density, in: 2018 IEEE International Conference on Robotics and Automation (ICRA), IEEE. pp. 3502–3507.

[27] Schmitt, M., Rous, M., Matsikis, A., Kraiss, K.F., 1999. Vision-based self-localization of a mobile robot using a virtual environment, in: Proceedings 1999 IEEE International Conference on Robotics and Automation (Cat. No. 99CH36288C), IEEE. pp. 2911–2916.

[28] Schneegans, S., Vorst, P., Zell, A., 2007. Using rfid snapshots for mobile robot self-localization., in: EMCR.

[29] Shang, W.A., 2019. Survey of mobile robot vision self-localization. Journal of Automation and Control Engineering Vol 7.

[30] Uchida, S., Sakoe, H., 1998. A monotonic and continuous two-dimensional warping based on dynamic programming, in: Proceedings. Fourteenth International Conference on Pattern Recognition (Cat. No. 98EX170), IEEE. pp. 521–524.

[31] Utz, H., Neubeck, A., Mayer, G., Kraetzschmar, G., 2002. Improving vision-based self-localization, in: Robot Soccer World Cup, Springer. pp. 25–40.

[32] Vidal, J., Lin, C.Y., 2016. Simple and robust localization system using ceiling landmarks and infrared light, in: 2016 12th IEEE international conference on control and automation (ICCA), IEEE. pp. 583–587.

[33] Vinotha, K., 2014. Bellman equation in dynamic programming. Int. J. Comput. Algorithm 3, 277–279.

[34] Wang, W., Luo, Z., Song, P., Sheng, S., Rao, Y., Soo, Y.G., Yeong, C.F., Duan, F., 2017. A ceiling feature-based vision control system for a service robot, in: 2017 36th Chinese Control Conference (CCC), IEEE. pp. 6614–6619.

[35] Zhang, C., Zhang, X., 2018. Visible light localization using conventional light fixtures and smartphones. IEEE Transactions on Mobile Computing 18, 2968–2983.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

A Case Study of Testing an Image Recognition Application

Chuanqi Tao[a,*], Dongyu Cao[a], Hongjing Guo[a] and Jerry Gao[b]

[a]Nanjing University of Aeronautics and Astronautics, China and [b]San Jose State University, USA

ARTICLE INFO

Article History:
Submitted 3.1.2021
Revised 6.1.2021
Second Revision 8.1.2021
Accepted 9.30.2021

Keywords:
Image Recognition
Testing AI Software
AI Software Quality Validation

ABSTRACT

High-quality Artificial intelligence (AI) software in different domains, like image recognition, has been widely emerged in people's daily life. They are built on machine learning models to implement intelligent features. However, the current research on image recognition software rarely discusses test questions, clear quality requirements, and evaluation methods. The quality of image recognition applications becomes more and more prominent. A three-dimensional(3D) classification decision table can help users to conduct classification-based test requirement analysis and modeling for any given mobile apps powered with AI functions in detection, classification, and prediction. This paper presents a case study of a realistic image recognition application called Calorie Mama using manual testing and automation testing with a 3D decision table. The study results indicate the proposed method is feasible and effective in quality evaluation.

1. Introduction

With the rapid development of big data analysis and artificial intelligence technology, AI software and applications have been widely accepted in our daily life including business, education, social media and so on. At present, AI software and applications are based on the most advanced machine learning models, and various artificial intelligence features are realized through large-scale data training.

The most important implementation of Artificial Intelligence is the imitation of human interactions—vision. Nowadays, there is an abundance of digital images captured by high-quality equipment. Most images are captured with phones. Artificial Intelligence is often used to process these images to extract knowledge, categorization, and labeling along with other advantages. Typical applications of image recognition include object recognition, face recognition, text parsing.

Detecting bugs and errors in software can be very costly. Sometimes bugs can be even deadly if it is a real-time application of software, such as some software that testing the software is very important to verify that the is used to help with surgeries in the hospital. Therefore, the product meets requirements and specifications. Software testing ensures the correctness, integrity, and high quality of the software by checking errors or bugs and fixing them in the initial design.

This paper focused on testing an image recognition application called Calorie Mama utilizing both manual testing and automation testing. Calorie Mama is a smartphone app that runs on Android and IOS devices. It uses deep learning to recognize food from food images and track nutrition based on the food in the image. It calculates the calorie based on that. We evaluated the performance, correctness, and quality of the app using both manual testing and automation testing.

This paper is written to provide our perspective views on image recognition software testing and quality evaluation. The paper is organized as follows. Section 2 discusses the review of AI software testing and image recognition. The third part elaborates methodologies of manual and automation testing. Then, the fourth part shows a case study of testing Calorie Mama APP using these two methods and presents the comparative results of test efficiency and coverage. At last, section 5 gives the conclusion.

*Corresponding author
Email address: taochuanqi@nuaa.edu.cn
ORCID: 0000-0002-0698-7307

DOI reference number: 10-18293/JVLC2021-N2-194

2. Related Work

An in-depth research is conducted in the field of AI testing. This research helped us in choosing our methodologies and impacted our testing approach as a whole.

Gao et al. [1] explained the various testing methods of the AI software testing wherein the authors mention context classification modeling, input and outcome classification modeling and decision tables. Various functional and non-functional quality parameters such as program correctness, system operations, performance, reliability and scalability are discussed to better understand the concept. In addition to this, the authors discuss the issues and challenges of AI testing. AI testing can be costly and time-consuming. There is a lack of adequate models and well-defined standards.

According to [2], AI can apply methods on data for software testing purposes like classifications, regression, clustering and dimensionality reduction. The paper also discusses the testing coverage containing requirement analysis, test planning, test development and execution.

In terms of test case generation, Zhu et al. [3] proposed a new method called datamorphic testing, which consists of three components: a set of seed test cases, a set of datamorphisms for transforming test cases, and a set of metamorphisms for checking test results.

AI software testing is different from traditional software testing, because AI software is characterized by dependence on big data, difficulty in predicting all application scenarios, and constant self-learning from past behavior. King et al. [4] discussed the issues and challenges in software testing. According to the authors, non-determinism is a huge issue. The same input to the system can produce different outputs. Testing has fuzzy oracles, i.e., determining the correctness can be a challenging task. The other challenges of testing include security, performance and scalability. Marijan et al. [5] stated that traditional systems have a fixed behaviour as they execute a set of rules and are typically pre-programmed. ML-based systems exhibit non-deterministic behavior as they use prediction algorithms or so. The quality parameters included correctness, robustness and reliability.

Metamorphic testing (MT) is a property-based software testing technique, which has been leveraged in many domains for addressing the test oracle problem and test case generation problem [6]. Chen et. al [7] introduced MT in 1998, which has been an effective AI-based software testing approach. MT has been applied to autonomous driving system [8,9], machine translation [10,11], ML classifiers [12,13], Google map App [14], search engines [15], facial age recognition software [16], and object detection system [17], etc., all of which have achieved good results.

GUI testing is also important in AI software testing. Rauf et al. [18] discussed the issues and challenges related to testing applications with a user interface. According to the authors, it becomes difficult to address the large number of states of GUI. Also, it is difficult to generate test cases each time the GUI changes. The testing can be platform-specific and thus shows a limitation. The methods prescribed and used in [18] did not address the problem of a huge number of states that even a small application's UI can have and thus can lead to a number of test cases. This paper has used particle swarm optimization (PSO), partition testing based on particle swarm optimization (PSO), test case minimization using an artificial neural network (ANN), Bayesian Network (BN).

King et al. [19] discussed the testing methods of Replication with Validation (RV) and Safe Adaptation with Validation (SAV). While testing a conventional application, we formulate creative test cases, manually explore the product, or write automated test scripts, testing AI-based products that focus on data and analytics. Testing supervised ML has two major phases: training validation and relevance testing. Training data is tested in the training validation phase which is verified during the validation phase. A genetic algorithm is used to generate test cases to cover all DU-pairs in [20]. As per the genetic algorithm, BP model is used which uses BP neural network.

Besides, other methods have been used to test AI software. The testing methods of RAP (Reconfiguration Automation Project) and FEID timeline set up were discussed in [21]. The automation of the several phases of the flight software testing procedure is the actual idea behind RAP. Also, it introduced Artificial Intelligence into the Space Shuttle flight software testing. Different models of AI systems were discussed in [22]. The authors discussed building testable AI systems, limiting the AI system to propositional logic and intervening variables in reducing testing. Ramanathan et al. [23] used symbolic decision procedures coupled with statistical hypothesis testing to validate machine learning algorithms for studying the correctness of intelligent systems. They also used an algorithm to analyze the robustness of a human detection algorithm built using the OpenCV open-source computer vision library.

In the field of image recognition, most researchers focus on recognition algorithms. Girshick et al. [24] proposed the R-CNN algorithm, which added selective search operations to the CNN network to identify candidate regions. The algorithm first divides the candidate regions of the input images and then extracts the characteristics of the candidate regions through the CNN network model for classification and recognition. He K et al. [25] proposed the SPP-Net algorithm, which reduced the process of image normalization and solved the problem of image information loss and storage. Girshick R [26] proposed the Fast R-CNN algorithm, which refers to the Region of Interest (RoI) and the

multi-task loss function method, and replaces SVM classification and linear regression with Softmax and SmoothLoss to realize the unification of classification and regression and reduce the disk space.

Moreover, Redmon et al. [27] proposed the YOLO algorithm, which can identify the categories and locations of multiple items in an image at one time, realizing end-to-end image recognition. First, the YOLO algorithm meshes the input images, calculates the confidence degree and classification probability of the existence of target objects in each grid, and removes the mesh without target objects by threshold. The YOLO algorithm runs fast, but has low recognition accuracy. Liu et al. [28] proposed SSD algorithm, which is a multi-target detection algorithm to directly predict the target category. It first extracts the feature map through the CNN network model, and then carries on the regression classification to the feature map. At the same time, SSD algorithm adds a multi-scale feature graph function, which can return the candidate boxes of different sizes on the feature graph of different levels, detect targets of different sizes, and improve the recognition accuracy.

However, the evaluation of image recognition system is relatively less but important. In [29], an implementation of Yolo-v2 image recognition and other test benches for a deep learning accelerator were presented. They converted the Yolo-v2 software to 16-bit floating point version and used it in the simulation and FPGA experiment during the chip development. Several other testbenches were designed and used to test various networks.

In [30], Yu et al. presented an image-driven tool, namely LIRAT, to record and replay test scripts across platforms, solving the problem of test script cross-platform replay for the first time. LIRAT recorded screenshots and layouts of the widgets, and leverages image understanding techniques to locate them in the replay process.

3. Methodology

The testing methodologies of the software ensure that the software meets the client's requirements. To do this, different types of strategies are used in accordance with the application to be tested. Every test methodology consists of an objective, method and results. A clear understanding of these terms is needed to conduct thorough testing of the application. We have used two different testing techniques to test image recognition applications using manual testing and automation testing.

3.1 Manual Test

The following are the test methods that we can use to efficiently and elegantly complete the software test process and ensure good quality.

1) Equivalence Partitioning Method

Equivalence class partitioning (ECP) is one of the software testing techniques that divides the data that is obtained as input of software into partitions of equivalent information with the help of which test cases are systematically derived. By principles, test cases are designed to provide coverage to each of these partitions once or at least once. The following table is an example of an equivalent partition for tree detection.

Table 1: Equivalence partitioning of tree detection

Equivalence Partitioning	
(1) The height of the tree	A) >20ft
	B) <20ft
(2) Picture completeness	A) The whole tree
	B) The trunk of the tree
	C) The root of the tree
	D) The branch of the tree
(3) Flowers	A) Without flowers
	B) With flowers

2) Boundary Value Testing

Boundary value analysis refers to the testing technique where tests are designed for validating software behaviors and functions by focusing on boundary values for each boundary in the system.

3) Category-Partition Testing

The Category-partition method is to divide the input domains of a component into N different disjoint partitions, and then select one value from each domain, combine them as one single test case. For example, it is used for vehicle detection and we choose to define the category of vehicle as follows:

A. Vehicle Type

1. Motor 2. Bus 3. Truck 4. Car

B. Vehicle Direction

1. Front 2. Side 3. Back

The result of the test cases is shown below.

Table 2: Test cases of the vehicle detection

Test Case #	Test Case	Description	Result
1	A1B1	Front of Motor	**Fail**
2	A1B2	Side of Motor	Pass
3	A1B3	Back of Motor	Pass
4	A2B1	Front of Bus	Pass
5	A2B2	Side of Bus	Pass
6	A2B3	Back of Bus	Pass
7	A3B1	Front of Truck	Pass
8	A3B2	Side of Truck	Pass
9	A3B3	Back of Truck	Pass
10	A4B1	Front of Car	Pass
11	A4B2	Side of Car	Pass
12	A4B3	Back of Car	Pass

4) Scenario-Based Testing

Scenario-based testing is one of the testing methods that consists of the business process flow tested end to end. Some test steps should be written in a way that completes and validates the positive flow of an application.

5) Decision Table

Testing is one of the best ways to deal with various combinations of input that produce different kinds of results. This is also referred to as Cause-Effect table. It provides a systematic way of stating complex business rules, which is useful for developers as well as for testers.

3.2 Automation test

Automation testing uses different types of tools, scripts, and software to perform test cases by reusing predefined actions. It is more reliable if the scripts are well written, because the machine can perform related tasks without error. Moreover, it is faster than manual testing which can be used when we have something repetitive to test.

It is impossible to automate all test cases. Therefore, we need to consider the test cause that can be automated. Different testing reasons can be automated, such as repetitive tasks and capturing results rather than manually collecting data and creating graphs. We can create a tool to capture the results for us so that we can save time and effort. In addition, the task of data entry also needs to be automated so that it is done automatically without having to enter the data manually or write it to the archive form.

We need to identify and create an automation plan by identifying the goal of the automation test. We need to know the type of test we want to do. After that, we will select the right tool that will help us with testing. It is important to pick the right tool to get a good result of automated testing. After selecting the right tools, we need to know the scope of the automation by selecting which tests to automate. It could be the features that are remarkable for the business, scenarios (which have a big amount of data), the technical feasibility of which business components are used, or the complexity of the test cases.

We usually use Appium as the main test framework. The main advantage of Appium over other test frameworks is that it is open-sourced and can be used to test native mobile apps. Another advantage of picking Appium is that it can be used across different platforms (mac and windows) and can test against various mobile operating systems such as Windows, iOS and Android. Appium comes with vendor-provided automation frameworks.

4. A Case Study

4.1 Test Setup

This paper took the test Calorie Mama APP as an example, using manual testing and automated testing respectively as seen in figure 1.

The Calorie Mama is an app designed to help the user achieve the weight goal he/she sets. In this app, one key function to do food tracking is to let the user take a picture of the food, and the app would recognize the food contents in the picture and display the food calories in the meal.

The test data is a mix of various sources: images from Google, images clicked in real-time using a smartphone camera. The experiments were performed with a high-resolution and high-quality camera.

Test coverage in software testing measures the factors including information about which part of the program is executed and how much code is utilized when running the test suite. Some of the benefits of test coverage are listed below. It assures the quality of the automation test and helps in identifying the exact portion of the code utilized. It makes sure that time, cost and complexity are under control and also identifies the gaps in requirements, test cases and errors.

The coverage of functional tests depends on the design of test cases. If the test cases are fully covered, the coverage of functional tests will be high. If N is the total test cases and M is the number of test cases for execution, the function-based test coverage percentage will be calculated as follows:

Test Coverage = M/N *100%

This paper focuses on function-based test coverage.

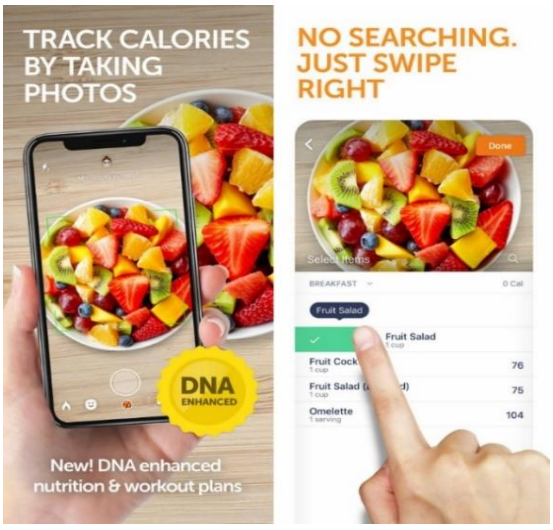


Figure 1: Information about Calorie Mama APP.

4.2 Test Experiment

(1) Manual Test

In this approach of manual testing, we selected conventional decision tables to test. A decision table is a table with various conditions and their corresponding actions. It is divided into four parts, condition stub,

action stub, condition entry, and action entry. We use decision tables to test manually from two aspects, namely detection of non-food and food items.

1)Detection of non-food items: Different non-food items are input into the Calorie Mama APP and the results were shown on the user interface. A summary of the detection of the non-food items can be seen in the following decision table. Bold content indicates that the test failed.

As we can see, the condition stub is designed as two conditions, including the state of the Internet and access to the Camera, which is essential for the image recognition software. When not turning on WIFI or Cellular, and not allowing access to the Camera, image recognition will not work. Besides, the application detected artificial pumpkin and artificial cake as food items. In contrast, it could not correctly identify the butter block. As a result, it failed in some of the cases.

Table 3: Decision table of the non-food items

	Turn on WIFI or Cellular	Allow access to Camera	Food item	Detected as food	Not detected as food
R1	T	T	Pen	F	T
R2	T	T	Apple	T	F
R3	T	T	**Artificial Pumpkin**	**T**	**F**
R4	T	T	**Butter Block**	**F**	**T**
R5	T	T	Banana	T	F
R6	T	T	Chicken Wings	T	F
R7	T	T	Clarified Butter	T	F
R8	T	T	**Artificial Cake**	**T**	**F**
R9	F	F	Glass of Water	-	-

2)Detection of food items: We divided the generic term of food items into four categories which are Indian cuisine, raw fruits and vegetables, a variety of apples and eggs, and food items in different backgrounds.

We sampled some food items under the Indian food category and fed the images to the application. The application was able to recognize some of the food items while it failed in many as seen in the table below.

Table 4: Decision table of Indian cuisine

	Turn on WIFI or Cellular	Allow access to Camera	Food item	Exact detection or correct choices	Offered wrong choices
R1	T	T	Lentils	T	F
R2	T	T	Sev	T	F
R3	T	T	**Potato Capsicum**	**F**	**T**
R4	T	T	**Okra**	**F**	**T**
R5	T	T	**Prawns &Okra**	**T**	**F**
R6	T	T	**Rice&Fish**	**F**	**T**
R7	T	T	Lamb Curry	T	F
R8	T	T	**Mixed Lentil &Rice**	**F**	**T**
R9	T	T	Vegetable Biryani	T	F
R10	T	T	Samosa	T	F

For raw fruits and vegetables, the application was given an input of raw fruits and vegetables. It recognized a majority of the food items but failed in a few cases as seen in table 5 below.

Table 5: Decision table of raw fruits and vegetables

	Turn on WIFI or Cellular	Allow access to Camera	Food item	Exact detection or correct choices	Offered wrong choices
R1	T	T	Apple	T	F
R2	T	T	Fig	T	F
R3	T	T	Mango	T	F
R4	T	T	Okra	T	F
R5	T	T	Horse Radish	T	F
R6	T	T	**Oppo Squash**	**F**	**T**
R7	T	T	**Bitter Gaurd**	**F**	**T**
R8	T	T	Lettuce	T	F
R9	T	T	**Mustard Greens**	**F**	**T**
R10	T	T	**Loquat**	**F**	**T**

For a variety of apples and eggs, we decided to test the application under different varieties of the same food items. For this particular case, we considered the different varieties of apples. As shown in table 6, the application failed to recognize a majority of the apple varieties.

Table 6: Decision table of apples

	Turn on WIFI or Cellular	Allow access to Camera	Food item	Exact detection or correct choices	Offered wrong choices
R1	T	T	**Cortland**	**F**	**T**
R2	T	T	**Gala**	**F**	**T**
R3	T	T	**Golden delicious**	**F**	**T**
R4	T	T	Granny Smith	T	F
R5	T	T	**Fuji**	**F**	**T**
R6	T	T	**Honey Crisp**	**F**	**T**
R7	T	T	**Macintosh**	**F**	**T**
R8	T	T	Red Delicious	T	F
R9	T	T	Dry Apple	T	F

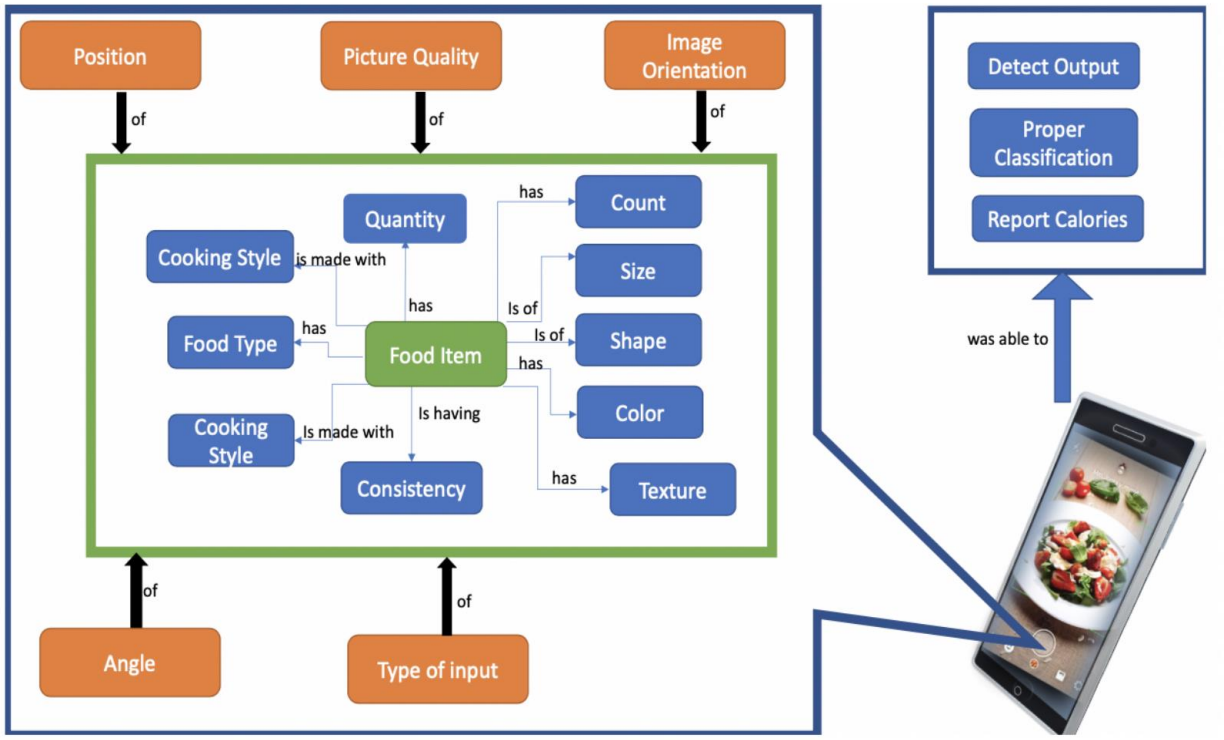


Figure 2: A logical data model.

Finally, for food items in different backgrounds, the background of food is a very important aspect and we decided to test the application with images of food items with different backgrounds. As seen in table 7, the Calorie Mama application was able to correctly recognize the food items when given inputs with red, blue, and wooden backgrounds. However, the application detected wrong when the egg is in a tray.

Table 7: Decision table of food items in different backgrounds

	Turn on WIFI or Cellular	Allow access to Camera	Food item	Exact detection or correct choices	Offered wrong choices
R1	T	T	Blue Back-ground	T	F
R2	T	T	Red Back-ground	T	F
R3	T	T	Wooden Background	T	F
R4	T	T	Egg in a bowl	T	F
R5	T	T	Egg on a plate	T	F
R6	T	T	Egg on a pan	T	F
R7	T	T	Egg in the glass	T	F
R8	T	T	Egg in a jar	T	F
R9	T	T	**Eggs in a tray**	**F**	**T**

After conducting the manual testing, we experienced its various drawbacks, and it is time-consuming. Also, load testing and performance testing are not possible under manual testing. Besides, regression test cases are very costly. Due to these drawbacks, we decided to shift to automation testing.

(2) Data Modeling

The three-dimensional (3D) classification decision table is influenced by the concept of conventional decision tables to conduct classification-based test requirement analysis and modeling for any given mobile apps powered with AI functions using a 3D tabular view.

As seen in figure 2, a logical data model is created after brainstorming and observing the various possibilities for the input image of food along with the context in which the image was clicked. This information was further utilized to create a 3D classification table.

The major testing focus for a 3D classification table is the mappings among classified disjoint context conditions, classified input selections, and classified AI function outputs. These mappings are known as image recognition function classification rules. Each of them represents the conjunction among three different views.

Test case design and generation based on a 3D classification decision table must cover these image recognition classification rules. Adequate image recognition function testing coverage could be assessed. Next, we introduce the construction of each one-dimensional model in the 3D decision table.

1) Input Modeling

The input classification refers to the parameters and their values that represent different test case scenarios.

Each parameter has multiple possible values which when combined with context values gives us the final set of test cases. The following figure shows Calorie Mama's input classification tree, which contains information about the type of food being clicked, such as what the food is, and the physical appearance of the food, such as quality, size, shape, consistency, etc.

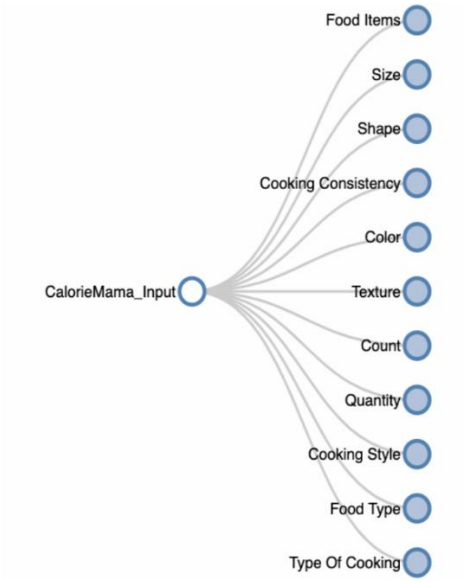


Figure 3: A sample input classification tree.

2) Context Modeling

The context classification tree contains information about the image context. It is basic information about the image itself and not specifically about the item in the image. For example, the context classification tree contains information like if the image is blurry or not well illuminated, what is the angle of the camera while clicking the image, if the image is rotated or so, etc. The following figure shows Calorie Mama's context classification tree.

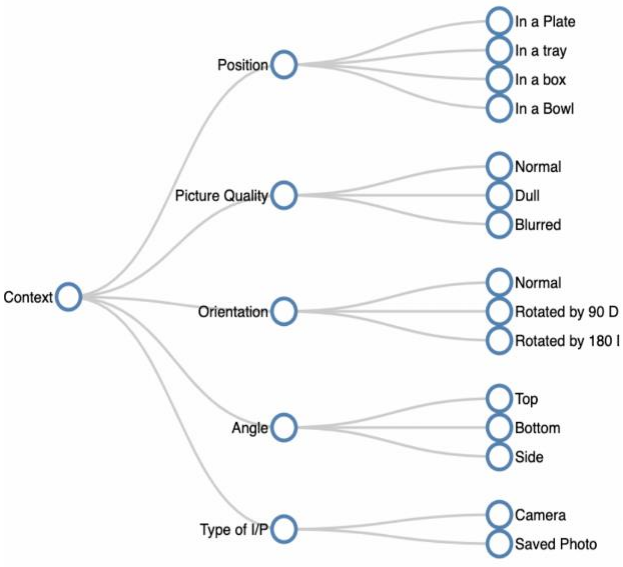


Figure 4: A sample context classification tree.

3) Output Classifications

The output classification tree contains information about the output. Various parameters regarding the output obtained from the application will be considered. This can be modified based on the requirements and results expected from the application. The following figure shows Calorie Mama's output classification tree.

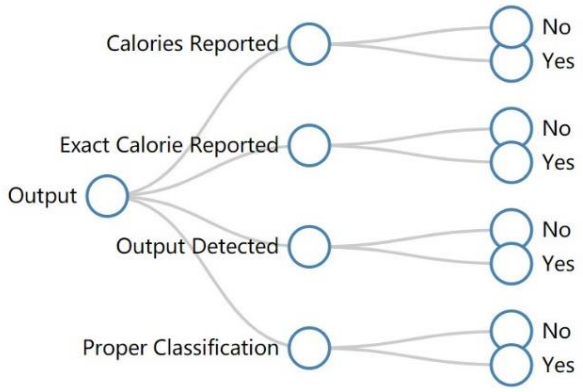


Figure 5: A sample output classification tree.

(3) Automation Test

After data modeling, we performed automation testing with minimal human assistance on top of the model. Automation testing can increase coverage for test data and come up with more concluding test results for the selected mobile app. We used Appium as an automation tool to perform automation on the mobile app. Appium acts as a server that launches the app into the simulator or a real device and can access the elements for processing the actions triggered by the automation script which we wrote in Java.

Steps to perform the automation were:

1) Install Appium server.

Appium is an open-source test automation framework used to run automation scripts on mobile apps. To install Appium on our development machine, we installed NodeJs and Node Package Manager which is required for installing Appium.

We had Homebrew installed already which is a smart package manager for installing packages on Mac machines. With the help of brew, we installed node by running the below command,

brew install node

This installs the Node and Node Package Manager.

Below command installs Appium.

npm install Appium

After the appium is installed, we can just run Appium command to launch the Appium server.

By default, Appium starts on port 4723.

2) Create the automation environment for Android.

After Appium is successfully installed and launched, the next step is to create the automation environment for the mobile operating system which we are using to automate our mobile app. This section discusses creating an automation environment on Android while the next section takes iOS into account.

To create an automation environment for Android, we install Android Studio.

Android Studio enables us to create Android emulators with customizable hardware specifications. As we launch a new emulator, it will behave as a real device connected to the machine and we can actually use it to launch and automate our target app.

3) Create the automation environment for iOS.

We install XCode for creating an automation environment for iOS mobile app. iOS lets us create iOS simulators that behave exactly like an iOS mobile device.

We can launch iOS simulators and can change the specifications of the OS and hardware as required. We can run any iOS app on these simulators using XCode or launching simulators after they are once initialized by XCode.

4) Launch simulator/ Connect a real device.

Either we can connect our real device to run the automation scripts on our app or we can use the simulators. If we are using a real device, then we need to install Android Debug Bridge to get the device IDs for proceeding with the automation.

We need device IDs to enter in the script so that Appium can connect with the connected device.

5) Install Eclipse.

The next step after creating the automation environment is to install Eclipse, as Eclipse will be the used IDE to write the automation scripts.

Download link: https://www.eclipse.org/downloads/ Before installing Eclipse, we need to have Java Development Kit installed in our machine. Download JDK:https://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html

After JDK is installed, we install Eclipse and are ready to start with our automation.

6) Create a maven project in Eclipse to write and run the automation script.

We start automation by creating a Maven project in Eclipse. Maven is a build automation tool that is used to automatically install all the dependencies involved or required by our project. We don't need to worry about installing the dependent libraries one by one. In a maven project, we use the Project Object Model (pom.xml) where we write all the names of the required dependencies in a fixed XML format and then Maven sets the platform for us thereafter.

We provide the dependencies of Appium, Selenium, TestNG in the Project Object Model and then start writing the scripts. We use TestNG to run our automation tests. Soon after the execution of tests, test results are visible in the Eclipse console.

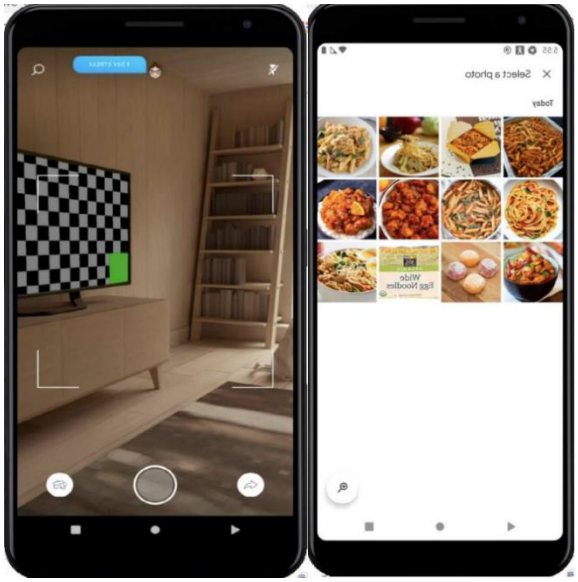


Figure 6: App can take input either through camera or from the gallery.

For the algorithm of the app automation, one image which is selected from the gallery of the phone is fed as an input into the target app as seen in figure 6, and the result of the execution is compared with the expected output. If the output from the target app is as expected, then the test case is displayed as passed or else failed. Also, when the app produces the output, more options, as provided by the app are taken into account. While showing the output to the user, there is an option to see more options from the suggestions coming from the app. The algorithm considers all those options as the output from the app and then decides if the test case is passed or failed.

4.3 Test Results

After applying manual testing and automation testing, we compare the coverages for both manual and automation tests. In manual testing, the coverage of the test case was limited due to timing. It was difficult to cover a larger set of data without the use of tools or scripts. On the other hand, automation testing has higher coverage because the tools and script helped us to cover more test cases. Figure 7 below shows that in automation testing we were able to cover more test sets of data than the manual testing over the same time. Approximately, in the automation testing, we were able to cover twice of what we covered in the manual testing.

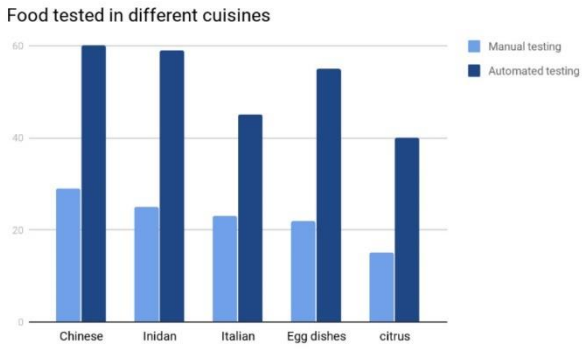


Figure 7: Test coverage for manual and automation test.

The app was able to detect objects, recognize them, and classify them with its name. However, it does not tell the count or sub-classification of the food item. Moreover, testing Calorie Mama App, required a lot of time to do both manual testing and automation testing. Manual testing needs to take more time to generate all decision tables, analyze different test causes and test manually. On the other hand, in automation testing, we spend days getting the script working correctly and program it to do the testing automatically.

The following figure shows the results of the manual testing and automation testing of the Calorie Mama APP. In manual testing, the total test food item across different cuisines was four hundred items and each cuisine has eighty food items. For example, of the 80 Chinese cuisines, 26 were detected as errors and 54 were detected as passes. As a whole, 132 of them were wrongly detected they were bugs in the app. This gives us a 33 failed percentage and the passing percentage is 67. The diagram below shows the failing and passing results.

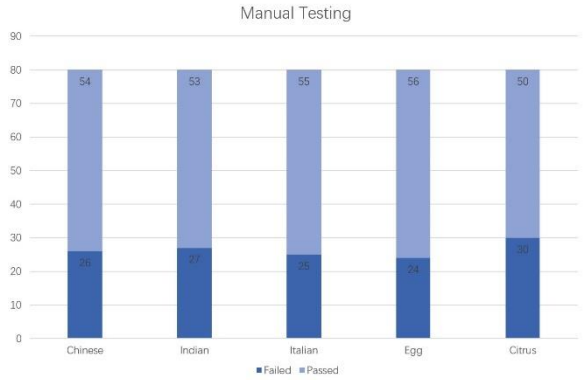


Figure 8: Manual testing.

In Automation testing, we tested four hundred different images in different cuisines similarly. We found out that out of the 400 images, 175 failed and 225 passed. This gives us a failure percentage of 43.75 and a passing percentage of 56.25 as shown in figure 9.

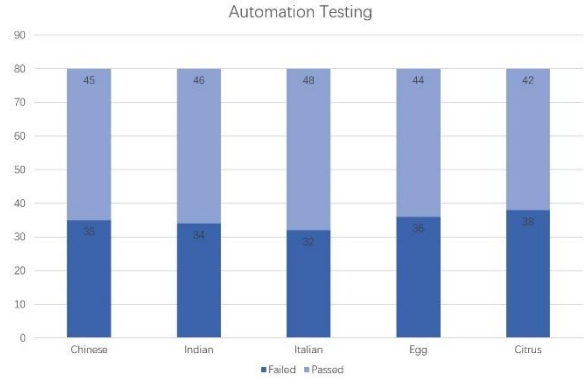


Figure 9: Automation testing.

Comparing the manual testing with automation testing, we can see that the errors that were found in the automation testing are higher than the errors that were found by the manual testing because the automation test allows us to test different inputs in a short time.

Also, in manual testing, it is more likely to make human mistakes because doing repeated tasks over time generates more errors by humans. Besides, manual testing can be expensive and time-consuming.

However, doing a repeated test using automation by writing a script and let the machine discover the error is more efficient. It helps them find errors without the need of performing redundant tasks. However, it needs talented and experienced people to do that, which is expensive. Besides, it is difficult to automate all kinds of testing where not everything can be redundant and reusable.

5. Conclusion

To sum up, we mainly leverage two methods to test the image recognition system, namely manual testing, and automation testing. We found that automation testing discovers more errors than manual testing.

In manual testing, the test is conducted by human testers inputting the use cases one by one, and observing the results. It is subject to human error; therefore, it is not one hundred percent accurate. On the other hand, in automation testing, the testers use tools and scripts to help them conduct the test among the image recognition software, which can save labor and time cost, thus improving testing efficiency.

For future work, we will evaluate more image recognition mobile apps with more datasets. Moreover, we plan to implement an automatic testing tool for detecting errors.

References

[1] Gao J., Tao C., Dou J. and Lu S., 2019, “Invited paper: What is AI software testing? and why,” 13th IEEE International Conference on Service-Oriented System Engineering, SOSE 2019, San Francisco, CA, USA, April 4-9, 2019, pp 27-36. doi: 10.1109/SOSE.2019.00015.

[2] Hourani H., Hammad A. and Lafi M., "The Impact of Artificial Intelligence on Software Testing," 2019 IEEE Jordan International Joint Conference on Electrical Engineering and Information Technology (JEEIT), Amman, Jordan, 2019, pp. 565-570. doi: 10.1109/JEEIT.2019.8717439.

[3] Zhu H., Liu D., Bayley I., Harrison R. and Cuzzolin F., "Datamorphic Testing: A Method for Testing Intelligent Applications," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 149-156. doi: 10.1109/AITest.2019.00018.

[4] King T. M., Arbon J., Santiago D., Adamo D., Chin W. and Shanmugam R., "AI for Testing Today and Tomorrow: Industry Perspectives," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 81-88. doi: 10.1109/AITest.2019.000-3.

[5] Marijan D., Gotlieb A. and Ahuja M. K., "Challenges of Testing Machine Learning-Based Systems," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 101-102. doi: 10.1109/AITest.2019.00010.

[6] Chen T. Y., Kuo F.C., Liu H., Poon P., Towey D., Tse T., and Zhou Z., "Metamorphic testing: A review of challenges and opportunities," ACM Computing Surveys 51(1), 4:1-4:27 (2018).

[7] Chen T. Y., Cheung S., and Yiu S., "Metamorphic Testing: A New Approach for Generating Next Test Cases," Technical Report HKUST-CS98-01, Department of Computer Science, Hong Kong University of Science and Technology, Hong Kong (1998).

[8] Tian Y., Pei K., Jana S., and Ray B., "DeepTest: Automated Testing of DeepNeural-Network-Driven Autonomous Cars," Proceedings of the 40th International Conference on Software Engineering (ICSE), pp. 303-314. Gothenburg, Sweden (2018).

[9] Zhang M., Zhang Y., Zhang L., Liu C., and Khurshid S., "DeepRoad: GAN-Based Metamorphic Testing and Input Validation Framework for Autonomous Driving Systems," 33rd IEEE/ACM International Conference on Automated Software Engineering (ASE), pp. 132-142. Montpellier, France (2018).

[10] Sun L. and Zhou Z. Q., "Metamorphic Testing for Machine Translations: MT4MT," 2018 25th Australasian Software Engineering Conference (ASWEC), 2018, pp. 96-100, doi: 10.1109/ASWEC.2018.00021.

[11] Pesu D., Zhou Z. Q., Zhen J. and Towey D., "A Monte Carlo Method for Metamorphic Testing of Machine Translation Services," 2018 IEEE/ACM 3rd International Workshop on Metamorphic Testing (MET), 2018, pp. 38-45.

[12] Murphy C., Kaiser G. E., Hu L., and Wu L., "Properties of machine learning applications for use in metamorphic testing," Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 867872. San Francisco, CA, USA (2008).

[13] Murphy C., Kaiser G. E., Hu L., and Wu L., "Properties of machine learning applications for use in metamorphic testing," Proceedings of the 20th International Conference on Software Engineering and Knowledge Engineering (SEKE), pp. 867872. San Francisco, CA, USA (2008).

[14] Brown J., Zhou Z., and Chow Y., "Metamorphic Testing of Navigation Software: A Pilot Study with Google Maps," 51st Hawaii International Conference on System Sciences (HICSS), pp. 1-10. Hilton Waikoloa Village, Hawaii, USA, (2018).doi: 10.24251/HICSS.2018.713.

[15] Zhou Z., Xiang S., and Chen T. Y., "Metamorphic testing for software quality assessment: A study of search engines," IEEE Transactions on Software Engineering 42(3), 264-284 (2016). doi: 10.1109/TSE.2015.2478001.

[16] Tao C., Gao J. and Wang T., "Testing and Quality Validation for AI Software–Perspectives, Issues, and Practices," IEEE Access, vol. 7, pp. 120164-120175, 2019. doi: 10.1109/ACCESS.2019.2937107.

[17] Wang S. and Su Z., "Metamorphic object insertion for testing object detection systems," Proceedings of the 35th IEEE/ACM International Conference on Automated Software Engineering (ASE '20). Association for Computing Machinery, New York, NY, USA, 1053–1065.

[18] Rauf A. and Alanazi M. N., "Using artificial intelligence to automatically test GUI," 9th International Conference on Computer Science Education, Vancouver, BC, 2014, pp. 3-5. doi: 10.1109/ICCSE.2014.6926420.

[19] King T. M., Arbon J., Santiago D., Adamo D., Chin W. and Shanmugam R., "AI for Testing Today and Tomorrow: Industry Perspectives," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 81-88.

[20] Ji S., Chen Q. and Zhang P., "Neural Network-Based Test Case Generation for Data-Flow Oriented Testing," 2019 IEEE International Conference On Artificial Intelligence Testing (AITest), Newark, CA, USA, 2019, pp. 35-36. doi: 10.1109/AITest.2019.00-11.

[21] DeMasie M. P. and Muratore J. F., "Artificial intelligence and expert systems in-flight software testing," IEEE/AIAA 10th Digital Avionics Systems Conference, Los Angeles, CA, USA, 1991, pp. 416-419.

[22] Liu G., Liu Q. and Zhang W., "Model-based testing and validation on artificial intelligence systems," Second International Multi-Symposiums on Computer and Computational Sciences (IMSCCS 2007), Iowa City, IA, 2007, pp. 445-449.

[23] Ramanathan A., Pullum L. L., Hussain F., Chakrabarty D. and Jha S. K., "Integrating symbolic and statistical methods for testing intelligent systems: Applications to machine learning and computer vision," 2016 Design, Automation & Test in Europe Conference & Exhibition (DATE), Dresden, 2016, pp. 786-791.

[24] Girshick R., Donahue J., Darrell T. and Malik J., "Rich Feature Hierarchies for Accurate Object Detection and Semantic Segmentation," 2014 IEEE Conference on Computer Vision and Pattern Recognition, Columbus, OH, 2014, pp. 580-587. doi: 10.1109/CVPR.2014.81.

[25] He K., Zhang X., Ren S. and Sun J., "Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition," in IEEE Transactions on Pattern Analysis and Machine Intelligence,2015,37(9):1904-1916.

[26] Girshick R., "Fast R-CNN," 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 2015, pp. 1440-1448. doi: 10.1109/ICCV.2015.169.

[27] Redmon J., Divvala S., Girshick R. and Farhadi A., "You Only Look Once: Unified, Real-Time Object Detection," 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016, pp. 779-788, doi: 10.1109/CVPR.2016.91.

[28] Liu W., Anguelov D., Erhan D., et al., "SSD: Single Shot MultiBox Detector," Welling M. (eds) Computer Vision – ECCV 2016. Lecture Notes in Computer Science, vol 9905. Springer, Cham.

[29] Kim C., Jeon I. S., Kwon Y., Kim H., Lyuh C. et al., "Implementation of Yolo-v2 Image Recognition and Other Testbenches for a CNN Accelerator," 2019 IEEE 9th International Conference on Consumer Electronics (ICCE-Berlin), Berlin, Germany, 2019, pp. 242-247. doi: 10.1109/ICCE-Berlin47944.2019.8966213.

[30] Yu S., Fang C., Feng Y., Zhao W. and Chen Z., "LIRAT: Layout and Image Recognition Driving Automated Mobile Testing of Cross-Platform," 2019 34th IEEE/ACM International Conference on Automated Software Engineering (ASE), San Diego, CA, USA, 2019, pp. 1066-1069. doi: 10.1109/ASE.2019.00103.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc/

Project Management and Learning Environments: a Case Study

Andrea Molinari

University of Trento, Dept. Of Industrial Engineering (ITALY)

Lappeenranta University of Technology, School of Industrial Engineering and Management (Finland)

andrea.molinari@unitn.it

ARTICLE INFO

Article History:
Submitted 3.1.2021
Revised 6.1.2021
Second Revision 8.1.2021
Accepted 9.30.2021

Keywords:
Project Management
Virtual Communities
E-learning

ABSTRACT

The paper analyses the addition of project management features of an LMS, a topic that LMSs and collaboration platforms have entirely ignored. Many project management-like platforms are available today, especially riding the media wave of agile methodologies. The question is not surprising in itself, as managing time, costs, and resources linked to the discipline of Project Management, is historically a problematic issue for the IT world from a cultural, technical, and organizational point of view. The problem becomes even more complex if applied to the management of learning projects, where we have time, constraints, deadlines, costs, resources like in any other project. LMSs and, in general, collaboration platforms do not include these features, forcing users to link external platforms or adapt existing features (like simple to-do lists) to more complex Project Management tasks. In this work, we will present the tests conducted in a collaborative platform based on the metaphor of virtual learning communities. These tests were on a component developed explicitly for managing projects, activities, and resources, integrated inside the LMS with all the other services (blog, forum, file sharing, calendar, reminders, etc.). The introduction of this new component within the system addresses the need to manage collaborative activities between learners, providing a tool for managing and controlling the progress of the activities assigned to the various community members.

1 INTRODUCTION

This paper presents some services provided inside a virtual community platform as extensions towards managing the users' time, specifically teachers, students, and administrative personnel involved in educational tasks. These services are rarely available (if any) inside Learning Management Systems (LMS) or are simple tasks lists, to-do lists, or academic tasks scheduled in some workflow systems. Functionalities that provide support to (education) projects are something significantly different. Project Management is a well-established discipline, where we can find additional standards, guidelines, and certification processes provided by worldwide organizations. The advent of web platforms claiming to provide Project Management services has changed the scenario, with a lot of competition in Project Management services. If we aim to provide a Project Management approach to the conduction of educational activities, we find different solutions: a) "traditional" Project Management platforms, fully equipped with standardized services for scope, time, cost, and resource management, but with a lack of support to collaboration and social processes inside project teams b) a lot of web platforms that support collaboration processes [1] and claim to support Project Management processes but that are not very efficient in this sense, lacking basic mechanisms like critical path method (CPM), timesheet cost and resource management, milestones, constraints, deadlines, etc. [2]

This paper explores what should be needed, in the author's opinion, inside a Learning Management System and inside educational contexts to support the many activities that fall into the definition of "project" [3]. We designed and realized a set of services oriented explicitly towards the Project Management concepts and methodological tool and added them to a virtual community platform (with all essential collaboration, social media-oriented services). From providing

DOI reference number: 10-18293/JVLC2021-N2-020

traditional collaboration, education-oriented services, the platform has been extended to a fully-fledged set of tools compatible with the most used Project Management standards, including task planning, costs, and resource management functionalities.

Since the advent of open-source LMSs, Technology-Enhanced Learning (TEL) is a consolidated research topic. A lot of tools and techniques are available for creating, delivering, and managing online educational paths with plenty of solutions for every educational institution. What is less known in the technological solutions that support our daily educational tasks is integrating with Project Management tools and techniques to manage tasks, resources, and costs within educational settings. We can find a plethora of platforms and cloud services available today, even for free, that support at different level activities related to a project. Implementing Project Management suggestions inside educational processes forces teachers or students to exit from the learning environment (for example, Moodle, Forma.LMS or Sakai LMSs), and use an external platform. This implementation, in turn, could be problematic as most of the items that are the subject of the educational project are inside the LMS, so the user is forced to duplicate the material. Then, if we consider the aforementioned trivial problem of document sharing, but we extend the analysis to other typical Project Management tasks (like checking deadlines), the situation is again complex: the educational platform and the Project Management platform are entirely disconnected.

We can mainly find inside LMS under the label "project" or "Project Management" is a sort of task-list or to-do list service that the user must manage directly. In a few other cases, the idea of managing projects is implemented with a set of steps of a predefined workflow, in some way linked to each other, but this is very limited even concerning elementary Project Management tools. Let's consider the world of education in its full complexity and what educators do every day. We can find plenty of activities that can be fully defined as a 'project'. Still, strangely software platforms that are so useful for educational purposes are not providing adequate support for this. We have decided to add a set of Project Management functionalities to our "Online Communities" system. "Online Communities" is a software platform created from scratch by our team, constantly updated with new services that our research group considers engaging in educational and, more generally, in collaborative contexts. The organization of the paper is the following: in Chapter 2, we will analyze the idea of projects in educational contexts, presenting how and where education and project management disciplines could intersect. Chapter 3 will briefly introduce "Online Communities", the collaboration platform where we added the services to support project management activities in educational contexts. In Chapter 4, an overview of the different services for project managers will be presented. Finally, Chapter 5 will explain a qualitative analysis with six experienced project managers who used our platform's various features and compared them with top project management software.

2. PROJECTS IN EDUCATIONAL CONTEXTS

The de-facto standard in the Project Management discipline [3] identifies what a "project" is: "a temporary endeavor undertaken to create a unique product, service, or result" [3], The discipline of Project Management [4] applies a set of different tools and techniques taken from different disciplines (CPM, breakdown structures, network techniques etc) to improve the use of various resources toward the accomplishment of a unique, complex, unique task, taking into consideration time, cost and quality constraints. After many years of mostly being ignored by the software industry (with some notable exceptions), nowadays, we can find on the market many different tools that project managers can choose to plan and manage their projects. Nevertheless, the application of tools and techniques derived from the standards is rare inside these platforms. Here the focus is mainly on extended functionalities of Agile and in general collaboration services, rather than focusing first on core services like critical path method (CPM), Critical Chain Method (CCM), PERT, etc. The perspective of using a collaborative platform as a Project Management tool is fascinating, but this should happen without compromises concerning what Project Management methods provide to millions of skilled project managers around the globe, thus limiting the power of Project Management concepts. Deadlines, cost management, task duration, resources, and many others are problems that everyone involved in project-related activities faces.

The next consideration relates to the presence of these tools inside another kind of collaboration software, i.e., LMS. If we look at educational contexts, there are plenty of academic tasks that are part of a "project" in the plain sense of the definition, enriched with many other aspects typically managed in project contexts. The many relations among students, between students and teachers, between students and the educational institution or external organizations (like stages), can be efficiently and profitably managed as projects. Moreover, using Project Management concepts and tools for students of any course would be a real growth in any direction their future professions will lead them.

Everybody talks about team working, tasks, milestones, deliverables, scope, risks, i.e., uses the typical jargon of Project Management Nevertheless, the application of this complex discipline has not been so widespread and is undoubtedly not applied extensively in education. According to several studies [5], even today, we have a wide range of tools and techniques available, projects are frequently out of time and out of budget [6]. The poor results shown by these (and other) researches in terms of success are subject to different interpretations. Among them, we can mention "Project Management continues to fail because included in the definition are a limited set of criteria for measuring success, cost, time

and quality, which even if these criteria are achieved simply demonstrate the chance of matching two best guesses and a phenomenon correctly" [6].

Project Managers, today, can choose among many techniques and software to plan and manage their projects. The widespread usage of network approaches, like Gantt charts, critical path method (CPM), etc., have simplified the planning and controlling steps. In contrast, Project Management software has reached a solid maturity level. The Work Breakdown Structure (WBS) has been adopted as a (graphical) tool to define the project's scope and delimit what should be done in the project, separating them by unwanted or unpaid requests. WBS is the hierarchical decomposition of the work to be executed by the project team to fulfill the project's objectives and make deliverables. It organizes and evaluates the overall scope of the project. Information for a WBS is taken from project objective statements, historical files of previous projects, and project performance reports. An appropriate WBS encourages a systematic planning process, reduces the possibility of omission of key project elements, and simplifies the project by dividing it into manageable units.

Another area heavily interlaced with Project Management and relevant for our argumentation is collaboration. Collaborative spaces are available within the project team to contribute to the success of the project's objectives. New generation tools of Project Management enable this functionality. The project is led and developed by the whole team, and each member has complete information about the project, with all the related documents. The project's progress is visible to everyone anytime, according to permissions granted to the subject. When the project manager is free from the routine tasks; s/he can put more effort into project vision and choose the direction for the project development. The authors of [7] discuss methods and tools for collaborative Project Management; if these elements are coped with the widely recognized collaborative nature of educational processes, we should expect a convergence of these two disciplines and the relative tools.

On the contrary, the only field where we have found the application of Project Management tools and techniques inside educational contexts is the production of learning objects [8]. Here the concept of Project Management is not focused on providing tools inside the LMS for the management of activities as a project, but rather on managing the creation of learning objects with the typical five phases of the lifecycle of a project (initiation, planning, executing, monitoring and controlling, closing). This means treating the production of learning objects using the project's lifecycle as stated in [3], but the LMS remains in the background with the traditional set of functionalities not equipped with Project Management functionalities.

In our opinion, educational processes in general (and not only the production of educational material) can profitably use the pillars of the discipline. Following this idea, we have integrated into our self-made LMS an entire set of Project Management functionalities. The management of tasks within an educational environment shows a series of constraints and issues that need to be managed with appropriate tools, like those supplied by Project Management. The pandemic we live in demonstrates that it is very important to provide a precise work plan for students who are not allowed to follow physical lectures. Our "Virtual communities" platform provides affiliated users a set of features strictly related to Project Management tools and techniques: a) define and manage projects and their scope b) prepare a fully-functional Work Breakdown Structure with predecessors and constraints c) implement the Critical Path Method in the calculation of start/finished dates and free/total slack d) assign resources to tasks and check their allocation e) assign and control costs associated to a project, with a combination of role-permission to a level that can adequately administer the security, confidentiality, and privacy of the activities.

The integration of these features inside a Learning Management system guarantees the increase of application fields for these platforms, allowing them to be used for traditional educational activities and more collaboration and cooperation-oriented tasks. The problem we see in today's collaboration platforms that claim to be Project Management-enabled platforms is precisely the approximation and imprecision of implementation of Project Management services. In our experience, this incompleteness causes users to start using the service and then abandoning it (and the platform consequently), or on the contrary, considering Project Management as the discipline of the colored sticky notes attached on a Kanban board. The world of education is impoverished in terms of Project Management tools and techniques. The proposed one could be an excellent way to improve the awareness of educational actors about how to manage their interaction with the institution. We use words like "educational projects", "educational tasks", "learning milestone", "educational deliverable" very frequently in our focused discussions. On the contrary, even inside modern LMSs, there are no accurate, native, theoretically-grounded services that could support educational actors in exerting their tasks under the umbrella of Project Management theories and tools.

3. PROJECT MANAGEMENT FUNCTIONALITIES IN LMS

The typical collaboration services available in enterprise platforms (like wikis, blogs, and collaborative planning tools. Together with file sharing (documents, reports, agendas, comments, etc. all these tools represent an apparent stimulus to consider an e-learning platform as a possible provider of support and services to PM. These services, like many others, are widespread both in collaborative environments and in e-learning platforms, as presented in [9][10]. Likewise, many different situations in educational settings could take advantage of PM services. Some examples can be the following:

- the management of a thesis assigned to a student is an actual project with tasks, milestones, deliverables, and costs (even if not directly sustained by somebody);
- a research project led by a teacher or researcher is, by definition, a project, involving again different resources, costs, deliverables, and milestones;
- an educational path and all the tasks that any participant has to manage is another example of a "project."
- a complex training path providing professionals with a certification at the end of the activities, with the respective assistance of external resources and tutors, is a project from the perspective of the organizing institution. Here we have a typical mix of educational needs (the LMS's most traditional services) and PM tasks
- we can certainly consider as a project the massive open online course (MOOC) initiative, with all tasks related to various phases of creation, marketing deployment, execution, support, and final certification. Both the institution that delivers the MOOC and the participant that has to perform tasks, in any case, must pay attention to milestones to respect deadlines.

The possibilities of using these tools and services inside LMSs is even stronger if we imagine using a Virtual Communities system not only for managing educational "communities", but also in larger collaboration contexts. Examples of this can be a research group, a recreation organization, a secretariat, a board of directors, a club, a sports team, etc. All these "communities" need services available inside LMSs (like document sharing, forums, wikis, FAQ, sync, and async communication, etc.). E-learning became so popular thanks to many factors, like network availability, multimedia, increased power of client workstations, flexibility, low costs, etc.. Still, the role of software platforms like Moodle™, Docebo™, Dokeus™, Sakay™, Webct™ is central. These platforms have proven to be effective in contexts not necessarily connected to academic education, posing the issue of the evolution of software platforms towards services that are not necessarily related to traditional academic tasks. Last but not least, the integration of e-learning (or collaborative) software platforms with the rest of the information system of the hosting organization represents clear evidence of the role of software platforms today in education.

From a meta-architectural point of view, e-learning platforms have based their pillars on the idea of "course" or "class". The meaning of this choice is that the primary container for relationships among users of the platform is a virtual place that resembles in some way what happens in any educational organization: collecting people in a (virtual) classroom. What emerged in past studies [9] and from our preliminary experiments is a need for a different funding paradigm for software platforms: the "community" or "virtual community". The virtual community is a container ready for didactic processes, but not only: research teams, recreation groups, friends, secretariats, the board of directors, colleagues, anything that could be an aggregation of people around the scope using virtual spaces on the Web. The application's core comprises some abstract entities, i.e., virtual communities as an aggregation of people to which some communication services are available to obtain specific objectives. "Online Communities" [10] is a space on the Web devoted to a collaboration objective, populated by people who communicate with each other using a series of communication systems. This approach could represent all the hierarchical relationships between different types of communities (such as faculties, didactic paths, master degrees, courses, etc.). The main characteristics of a virtual community could be the following:

- a community is a composition of services for a virtual space of interaction involving end-users for that community;
- the services are general applications that enable the users to communicate synchronously and asynchronously, to publish contents, exchange files, coordinate events, etc.;
- a manager of the community activates the potential services of a community according to the needs, and the users of a community can use them with different rights and duties;
- communities can be aggregated into larger communities with hierarchic mechanisms and infinite nesting levels;
- the communities can be aggregated arbitrarily into larger communities disregarding the possible position of a hierarchical structure;
- all users are recognized.

The addition of Project Management services inside e-learning came mainly from the experience of the team in the techniques of Project Management, on the one hand, but also from everyday tasks: consider, for example, as part of learning community college, the need for a teacher to coordinate several undergraduates involved in the long task of drawing up their thesis. The individual, the professor, or those shared between them have often intertwined/ associates and impose the need to manage time, deadlines, relationships, and mutual dependencies. More complicated is the situation on the teacher's side, where s/he could have more thesis to follow, so more projects of this type to manage. We, therefore, believe that the lack of a tool of this kind can be solved naturally with valuable tools for planning and managing existing projects, but these:

- do not integrate platforms
- on average, they are complex
- they are much more appropriate for people with specific expertise in the complex and multifaceted discipline of Project Management

The approach we have followed in the elicitation of requirements wanted to incorporate functional needs very different from one another so that you can create many services with them, most of the time very similar but individually functional on their own for the end-user:

- Personal To-Do List: in this case, the list of tasks of the project is not shared with other users, but is personal reminders of the activities to be carried out in a given period. Tasks can include everything, from social activities to complex projects.
- Brainstorming: here, we can use the list of tasks for brainstorming with people who do not have the chance to meet either physically or through video conferencing. Indeed, given the opportunity to participate in forums, contribute to wikis and glossaries, write a FAQ, attach files, etc., users can add their opinion in various ways to the single tasks, simply reusing the mentioned services that are already available inside the platform. This integration with many services already available is not completed for all services. It will be developed within the service task list/project management evolution roadmap in the following months. We implemented the integration with file management services, which is a crucial service transversal to everything inside the platform. We will provide some examples of this integration in the following pages.
- Organization of meeting: planning a meeting, sending invitations to participants, defining the date, assigning documents to the meeting, all this can be easily implemented inside "Online Communities) via the use of one task named, for example, "Meeting". Subsequently, participants can attach their feedback or materials to this. Then, another simple integration is the "online presence" service, to see in the moment of the conference call, if the person is available or not.
- Bulletin-board system: the services related to task management can be used as a simple bulletin-board system, also shared by multiple users. You can share files and exchange messages with the planning of tasks. In the bulletin boards, tasks can be set by the task leader, updated by the users that have the proper permissions
- Project planning, execution, and monitoring: this is a full-fledged service where you can create and redefine the structure of the task gradually. The inspiration of this service is the category of software devoted explicitly to implementing PM tools and techniques like WBS, Gantt, PERT, CPM, milestones, constraints, deliverables, etc. With this service, it is possible to assign responsibilities to users, and then associate them with each task. During the project's execution, the resource's contribution is updated with a % of completion. This indicates whether the job is completed or not and to what extend. Aggregating all this information, the project manager can check the degree of progress of each activity under the project. For every task of the project, there is the possibility of attaching files

The shortlist just given is not intended to analyze all the possible scenarios of use of the PM services but can show only some examples. As can be seen from the list, the proposed uses have entirely different nature and are not targeted to manage activities with a rigid structure defined as "a-priori". We can even note how the examples discussed above can be transformed with extreme simplicity from case to case. For example, a To-Do List can change very simply into brainstorming or even into more sophisticated projects. Everything happens through the simple assignment of some users to a given Task and the addition of some files.

4. IMPLEMENTING PROJECT MANAGEMENT SERVICES INSIDE AN LMS

The idea of implementing Project Management services inside learning contexts benefitted a lot from the availability of "Online Communities", the virtual space dedicated to each community (what we call the virtual community). This collaboration space provided services to the users, so it was simply a matter of creating new Project Management (PM) discipline services. However, being aware of all the platform provides and integrating the PM services with this. Another fundamental factor from the virtual community concept that we used in assembling the new services was the concepts of 'roles', 'rights', and 'permissions'. Those are assigned to each user for each separated community, allowing a fine-grained, sophisticated way of managing and controlling "who does what" on a specific task.

One of the implementation choices was how to consider the project and its component activities. A project is created as a tree of tasks with arbitrary depth, whose nodes and leaves represent tasks, milestones, or deliverables. To manage the structure mentioned above, we decided to implement a Work Breakdown Structure (WBS), the standard project management tool used for the analytical breakdown of a project at different levels of detail. Each activity has a numerical index that determines the level, composition, or membership hierarchy.

The Project Management services contain some sophisticated features typical of top software solutions, strictly related to the critical path method (CPM) and its calculation mechanisms. These options have been added in the recent version on the platform as a substantial improvement requested specifically by skilled users complaining about the lack of these options. These options (fig.1) allow us to obtain the same results we can get, in terms of calculation of time, start, finish, etc., with Project Management professional tools.

The options are:

- defining a project calendar with working and non-working days
- specifying the start date of the project from which the CPM will start to calculate all start/finish dates according to the predecessors
- default resources: this allows you to set the resources you want to automatically assign when a new task is created
- Milestone: allows the project to use "Milestone", i.e., tasks that have a duration equal to zero days and represent crucial moments in the project to be highlighted to stakeholders.
- Project visibility: this option sets the project visibility

for the involved resources. It can be limited only to the assigned tasks to those resources or when tasks are complete.
- Task completion: this option sets the possibility for the created project to have task completion confirmed by a manager. Therefore, a task will be completed when all its resources set completion to 100%, or when verification by a manager is required.
- Summary task: this allows the project to have the robust feature typical of the WBS, i.e., tasks that summarized all data (start, finish, costs, etc.) of the subtasks. If you are using the project in the CPM mode, it will not be possible to create dependencies from and towards summary tasks. This is a simplification compared to full-fledged software, as these platforms implement the CPM also for the summary tasks. In this situation, the calculus is more complicated for the calculation engine, and we have decided to skip this version.
- Estimated duration: this is instead an exciting niche feature available only in the most advanced software, but we consider it very important. This option allows to set the duration of tasks in estimated days, i.e., days of duration considering a linear calendar of 365/366 days per year without non-working days. For example, a task with a duration "6?d", is a task with 6 days estimated duration.

Another professional feature of the PM service is the possibility of setting, on each task or project, constraints and deadlines. A task has the following features:

- Status: The status indicates at what stage of development the project is. Son provided the states active (the task is in progress), inactive (the task is still in progress), pending, and completed.
- Priority: can be low, standard, and high. It is an indicator of the urgency of the execution of a task.
- Temporal constraints: An activity has three dates that mark its production cycle: start date (the start of the task), end date, deadline (maximum limit for completing the task). To allow the creation of milestones (used to indicate the achievement of the objectives set at the design stage) we let the three abovementioned dates to coincide.
- Percentage of completion;
    - Category: allow a subdivision of projects according to the subject
    - Description;
    - Attachment: you can attach any file that is available inside the file repository of the community.

Tasks and users can also be shared among different communities with the exact inheritance mechanism. On the one hand, users have an institutional role inside the organization and one or more functional roles in each community in which they participate. Examples of institutional roles are those of the classic academic institution (student, teacher, etc.) As examples of functional roles, we have a community administrator, participant, moderator, blogger, secretary, member, dean, writer, etc. Administrators can create roles freely, assign users with individual permissions to each service available in the platform and PM services.

We, therefore, decided to add an extra feature that takes advantage of the possibilities of the creation of a project inside a virtual community: a member of the virtual community can be a resource of the project (fig.3). In educational communities, this is the case very frequently, where the community itself exists because of the need to manage a project. A workgroup, for example, created with some students that have to perform a common educational task, with milestones, deadlines, and detailed WBS perfectly fits as an application scenario of the illustrated feature. We can also add external users, so avoiding limiting the management of the project to people enrolled in the community.

This feature, natural as it may seem, has been very complicated to implement because of the need to validate the users' actions inside the different parts of the portal. In this perspective, external users typically have minimal actions to perform inside the platform. At the same time, in educational projects, they can act as an essential stakeholder and thus need different permissions on the project. For example, imagine an enterprise tutor following a workgroup to support her ideas. S/he can be an external person, with supervisory tasks to perform on what the workgroup is doing. Still, at the same time, her high-level permission on that specific workgroup should be isolated from the rest of the portal.

The Project Management service provides the user a general activity for the project at level zero of the WBS, thus providing a consistent approach to creating both complex projects and simple reminder/to-do lists and following standard planning procedures commonly available in scheduling software. This particular task indicates the project's root, distinguishing it from the others only by the absence of a parent.

Starting from this summary task, the user can manage the rest of the WBS, together with the rest of the inputs (duration and predecessors) for the CPM engine (Fig. 4). This feature implies a set of best practices in planning a project, starting from a) the definition of a calendar, b) the start date of the project c) the tasks and their dependencies. With these inputs, our system can produce as output a) the start and finish date of all the tasks, b) the critical path, i.e., the tasks that have a total float equal to zero c) the finish date of the project. What we expect in a Project Management context is not so "normal" in educational contexts. For example, forcing thesis students to think about their thesis as a "project" with tasks, deadlines, constraints, etc. has dramatically improved both students' and supervisors' daily work.

As a further implementation, we have improved the previous version of our services, adding for each task or project the possibility of adding constraints and deadlines. A task has several other features, like a status (indicates at what stage of development is the project), priority, temporal constraints, and the creation of milestones (used to indicate the achievement of the

objectives set at the design stage).

From a security and protection perspective, the community where the project is created/managed guarantees a "sandbox" for the permissions' management. A user holding the appropriate permissions is allowed to create a project with an arbitrary number of sub-tasks to which different resources can be assigned (Fig.5). The roles that we have decided to support are:

- Owner of the project: role assigned when creating the project. The owner is the user with total control over the project and has no limit in respect to assignment of roles, cancellation of tasks, attachment, etc.
- Manager: this role will have the same potential as the project owner, with some restrictions on the tasks created by other managers. A manager may appoint other managers or simple resources, and this appointment can be made only on tasks whose owner is the creator of the task itself. The same goes for the cancellation and modification of activities. Note how the role of the project owner is separated from the others because of the control on every part of the task list regardless of the assignments. The task manager and the owner will also be asked to indicate the status of a task or project, thus introducing control over the work of other users.
- Resources or executors of the project: these users will have a limited subset of actions since their primary purpose is to perform the task and inform the manager through a report.
- Guest: this role is for those users who want to enable you to view a project without, however, afford to interact with it in any way.

You can create three different types of projects or task lists on the platform depending on confidentiality and the context required. You can create personal and public projects within a community or personal projects at the portal level, i.e., outside any specific community. A task list is visible only to the creator of the same and to the assigned people. However, a public project provides the necessary permissions for users with admin rights within the community to view and interact with all users involved, inviting external people from other communities or even not enrolled in the platform. We get more flexibility in a portal task list, conceptually associated with a super-community, where all subscribers to the portal (here we are at the highest level of the communities' hierarchy) are considered within the same context. We, therefore, can potentially engage all people registered with the platform in a single project, regardless of the inclusion in any community.

Another example of beneficial integration of Project Management services into a collaboration platform is attachments' management. Every task of a project can have some interesting documents attached with it to explain/clarify/deepen what the task has to do. In educational projects, this feature can be even more important if not indispensable, for example, if we consider a task as educational homework and the attachment as the instructions and content of the task to be performed. So in educational usage of project management tasks, attachments should be almost considered as a requirement rather than simply a desired feature. The interesting thing is that this feature, considering the concept of community that protects the visibility of documents and limits access to the members with appropriate authorizations, is substantially already available. In an educational project, we can add documents to the file repository of the community, then create the tasks list, and finally attach the appropriate documents to the relative tasks. The cost of implementing this feature has been minimal compared to the benefits brought to the Project Management service. (Fig.6)

Budgeting is another relevant feature, even if in educational contexts it is not so common. The implementation has been oriented to the maximum easiness because we knew that a certain level of complexity and project management knowledge is needed when dealing with the budget. The feature allows classifying resources in work or material resources. According to how the relative fees will be charged on the project, it assigns them an hourly or quantity cost. Users can then allocate resources to tasks, and the engine will calculate the budget for the different tasks using a sort of OLAP cube (Fig.7). Availability for every resource, of the day-by-day assignment plan on each task of the project is a direct consequence of the assignments. This availability, in turn, provides allocation charts, resources' overloading information, and other double-checking tools commonly available in full-fledged software tools.

Finally, to summarize the most important features of this new service, we have to mention the reporting facilities provided to selected users. This service "simply" presents all data deriving from the previous planning activities, providing some primary reporting mechanism. In the current version of the platform, this feature is still in its preliminary steps, not because of lack of data, but mainly because of three different reasons: a) need to implement core features before this; b) test the feature extensively to ask users which kind of reports, among the many possible, they would like to have, also considering the possible crossing with data not necessarily related to project management but rather, with educational aspects of the community; c) the possibility of exporting data and creating reporting with external BI/reporting tools, rather than implement them internally.

Figure 8 shows an example of reporting obtained from a project planned inside a community. We can see details about the WBS with its tasks, completion level, and associated costs to any single task.

5. DISCUSSION AND RESULTS

To test the above-presented services, we asked the collaboration of six project managers with a tested

experience in the educational field, asking them to plan their project using our platform's contemporarily Microsoft Project and services. The projects were of different sizes, different complexity, and a different number of involved resources, so they were not perfectly comparable. Our interest was to test the functionalities of the services created inside "Online Communities", and especially have feedback from those users, particularly experts in Microsoft Project. This constructive input was helpful to understand if our services could be a valuable contribution to what they were usually doing with their favorite software.

It is not easy to compete with an established software since the 80s like Microsoft Project, with millions of users and a development team that can extend the platform towards the desired features. Our aim was mainly to understand the appreciation for what we included, especially the advantages of being integrated with a web platform that provides different services concerning pure project management services.

We asked our testers to implement their educational projects using both the services presented in this paper and Microsoft Project. The PMs involved in the experimentation have a medium to a high level of experience in using MS Project, so their judgment could have been very severe concerning our platform. The educational projects were different but substantially falling into these macro-categories: bachelor/master theses, long-term assignments, planning new Master's and bachelor degrees, educational materials, software development. Each Project Manager plans over three months from 3 to 5 projects in both versions, replicating the same tasks, duration, predecessors, and the rest of the requirements of the critical path method. We then collected their quantitative feedback about some analytical dimensions. The dimensions of analysis have the following characteristics and features offered by the two platforms:

- Project settings Definition
- Calendar exceptions management
- Work breakdown structure creation
- Predecessors, Leads, and lags
- Milestones
- Constraints
- Assigning resources to tasks
- Attachments
- Reporting functions

For each of these investigation areas and for each of the above-listed features of the platforms, subjects have expressed an appreciation score, an overall judgment on the production of a project plan, and the user experience with them. Finally, we asked to give a qualitative suggestion for our platform in a final "desired area of improvements" answer. As a qualitative integration, a forum for general discussion has been created and made available to the community participants where the test took place. The final results also consider the valuable comments for the final analysis, so we excluded all the requests of clarification and bugs notification. We kept the judgments useful for our research. Here follows a summary of the questions and the quantitative results expressed on a Likert scale of 1 to 5, with 5 being an excellent evaluation. (Fig.9).

From the results emerges a strong appreciation of the MS Project features (as primarily expected), but some exceptions are notable and can be an interesting field of improvements and promotion for the use of our platform. The first notable exception regards the assignment of resources to tasks. This task is notoriously uncomfortable in any project management software. Still, from qualitative comments, we have noticed that the evident appreciation in favor of "Online Communities" derives from the availability of resources from the community members. This availability lets the project manager understand the project team, assign it to the various tasks, control them in the sense of "members of a community", not unknown resources listed in the software.

The second feature that received higher appreciation compared to MS Project has been attachment management. This result is another expected one, both because of the lack of web availability for the Microsoft solution and for the beneficial idea of having a file repository, organized in folders and protected with authentication/authorization mechanisms, from where to pick up the file that I want to attach to that specific task. All comments from our users converged on this helpful feature, paradoxically a feature recommended for MS Project. Finally, the overall judgments on the user experience and the applicability of the project management tool in educational contexts. Here the comments have been richer of comments and appreciation, even if we must consider that MS Project is a desktop application, full of functionalities that most users don't know. That creates a sometimes clumsy and poorly efficient user experience. More important for us is the positive results compared to the usability and usefulness of project management tools, which confirms our initial research question about the needs of these services for whoever is involved in educational processes.

Other comments on the forum that could be relevant for the research follow:

- web-based features of "Online Communities" have been very appreciated compared to MS Project: a web-based software tool, even if not very equipped with lots of functionalities, is much more helpful for today's project managers than a desktop, Windows-only tools, especially in an educational context where limited budget and low expertise on project management create an entry-barrier to the usage of these services;
- a particularly appreciated feature is the community-based management of resources, which certainly need many improvements (like some cost features, profile management internal to the platform, skills portfolio management, etc.), but even in the current

version could be very useful. Consider again that other services of the platform, apparently disconnected from Project Management services and not created for that purpose, have revealed their utility when the project management services have been added to the platform. A typical example is the "Curriculum" service, implemented for teachers to know the CV of the community members. In a Project Management context, this could be used by the project manager to understand skills and competencies of the project's resources;

- The forum has been helpful for users to communicate some minor bugs related to screen refresh and CPM incoherences
- All the users have reported the lack of some functionalities that available in MS Project, mainly advanced features not related to the CPM implementation (custom fields, total slack, recurring tasks, timesheet management): excellent suggestions for future improvements;
- Few user-experience suggestions were related to the assignment of resources and management of the CPM. All the users, being project managers with long experience, were aware of the many improvements on one side and the complexity of the needs in Project Management. Not everything in Project Management is simple or can be simplified.
- A few comments were about several features of "Online Communities" that have been considered even better than stand-alone MS Project. These comments are not surprising, and it's not a significant element of proudness because it is straightforward to propose something better when somebody else already opened the path. What we consider instead a valid argument is the availability of non-predictive project management features inside e-learning contexts.

6. CONCLUSIONS

In this paper, we presented a new set of features of our LMS called "Online Communities" that extend the platform's services towards the use of Project Management tools and techniques inside educational and collaborative contexts. These tools and techniques, made available for any user inside the platform, allowed to implement the idea of managing a project inside a collaborative and educational environment. This approach revealed two positive aspects: a) the appropriateness of Project Management concepts inside educational contexts, because many of the activities we perform during educational tasks can be seen as part of a project b) the advantage of implementing these services inside a virtual communities' environments, that provides a natural and fertile ground for the development of these services, the integration with other already-available services, and their availability to community members. A qualitative investigation has been performed using professional Project Managers and comparing their user experience in planning (educational) projects with a top-ranked project management software and with the services provided by "Online Communities". Results are surprisingly encouraging, paradoxically because "Online Communities" has not implemented many services mostly considered "not very useful" or "rarely used" by project managers.

On the contrary, complete and up-to-date services of Project Management integrated into a collaborative platform seem to help users in daily operations, with a higher level of usefulness compared to trivial to-do lists or simple task managers. Results can not be interpreted as extensive research, but due to the experience and professional quality of the involved project managers both from the project and educational side, we consider these results very interesting. They validate our original hypothesis substantially, even if not numerically. Educational world and LMSs in particular, need to incapsulate project management services into the offered services. The next evolutionary steps are the completion of some project management services with specific features (overallocation management, budgeting details, cost-type resources, charting, and reporting) and greater integration with the other services provided by "Online Communities", like calendar management, resource booking, grade assignments to assigned tasks, etc.

REFERENCES

[1] Levitt R. (2011) Towards Project Management 2.0, Engineering Project Organization Journal, 1:3, 197-210

[2] Molinari A. (2015), Collaboration Services As A New Perspective For E-Learning Systems, Multidisciplinary Academic Conference on Education, Teaching and Learning (MAC-ETL 2015), Prague, 6-8 2015

[3] Project Management Institute. A Guide To The Project Management Body Of Knowledge (PMBOK Guides), 5th Edition. Project Management Institute, 2013

[4] Atkinson R (1999) Project Management: cost, time and quality, two best guesses and a phenomenon, its time to accept other success criteriam International Journal of Project Management Vol. 17, No. 6, pp. 337-342, 1999 # 1999 Elsevier Science Ltd and IPMA.

[5] Meredith, J. R., & Mantel Jr, S. J. (2011). *Project Management: a managerial approach*. John Wiley & Sons.

[6] The Standish Group (2013), Chaos Manifesto 2013: Think Big, Act Small, retrieved from http://www.versionone.com/assets/img/files/CHAOSManifesto2013.pdf

[7] Mohamed, B., & Koehler, T. (2012). The Effect of Project-Based Web 2.0-Learning on Students'

Outcomes. In *Towards Learning and Instruction in Web 3.0* (pp. 51-70). Springer New York.

[8] Nahla El Zant El Kadhi, Hanaa Al-Sharrah (2011) New Methodology for Developing Digital Curricula, ICCGI 2011: The Sixth International Multi-Conference on Computing in the Global Information Technology, pp 148-153, ISBN: 978-1-61208-139-7

[9] Jackson, D. W. (2010). Tool, Collaboration vs Communication: Selecting the Appropriate. Law Library Journal, 102 (2), 315-324.

[10] Colazzo, L.; Molinari, A.; Villa, N. (2009). Collaboration vs. Participation: The Role of Virtual Communities in a Web 2.0 World, Education Technology and Computer, 2009. ICETC '09. IEEE International Conference, 321-325, 17-20 April 2009 Singapore

FIGURES

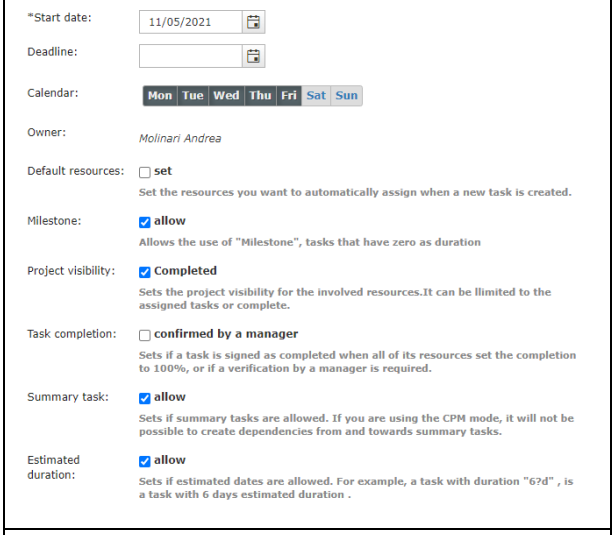


Fig.1 – Options for advanced settings of an educational project

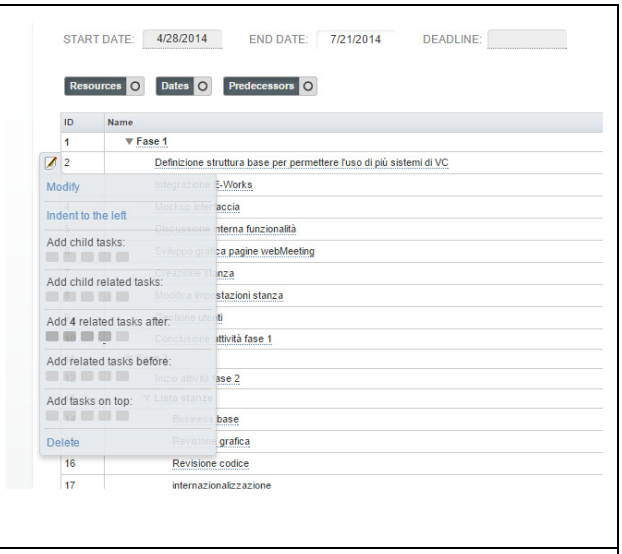


Fig.2 Structuring a WBS with child tasks and predecessors

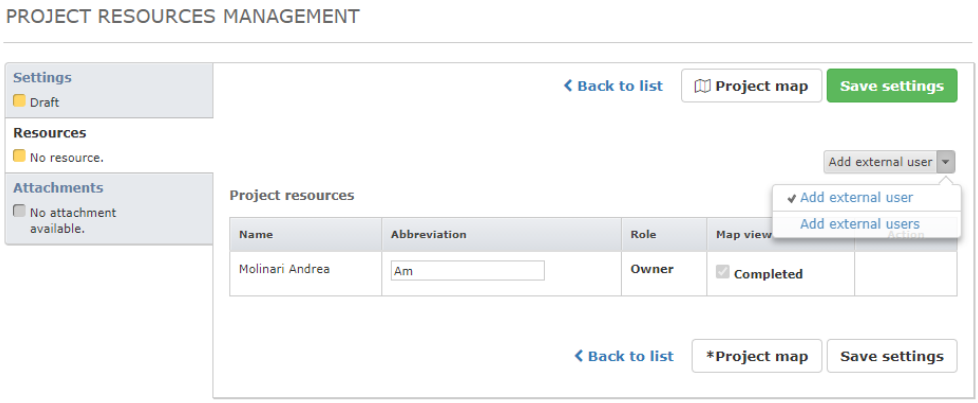


Fig.3 Managing resources and roles inside the project

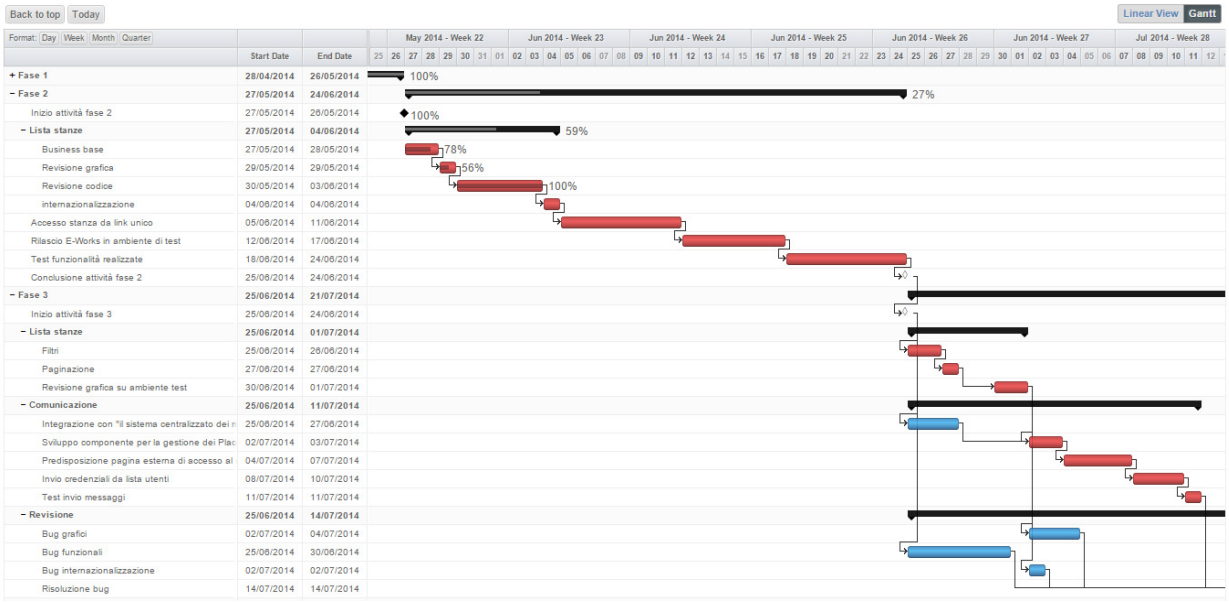


Fig.4: the WBS with critical path, predecessors, and completion percentage

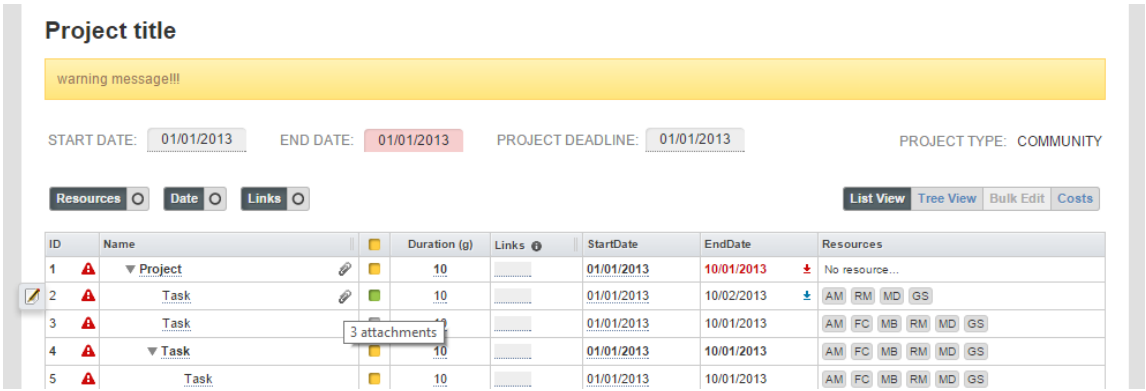


Fig.5: WBS with assignments of resources taken from the community's users

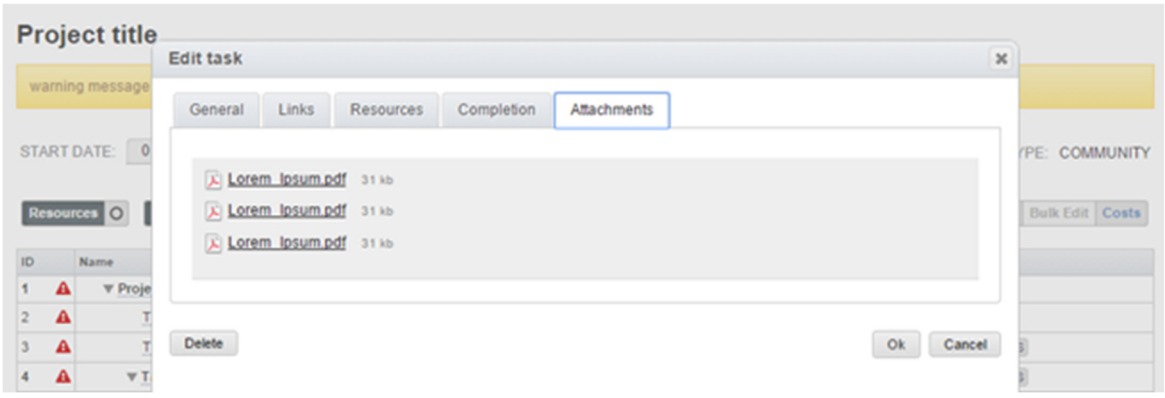


Fig.6: Property of a task with the possibility of attaching files taken from the community's repository

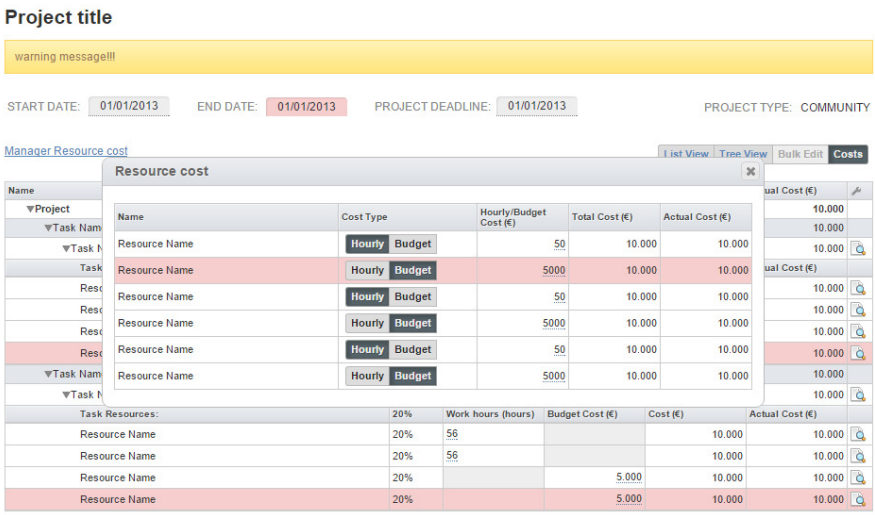


Fig.7 Resource Management with costs and allocation availability for each resource.

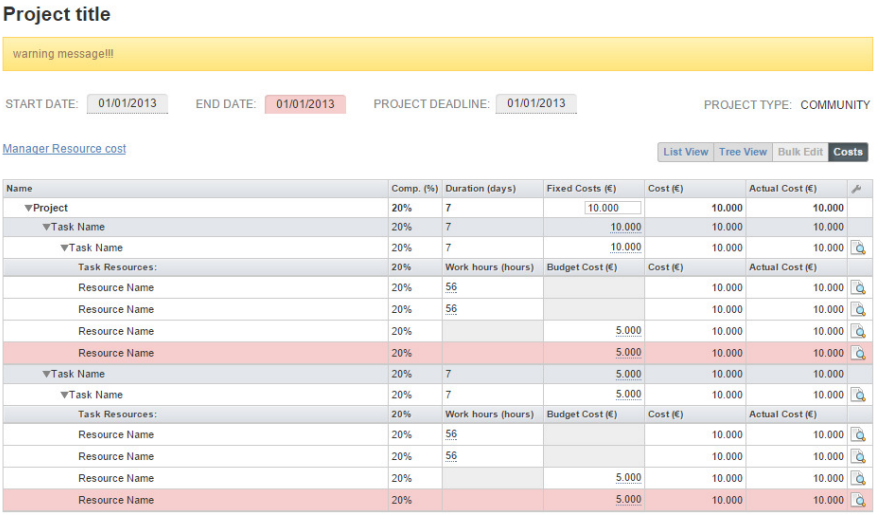


Fig.8 WBS, tasks, completion, and associated costs to any single tasks

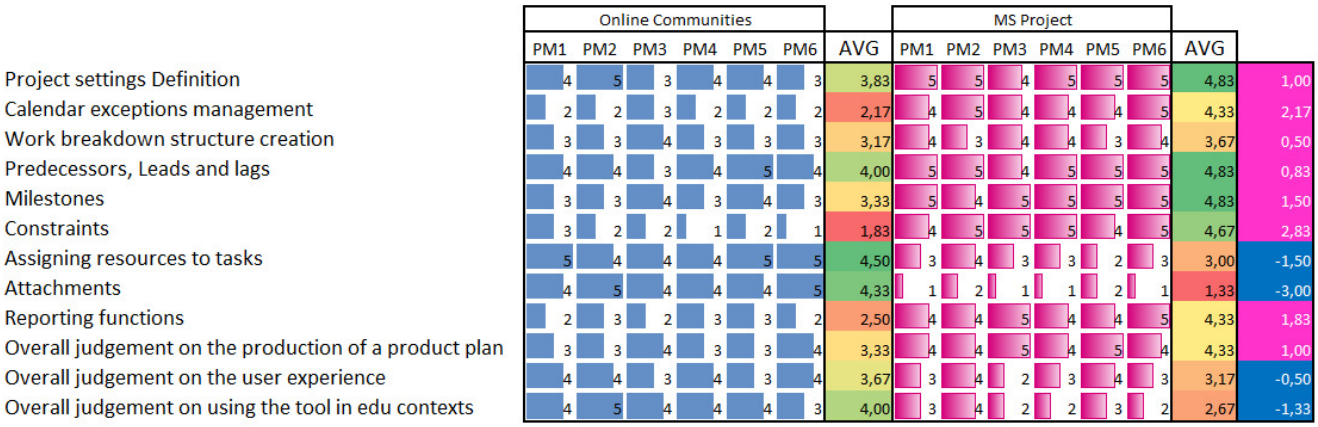


Fig.9 The results of the experiment of the new PM services on 6 users

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Supporting Emotion Automatic Detection and Analysis over Real-Life Text Corpora via Deep Learning: Model, Methodology, and Framework*

Alfredo Cuzzocrea[a], Giosué Lo Bosco[b,c], Mariano Maiorana[d], Giovanni Pilato[e] and Daniele Schicchi[f,c,*]

[a] iDEA Lab, University of Calabria, Rende, Italy & LORIA, Nancy, France
[b] Dipartimento di Matematica e Informatica, Universitá degli Studi di Palermo, Via Archirafi 34, 90123 Palermo, Italy
[c] Dipartimento SIT, Istituto Euro-Mediterraneo di Scienza e Tecnologia, Via Michele Miraglia 20, 90139 Palermo
[d] Cluster Reply SRL, Via Robert Kock 1/4, 20152 Milano, Italy
[e] CNR, Istituto di Calcolo e Reti ad Alte Prestazioni, Consiglio Nazionale delle Ricerche, Via Ugo La Malfa 153, 90146 Palermo, Italy
[f] CNR, Istituto di Tecnologie Didattiche , Via Ugo La Malfa 153, 90146, Palermo, Italy

ARTICLE INFO

Article History:
Submitted
Revised 6.1.2021
Second Revision 8.1.2021
Accepted 9.7.2021

Keywords:
Satire Detection, Deep Learning, NLP

ABSTRACT

This paper describes an approach for supporting automatic satire detection through effective deep learning (DL) architecture that has been shown to be useful for addressing sarcasm/irony detection problems. We both trained and tested the system exploiting articles derived from two important satiric blogs, *Lercio* and *IlFattoQuotidiano*, and significant Italian newspapers.

1. Introduction

Satire is a way of criticizing people (or ideas) by ridiculing them on political, social, and morals topics (e.g., [33]. Most of the time, such a language form is utilized to influence people's opinions. It is a figurative form of language that leverages comedic devices such as *parody* (i.e to imitate techniques and style of some person, place or thing), *exaggeration* (i.e to represent something beyond normality make it ridiculous), *incongruity* (i.e to present things that are absurd concerning the context), *reversal* (i.e to present the opposite of normal order), *irony/sarcasm* (i.e to say something that is the opposite of what a person mean). Moreover, satire masks emotions like irritation and disappointment by using ironic content.

The easy way of denouncing political and societal problems exploiting humor has brought consensus to satire that has been widely accepted. It leads people to constructive social criticism, to participate actively in the socio-political life, representing a sign of democracy. Unfortunately, the ironic nature of satire tends to mislead subjects that can believe the humorous news as they were real; therefore, satirical news can be deceptive and harmful.

Detecting satire is one of the most challenging computational linguistics tasks, natural language processing, and social multimedia sentiment analysis. It differs from irony detection since satire *mocks* something or someone, while irony is intended to be a way for causing laughter. Tackling such a task means both to pinpoint linguistic entities that characterize satire and look at how they are used to express a more complex meaning.

As satirical texts include figurative communication for expressing ideas/opinions concerning people, sentiment anal-

* This research has been made in the context of the Excellence Chair in Computer Engineering – Big Data Management and Analytics at LORIA, Nancy, France

*Corresponding author
alfredo.cuzzocrea@unical.it (A. Cuzzocrea); giosue.lobosco@unipa.it (G. Lo Bosco); m.maiorana@reply.it (M. Maiorana); giovanni.pilato@icar.cnr.it (G. Pilato); daniele.schicchi@itd.cnr.it (D. Schicchi)
ORCID(s): 0000-0002-7104-6415 (A. Cuzzocrea); 0000-0002-1602-0693 (G. Lo Bosco); 0000-0002-6254-2249 (G. Pilato); 0000-0003-0154-2736 (D. Schicchi)

DOI reference number: 10.18293/JVLC2021-N2-016

ysis systems may be negatively affected. In this case, satire should be adequately addressed to avoid performances degradation of such systems, mainly if sarcasm/irony is used [1]. Moreover, reliably detecting satire can benefit many other research areas where figurative language usage can be a problem, such as *Affective Computing* [29]. An autonomous way of detecting satire might help computers interpret human interaction and notice its emotional state, improving the human-computer experience. On the basis of a larger vision, these topics are also interesting due to their integration with modern big data topics (e.g., [6, 10, 28, 9]), especially due to the algorithmic side represented by machine and deep learning tools. In this paper, we tackle automatic satire detection through effective deep learning (DL) architecture that has been shown to be effective for addressing the sarcasm/irony detection problem [19]. The Neural Network (NN) exploits articles derived from two important satiric blogs, *Lercio* and *IlFattoQuotidiano*, and major Italian newspapers. The dataset has been specifically created for the task, and it includes news concerning similar topics. Experiments show an optimal performance achieved by the network that is capable of performing well on satire recognition. The network demonstrates the ability to detect satire in a context where it is not marked as in IlFattoQuotidiano. In fact, in this special case, news are so realistic that they seem to be true [29]. An autonomous way of detecting satire might help computers interpret human interaction and notice its emotional state, improving the human-computer experience. On the other hand, studying these techniques as combined with the emerging *big data trend* (e.g., [17, 16, 12, 14, 15]) is an interesting challenge. A preliminary version of this paper appears in [13].

2. The Overall Proposed Methodology

Recognizing *satire* can be modeled as a classification task subdividing *satiric* and *non-satiric* articles in two different classes. Such a task has been widely tackled by using machine learning algorithms, and it has been shown that it is important to consider various aspects related to the application domain. For what concerns the subject problem, many factors should be taken into account: the way the text is represented and how it is structured (sec. 2.1), the model's architecture for tackling the task and its tuning (sec 2.2 and 2.3). Le Hoang Son et al. [22] have introduced a deep learning model that promises optimal performances for detecting sarcasm/irony. We believe that such a network can also help recognizing the main aspects of the satire; a detailed description is given in sec. 2.2.

2.1. Preprocessing

The preprocessing phase deals with the input arrangement to make it analyzable to the model as best as possible. Most of the time, the text is changed by removing punctuation marks, stop-words, etc. In this case, since the articles have been harvested from online resources we focused on the removal of the *author's name*, *HTML tags*, *hyperlinks*, and *hashtags*. Subsequently, the input text is split into tokens (i.e., words and punctuation marks) using NLTK [1]. To level out the lengths of the articles, we have analyzed the cumulative frequency of the length of the texts, and then we have selected a value $L = 4500$ words such that we considered 95% of the entire set of articles. Finally, each token is mapped to a 300-dimensional space by a pre-trained embedding tool that relies on FastText [5, 21]. Therefore, each article is represented by a matrix of real values of size $(L, 300)$. We crop texts longer than L, and we pad with 0s texts that are shorter.

2.2. Architecture

The network's architecture is inspired from the one presented by Le Hoang Son et al [22], that exploits *Bidirectional Long Short Term Memory* (BiLSTM), *Soft Attention Mechanism*, *Convolutional* NNs, and *Fully Connected* NNs. Moreover, such a model consider five different auxiliary characteristics that have been shown to be relevant to sarcasm/irony detection: number of exclamation marks (!), number of question marks (?), number of periods (.), number of capital letters, number of uses of *or*. A complete model representation is given in figure 1.

2.2.1. Input Layer

The first network's layer is the *Input* layer which manage the pre-processed text in order to allow the analysis by the BiLSTM.

2.2.2. BiLSTM Layer

BiLSTM is composed of two LSTM layers which examine respectively the input sequence in *forward* (from the first token x_0 to the last one x_T) and *backward* (from the last token x_T to the first one x_0) ways. LSTM *cell*, is a neural unit created specifically for overcoming the vanish/exploding gradient problem [4] that affects the training phase by using the backpropagation through time algorithm. The *cell* is composed of a set of *gates* (i.e input, forget, and output gate) which control the flow of information. The *forget* gate deals with choosing the information part should be kept and what should be gotten rid, the *input* gate proposes new information that is worth to be considered, and the *output* gate mix the contributes given by both the *input* and *forget* gates for creating the final cell's output. LSTM cell leverages two *feedback* loops (i.e internal and external) which allow to track the sequence of elements the cell has already analyzed through a sequence of internal states $h_1, \ldots, h_T$. The final output of the LSTM cell is its final internal state that is strictly dependent of the previous ones. The formulation of a LSTM unit, named *memory unit*, is described in by the following equations [23]:

$$
\begin{aligned}
f_t &= \sigma(W_f x_t + U_f h_{t-1} + b_f) \\
i_t &= \sigma(W_i x_t + U_i h_{t-1} + b_i) \\
o_t &= \sigma(W_o x_t + U_o h_{t-1} + b_o) \\
c_t &= tanh(W_c x_t + U_c h_{t-1} + b_c) \\
s_t &= f_t \odot s_{t-1} + i_t \odot c_t \\
h_t &= tanh(s_t) \odot o_t
\end{aligned}
$$

[1] www.nltk.org

where f_t, i_t, o_t are respectively the input, forget and output gates, the $\odot$ is the element-wise multiplication, the b_f, b_i, b_o, b_c are bias vectors, while tanh is the hyperbolic tangent and sigma is the sigmoid function.

The analysis of the input text in these two opposite directions create two representation of the input sequence: straight and reversed. BiLSTM layer merges the output of the two LSTM layers into a single output by concatenating them. The final vector, if examined through the soft attention, allow the network to capture the salient words considering the input text totally.

2.2.3. Soft Attention Layer

The Soft Attention is a mechanism that weight the input sequence elements on the basis of their relevance for the classification task, suggesting on what elements leverage for classifying the input correctly. It exploits the sequence of LSTM states during the examination of the input sequence.

The attention layer's output is the *context-vector*. It is computed as the weighted sum of the *attention weights* α_t and the LSTM's states $h_0, \ldots, h_T$. The approach is described by the following formulas, considering w_α the weights matrix:

$$
\begin{aligned}
z_t &= h_t w_\alpha \\
\alpha_t &= \frac{e^{z_t}}{\sum_{i=1}^{T} e^{z_i}} \\
c &= \sum_{i=1}^{T} \alpha_i h_i
\end{aligned}
$$

In this case, the context-vector c is extended by concatenating the auxiliary features. Finally, one-dimensional vector C which contains the analysis of the BiLSTM layer and the Pragmatic features becomes the input of the next convolutional layer.

2.2.4. Convolutional Layer

We stacked three convolutional layers for the feature learning. Each convolving filter of size s slides over the input vector to compute a localized feature vector v_j for each possible word through a nonlinear activation function. For each filter, a transition matrix T is generated. Such a matrix is iteratively applied to a part of the input vector to compute the features as following :

$$v_j = f(\langle T, F_{j:j+s-1} \rangle + b_a)$$

where $\langle \cdot, \cdot \rangle$ is the inner product, $F_{g,l}$ is the part of the input vector which includes elements from position g to position l, b_a is a bias related to the specific filter, and f is a non linear function.

The output of the convolutional layers is a vector of features $v = v_1, v_2, \ldots, v_{n-s+1}$ where n is the length of the input vector.

A max-pooling layer then processes the convolutional layer's output. Such a layer extracts the largest computed feature for each filter, considering only the most relevant ones. The output layer then analyzes the output vector that included the selected features.

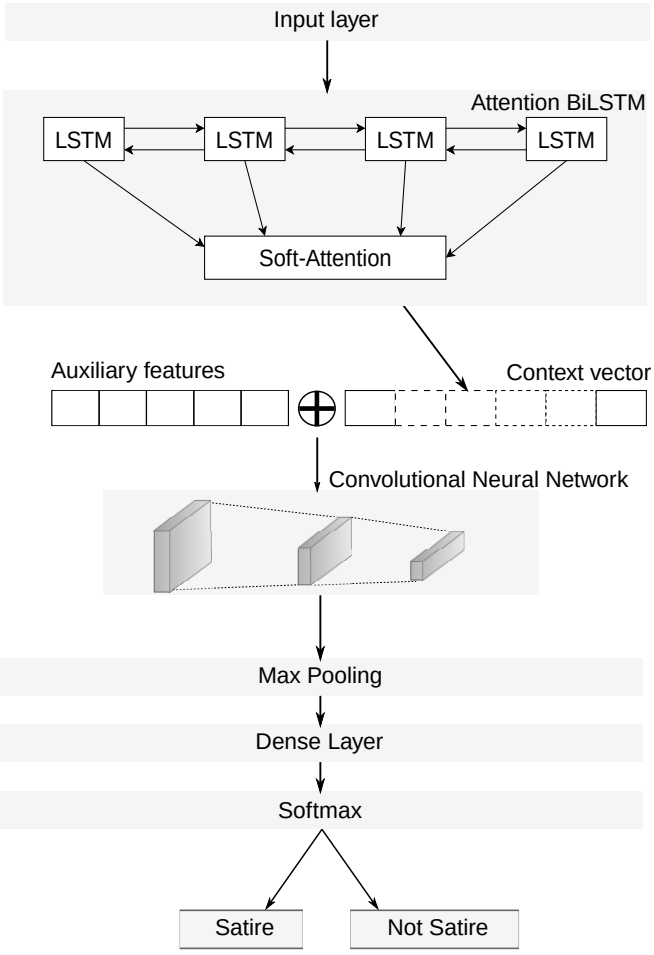


Figure 1: The representation of the Neural Network's architecture. The first layer manages the input in order to make it available for analysis. BiLSTM layer analyses the input in the forward and backward way to give a complete representation of the text. The attention mechanism is exploited for detecting the most relevant words for accomplishing the classification task. Its output is concatenated to the auxiliary features and then it is given as input to the convolutional layer. Such a layer extract prominent features, which are processed by a fully connected layer activated by softmax.

2.2.5. Output Layer

The output layer is a Fully Connected NN activated by Softmax. Such a layer takes as input the features extracted by the max-pooling layer. Employing the Softmax activation function computes the probability that the input text belongs to the either *satiric* or *non-satiric* class.

2.3. Parameters

Hyperparameters have been chosen empirically and taking inspiration from [22, 1]. Different tries have shown that taking a small learning rate and using a small minibatch coupled with Dropout regularization factors helps the network improve its performance by diminishing the loss. A complete list of them can be found in table 1.

Table 1
List of the model's hyperparameters.

Embedding size	300
LSTM neurons	500
Batch size	10
Convolutional layers	3
Kernel size	3
Convolutional activation function	ReLU
Dropout BiLSTM Dropout ConvNet	0.2 0.4
Optimization algorithm	Adam
Learning rate	0.0001
Dense layer neurons	350

3. Satire Detection: Emerging Challenges and Open Issues

The automatic satire detection task opens several new and challenging research perspectives. In the following, we highlight those that have been scored as relevant by our study.

Language can convey information that can be interpretable depending on the background of knowledge and the personality of both the reader and the writer. Even human readers sometimes have difficulties in recognizing satire [2].

3.1. Fake News Detection

One of the most interesting issues is the automatic detection of fake news. This problem is relevant since the fake news phenomenon is widespread and involves different aspects of life, politics, society, and the economy. In [24] it has been studied the performance of different machine learning techniques to three distinct datasets to find if the choice of specific datasets induces a bias in the task of fake news detection. Along with this study, BERT and the pre-trained models gave the best performances for fake news detection, especially when tiny datasets are involved. Another key and challenging problem is the design of automatic recognition systems capable of differentiating between deliberate deception and irony or satire. The former ones are actually fake news, but the latter ones, even if are close to fake news, are characterized by the entirely different intentions of the author. As highlighted by [11], the difference between the two is very subtle. It is sometimes difficult even for people to distinguish between them, particularly those who do not have a particular sense of humor.

One aspect of tackling, strictly related to the issues mentioned above, is to determine a more fine-grained classification: e.g., distinguishing between deceiving someone (hoax), criticizing someone or specific ideas by using humor and irony (satire), or deliberately publishing false statements of facts (fake news).

3.2. Text Feature Selection

An ensembled text feature selection method has been introduced in [30], which includes unigrams, semantic, psycholinguistic, and statistical features, and data mining techniques. A set of binary classifiers exploit these features to determine satiric news and ironic customer reviews.

Among the latest techniques arising, there is the detection of non-common tokens, specific text elements, repeated words, question and exclamation marks, emoticons, etc. [11]. These are usually neglected during the traditional preprocessing phase of the text.

Another area to investigate is the role played by Parts-of-speech (POS) Tags assigned to each word of the document, as well as the combination of knowledge acquired from tools like SlangNet [18], Colloquial WordNet [26], SentiWordNet [3], and SentiStrength [34]. The aim is to detect slang and colloquial expressions and understand the sentiment expressed in the textual document.

3.3. Specific Classification Issues and Methodologies

One challenge is to identify unsupervised or semi supervised methodologies for identifying satirical articles. A closely related aspect of this is also to make the classifier learn to recognize satire based on content, without being influenced by possible elements that may lead back to the source of the text (e.g., a satirical site will always propose satire articles rather than real news).

In the literature a semi-supervised methodology has been introduced by Tsur et al. [36] to detect sarcastic and non-sarcastic sentences. Their proposal used a list of sentence patterns based on high-and low-frequency words. Furthermore, they took into account some syntactic features like the length of a sentence, frequency of special and different case letters, etc.

For what is concerning the issue to investigate if the classifier learns to recognize the source of information instead of the satirical content, an adversarial training can be helpful to improve the robustness of the proposed models. An approach to tackle this problem has been illustrated in [27] by using a model for satire detection with an adversarial component to control the features that may be related to the publication source taken into account.

3.4. Rule-Based Approaches

A challenging aspect is to find approaches that overcome the limits of simply exploiting the textual features of a text trying to tackle the satire detection problem as a classical text classification one. In this context, it is desirable a comparison between the traditional classification approaches and those ones based on inferences, like, for example, that one proposed by Goldwasser et al. in [20] where common-sense inferences are used. The authors represent the structure of the text by capturing the main entities, their activities, and their utterances. The result of this procedure is a Narrative Representation Graph (NRG), which is exploited to compute how likely are the events and interactions to arise in a real, or a satirical context.

3.5. Exploring the Role of Multimodality

Using only the text to determine whether an article is satire or not can be reductive or even misleading. For this reason, one of the challenges is to consider various aspects:

for example, satirical images, cartoons, and other elements that may contribute to the determination of a particular article as satirical.

A multimodal approach to the individuation of satire has been proposed by Li et al [25], based on a visio-linguistic model named ViLBERT. The goal is to use multi-modal data to overcome the traditional use of only textual content f to classify articles into satirical and factual news.

3.6. Emotions and Satire Detection

Satire involves many aspects of personality and communication. For this reason, it is relevant to investigate the relationship between the emotion expressed in a text and the detection of satire. An interesting approach has been reported in [35], where a corpus of satirical and non-satirical news articles has been analyzed to detect satire by exploiting sentiment analysis and a social cognition engine.

4. Conclusions and Future Work

Satire aims at criticizing either something or someone leveraging on comedic devices. Its automatically detection is a non-trivial task that have to consider the components it is composed such as *parody*, *exaggeration*, *reversal*, *irony/sarcasm* which often are related to stand-alone research topics.

In this paper, we have introduced a powerful DL model that tackles the satire detection problem by examining lexical, syntactical, and auxiliary features. To support the analysis by the system, we exploited an effective pre-trained embedding tool based on FastText.

Future work will further analyze the network's behavior by exploiting incremental data [8] and clustering [7]. Moreover, we are going to study how satire might affect the text comprehension [31] and if it might be reproduced through automatic creative processes [32].

Acknowledgements

This research has been partially supported by the French PIA project "Lorraine Université d'Excellence", reference ANR-15-IDEX-04-LUE.

References

[1] Alcamo, T., Cuzzocrea, A., Lo Bosco, G., Pilato, G., Schicchi, D., 2020. Analysis and comparison of deep learning networks for supporting sentiment mining in text corpora, in: 22th International Conference on Information Integration and Web-based Applications and Services (iiWAS2020).

[2] Allcott, H., Gentzkow, M., 2017. Social media and fake news in the 2016 election. Journal of economic perspectives 31, 211–36.

[3] Baccianella, S., Esuli, A., Sebastiani, F., 2010. Sentiwordnet 3.0: an enhanced lexical resource for sentiment analysis and opinion mining., in: Lrec, pp. 2200–2204.

[4] Bengio, Y., Simard, P., Frasconi, P., 1994. Learning long-term dependencies with gradient descent is difficult. IEEE Transactions on Neural Networks 5, 157–166. doi:10.1109/72.279181.

[5] Bojanowski, P., Grave, E., Joulin, A., Mikolov, T., 2016. Enriching word vectors with subword information. CoRR abs/1607.04606.

[6] Campan, A., Cuzzocrea, A., Truta, T.M., 2017. Fighting fake news spread in online social networks: Actual trends and future research directions, in: 2017 IEEE International Conference on Big Data (Big Data), IEEE. pp. 4453–4457.

[7] Casalino, G., Castellano, G., Mencar, C., 2018. Incremental adaptive semi-supervised fuzzy clustering for data stream classification, in: 2018 IEEE Conference on Evolving and Adaptive Intelligent Systems (EAIS), pp. 1–7. doi:10.1109/EAIS.2018.8397172.

[8] Casalino, G., Castiello, C., Del Buono, N., Mencar, C., 2018. A framework for intelligent twitter data analysis with non-negative matrix factorization. International Journal of Web Information Systems .

[9] Castillo-Zúñiga, I., Rosas, F.J.L., Rodríguez-Martínez, L.C., Arteaga, J.M., López-Veyna, J.I., Rodríguez-Díaz, M.A., 2020. Internet data analysis methodology for cyberterrorism vocabulary detection, combining techniques of big data analytics, NLP and semantic web. Int. J. Semantic Web Inf. Syst. 16, 69–86.

[10] Ceci, M., Cuzzocrea, A., Malerba, D., 2015. Effectively and efficiently supporting roll-up and drill-down olap operations over continuous dimensions via hierarchical clustering. Journal of Intelligent Information Systems 44, 309–333.

[11] Chorać, M., Demestichas, K., Giełczyk, A., Herrero, Á., Ksieniewicz, P., Remoundou, K., Urda, D., Woźniak, M., 2020. Advanced machine learning techniques for fake news (online disinformation) detection: A systematic mapping study. Applied Soft Computing , 107050.

[12] Cuzzocrea, A., 2006. Improving range-sum query evaluation on data cubes via polynomial approximation. Data & Knowledge Engineering 56, 85–121.

[13] Cuzzocrea, A., Lo Bosco, G., Maiorana, M., Pilato, G., Schicchi, D., 2021. A novel approach for supporting italian satire detection through deep learning, in: Flexible Query Answering Systems - 134th International Conference, FQAS 2021, Bratislava, Slovakia, September 19–24, 2021, Proceedings, Springer.

[14] Cuzzocrea, A., Matrangolo, U., 2004. Analytical synopses for approximate query answering in olap environments, in: International Conference on Database and Expert Systems Applications, Springer. pp. 359–370.

[15] Cuzzocrea, A., Moussa, R., Xu, G., 2013. Olap*: effectively and efficiently supporting parallel olap over big data, in: International Conference on Model and Data Engineering, Springer. pp. 38–49.

[16] Cuzzocrea, A., Saccà, D., Serafino, P., 2006. A hierarchy-driven compression technique for advanced olap visualization of multidimensional data cubes, in: International Conference on Data Warehousing and Knowledge Discovery, Springer. pp. 106–119.

[17] Cuzzocrea, A., Serafino, P., 2009. Lcs-hist: taming massive high-dimensional data cube compression, in: Proceedings of the 12th International Conference on Extending Database Technology: Advances in Database Technology, pp. 768–779.

[18] Dhuliawala, S., Kanojia, D., Bhattacharyya, P., 2016. Slangnet: A wordnet like resource for english slang, in: Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pp. 4329–4332.

[19] Di Gangi, M.A., Lo Bosco, G., Pilato, G., 2019. Effectiveness of data-driven induction of semantic spaces and traditional classifiers for sarcasm detection. Natural Language Engineering 25, 257–285. doi:10.1017/S1351324919000019.

[20] Goldwasser, D., Zhang, X., 2016. Understanding satirical articles using common-sense. Transactions of the Association for Computational Linguistics 4, 537–549.

[21] Grave, E., Bojanowski, P., Gupta, P., Joulin, A., Mikolov, T., 2018. Learning word vectors for 157 languages, in: Proceedings of the International Conference on Language Resources and Evaluation (LREC 2018).

[22] Hoang Son, L., Kumar, A., Raj Saurabh, S., Arora, A., Nayyar, A., Abdel-Basset, M., 2019. Sarcasm detection using soft attention-based bidirectional long short-term memory model with convolution network. IEEE Access 7, 23319–23328.

[23] Hochreiter, S., Schmidhuber, J., 1997. Long short-term memory. Neural computation 9, 1735–1780.

[24] Khan, J.Y., Khondaker, M.T.I., Afroz, S., Uddin, G., Iqbal, A., 2021.

A benchmark study of machine learning models for online fake news detection. Machine Learning with Applications 4, 100032.

[25] Li, L., Levi, O., Hosseini, P., Broniatowski, D.A., 2020. A multi-modal method for satire detection using textual and visual cues. arXiv preprint arXiv:2010.06671 .

[26] McCrae, J.P., Wood, I., Hicks, A., 2017. The colloquial wordnet: Extending princeton wordnet with neologisms, in: International Conference on Language, Data and Knowledge, Springer. pp. 194–202.

[27] McHardy, R., Adel, H., Klinger, R., 2019. Adversarial training for satire detection: Controlling for confounding variables. arXiv preprint arXiv:1902.11145 .

[28] Novo-Lourés, M., Pavón, R., Laza, R., Ruano-Ordás, D., Méndez, J.R., 2020. Using natural language preprocessing architecture (NLPA) for big data text sources. Sci. Program. 2020, 2390941:1–2390941:13.

[29] Picard, R.W., 2000. Affective computing. MIT press.

[30] Ravi, K., Ravi, V., 2017. A novel automatic satire and irony detection using ensembled feature selection and data mining. Knowledge-Based Systems 120, 15–33.

[31] Schicchi, D., Lo Bosco, G., Pilato, G., 2019. Machine learning models for measuring syntax complexity of english text, in: Biologically Inspired Cognitive Architectures Meeting, Springer. pp. 449–454.

[32] Schicchi, D., Pilato, G., 2017. Wordy: a semi-automatic methodology aimed at the creation of neologisms based on a semantic network and blending devices, in: Conference on Complex, Intelligent, and Software Intensive Systems, Springer. pp. 236–248.

[33] Sinha, A., Patekar, P., Mamidi, R., 2019. Unsupervised approach for monitoring satire on social media, in: Majumder, P., Mitra, M., Gangopadhyay, S., Mehta, P. (Eds.), FIRE '19: Forum for Information Retrieval Evaluation, Kolkata, India, December, 2019, ACM. pp. 36–41.

[34] Thelwall, M., 2017. The heart and soul of the web? sentiment strength detection in the social web with sentistrength, in: Cyberemotions. Springer, pp. 119–134.

[35] Thu, P.P., Aung, T.N., 2017. Effective analysis of emotion-based satire detection model on various machine learning algorithms, in: 2017 IEEE 6th global conference on consumer electronics (GCCE), IEEE. pp. 1–5.

[36] Tsur, O., Davidov, D., Rappoport, A., 2010. Icwsm—a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews, in: fourth international AAAI conference on weblogs and social media.

Journal of Visual Language and Computing

journal homepage: www.ksiresearch.org/jvlc

Graphical Animations of the NS(L)PK Authentication Protocols[⋆,⋆⋆]

Thet Wai Mon[a], Dang Duy Bui[a], Duong Dinh Tran[a], Canh Minh Do[a] and Kazuhiro Ogata[a,*]

[a]School of Information Science, Japan Advanced Institute of Science and Technology (JAIST), 1-1 Asahidai, Nomi, Ishikawa 923-1292, Japan

ARTICLE INFO

Article History:
Submitted 3.1.2021
Revised 6.1.2021
Second Revision 8.1.2021
Accepted 10.10.2021

Keywords:
graphical animation
NSPK authentication protocol
NSLPK authentication protocol
state machine
state picture design

ABSTRACT

NSLPK is visualized using SMGA so that human users can visually perceive non-trivial characteristics of the protocol by observing graphical animations. NSPK is a public-key authentication protocol invented by Needham and Schroeder and NSLPK is a revised version of NSPK by Lowe. These characteristics could be used as lemmas to formally verify that NSLPK enjoys desired properties. We first carefully make a state picture design for NSLPK to produce good graphical animations with SMGA and then find out non-trivial characteristics of the protocol by observing its graphical animations. Finally, we also confirm the guessed characteristics using model checking. The work demonstrates that SMGA can be applied to a wider class of systems/protocols, authentication protocols in particular. The visualization of NSLPK is different from ordinary message sequence diagrams that have been often used for security protocols. It is convenient that message sequence diagrams can be automatically generated in a graphically animated way for some cases such that we need to see the order in which way what messages are sent, faked and/or received. Thus, we have revised SMGA so that message sequence diagrams can be automatically generated in a graphically animated way.

1. Introduction

Authentication protocols have become important technical components in this advanced highly networked world. If authentication protocols have some flaws (security holes), users' credentials may be leaked to malicious third parties. It is then really important to make sure that authentication protocols are reliable and truly secure. Therefore, we need to use some technologies for this purpose. One possible technology is formal verification with theorem proving in which one challenging task is lemma conjecture. If human users carefully observe graphical animations of a state machine, they could recognize the characteristics from which they could conjecture useful lemmas. We aim to come up with a better way to conjecture lemmas in much fewer efforts and less time by observing animations produced by the State Machine Graphical Animation tool (SMGA) to complete formal proof.

SMGA [11] has been developed to visualize graphical animations of state machines that can be used to formalize security protocols. The main purpose of SMGA is to help human users be able to visually perceive non-trivial characteristics of state machines by observing their graphical animations because humans are good at visual perception [7]. SMGA takes a finite state sequence of a state machine formalizing a protocol and produces its graphical animation by regarding the state sequence as a movie film. Observing such a graphical animation allows us to guess the characteristics of the state machine. We confirm whether the state machine enjoys guessed characteristics because such characteristics may or may not be true properties of the state machine. One possible way to do so is model checking. However, it does not guarantee that the state machine enjoys the properties when the reachable state space is unbounded. If that is the case, we should use some other techniques, such as theorem proving to make sure that the system enjoys the guessed properties. Several case studies of some protocols have been

⋆This work was partially supported by JST SICORP Grant Number JPMJSC20C2, Japan and FY2020 grant-in-aid for new technology research activities at universities (SHIBUYA SCIENCE CULTURE AND SPORTS FOUNDATION).

⋆⋆The present paper is an extended and revised version of the paper [9] presented at DMSVIVA 2021.

*Corresponding author

thetwaimon@jaist.ac.jp (T.W. Mon); bddang@jaist.ac.jp (D.D. Bui); duongtd@jaist.ac.jp (D.D. Tran); canhdominh@jaist.ac.jp (C.M. Do); ogata@jaist.ac.jp (K. Ogata)

ORCID(s): 0000-0002-2700-1762 (D.D. Bui); 0000-0001-7092-2084 (D.D. Tran); 0000-0002-1601-4584 (C.M. Do); 0000-0002-4441-3259 (K. Ogata)

DOI reference number: 10.18293/JVLC2021-N2-005

conducted with SMGA, such as Qlock [1] and MCS [12, 3], shared-memory mutual exclusion protocols, Suzuki-Kasami protocol [2], a distributed mutual exclusion protocol, and ABP [11], a communication protocol. Authentication protocols have not been yet, and this work is the very first time of tackling authentication protocols with SMGA.

Authentication protocols are often visualized as message sequence diagrams called the Alice-Bob format, where Alice and Bob are principals. It is possible to grasp the messages exchanged by Alice and Bob and the order of the messages sent and received by principals and faked by Cathy, an intruder when such protocols are visualized as message sequence diagrams. We need to take into account the existence of intruders so as to formally verify that authentication protocols enjoy desired properties, such as the nonce secrecy property. Cathy plays an ordinary principal from the Alice and Bob point of view but does something against authentication protocols, such as faking messages based on the gleaned information. Thus, many messages may be faked by Cathy. If many messages appear, it may not be straightforward to comprehend message sequence diagrams. This is why we came up with a different way to visualize NSLPK than message sequence diagrams.

We aim at coming up with a brand-new way to visualize the behavior of authentication protocols. Since it is known that state picture designs affect how well human users can detect non-trivial characteristics of protocols [3], we carefully make a state picture design of the NSLPK protocol and based on it to produce good graphical animations. By observing the graphical animations, some non-trivial characteristics are guessed and checked with Maude [4]. In the paper, we mainly focus on how to design the state picture of the NSLPK protocol and how some characteristics could be found by observing graphical animations with detailed experiments.

However, it is convenient that message sequence diagrams can be automatically generated in a graphically animated way for some cases such that we need to see the order in which way what messages are sent, faked and/or received. Thus, we have implemented SMGA-SD that is a tool that automatically generates message sequence diagrams in an animated way from a finite sequence of states. We have integrated SMGA-SD with SMGA. We have visualized NSPK and NSLPK in SMGA-SD.

The remaining part of the paper is organized as follows. Sect. 2 gives some preliminaries such as state machine, Maude, and SMGA. Sect. 3 describes the NSLPK protocol and Sect. 4 describes its formal specification. Sect. 5 presents the state picture design of the NSLPK protocol in which the idea and the design are mainly conveyed. Sect. 6 reports on how we can find characteristics by observing graphical animations. Sect. 7 describes SMGA-SD and some experiments with SMGA-SD. Sect. 8 concludes the paper with some pieces of future work.

2. Preliminaries

This section describes some preliminaries needed to comprehend what follows in the present paper: state machines, Maude, SMGA, NSLPK protocol. State machines are mathematical models used to formalize systems. Maude is a rewriting specification/programming language in which state machines can be described. Maude also refers to its processor equipped with model checking functionality. NSLPK protocol is an authentication protocol and used as one main example in the present paper.

2.1. State machines

A state machine is a mathematical model of computation. Based on the current state and given input, state machine performs state transitions and produces outputs. A state machine $M \triangleq \langle S, I, T \rangle$ consists of a set S of states, a set $I \subseteq S$ of initial states, and a binary relation $T \subseteq S \times S$ over states. $(s, s') \in T$ is called a state transition where s' is successor state of s and may be written as $s \rightarrow_M s'$. The set $R \subseteq S$ of reachable states with respect to (wrt) M is inductively defined as follows: $I \subseteq R$ and if $s \in R$ and $s \rightarrow_M s'$, then $s' \in R$. A state predicate p is an invariant property wrt M if and only if $(\forall s \in R)\ p(s)$ that is $p(s)$ holds for all $s \in R$. A state predicate p can be interpreted as a set P of states such that $(\forall s \in P)\ p(s)$ and $(\forall s \notin P)\ \neg p(s)$. A finite sequence $s_0, \ldots, s_i, s_{i+1}, \ldots, s_n$ of states is called a finite computation of M if $s_0 \in I$ and $(s_i, s_{i+1}) \in T$ for each $i = 0, \ldots, n-1$.

Systems can be formalized as state machines. States are expressed as braced soups of observable components. Soups are associative-commutative collections, and observable components are name-value pairs. That is a state of S is expressed as associative-commutative collection of name-value pairs. The juxtaposition operator is used as the constructor of soups. Let $oc1, oc2, oc3$ be observable components, and $oc1\ oc2\ oc3$ is the soup of observable components. Then a state that consists of these three observable components can be expressed as $\{oc1\ oc2\ oc3\}$, which equals $\{oc3\ oc1\ oc2\}$ and some others due to associativity and commutativity. To specify state transitions we use Maude as a formal specification language.

Let us consider the hand game 'Rock Paper Scissors' between a human (you) and a machine and a system (called RPS) that is a series of matches of the games. Each state of the system is expressed as follows:

`{(pair: n(`X`,`Y`)) (result: `Z`)}`

where X is your current choice, Y is the computer's current choice and Z is the result of the match, where each of X and Y is one of `rock`, `paper` and `scissors`, and Z is one of `win`, `lose` and `draw`. Let us suppose that X, Y and Z are initially set to `rock`, `rock` and `draw`. Given a state `{(pair: n(`X`,`Y`)) (result: `Z`)}`, each of X and Y is randomly chosen from `rock`, `paper` and `scissors`. Once X and Y are fixed, we know the result Z from them. This can decide all state transitions of the state machine formalizing the system.

2.2. Maude

Maude [4] is a rewriting logic-based specification/programming language supporting both equational and rewriting logic. Maude makes it possible to describe soups, observable components and braced soups of observable components. When M is specifying in Maude, T is specified as rewrite rules. A rewrite rule starts with the keyword `rl`, followed by a label enclosed with square bracket and a colon, two patterns (terms that may contain variables) connected with =>, and ends with a full stop. A conditional rule starts with the keyword `crl` and has a condition with the keyword `if` before full stop. The following are forms of a rewrite rule and conditional rewrite rule:

`rl` $[lb] : l$ `=>` r .

where lb is a label. An instance of l is replaced with the corresponding instance of r.

`crl` $[lb] : l$ `=>` r `if` $\ldots$ `/\` c_i `/\` $\ldots$.

where lb is a label and c_i is part of the condition, which may be an equation $lc_i = rc_i$ or a matching equation $lc_i := rc_i$. The negation of $lc_i = rc_i$ can be written as $(lc_i =/= rc_i) =$ `true`, where `= true` can be omitted. If the condition $\ldots$ `/\` c_i `/\` $\ldots$ holds, an instance of l is replaced with the corresponding instance of r.

The state transitions of RPS is specified as the following rewrite rule:

```
clr [game-match] : {(pair: n(X,Y)) (result: Z)}
=> {(pair: n(X1,Y1)) (result: Z1)}
if X1 Xs1 := rock paper scissors /\
   Y1 Ys1 := rock paper scissors /\
   Z1 := result(X1,Y1) .
```

where `rock paper scissors` is the associative-commutative collection of `rock`, `paper` and `scissors` and `result` is the function that judges the game match based on `X1` and `Y1`. The first two matching equations in the condition randomly choose one of `rock`, `paper` and `scissors` and assign it to each of `X1` and `Y1`, and the third matching equation uses the function `result` with `X1` and `Y1` and assigns the result to `Z1`.

Maude is equipped with model checking facilities (a reachability analyzer and an LTL model checker). Maude provides the search command that allows finding a state reachable from s such that the state matches pattern p and satisfies condition c:

`search [n,m] in MOD :` s `=>*` p `such that` c .

where `MOD` is the name of the Maude module specifying the state machine under model checking, `n` and `m` are optional arguments stating a bound on the number of desired solutions and the maximum depth of the search, respectively. `n` typically is 1 and s is a given state (typically an initial state of the state machine). p is pattern and c is a condition. The condition part `such that` c can be omitted. The search command searches the reachable states from s for at most n states that can match the pattern p and make the condition c true. In this paper, Maude search command is used to confirm the characteristics guessed by observing graphical animations of NSLPK.

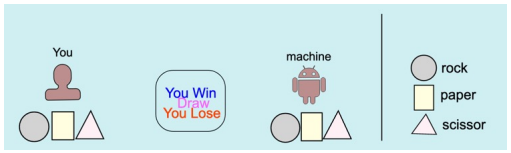


Figure 1: A state picture design for RPS

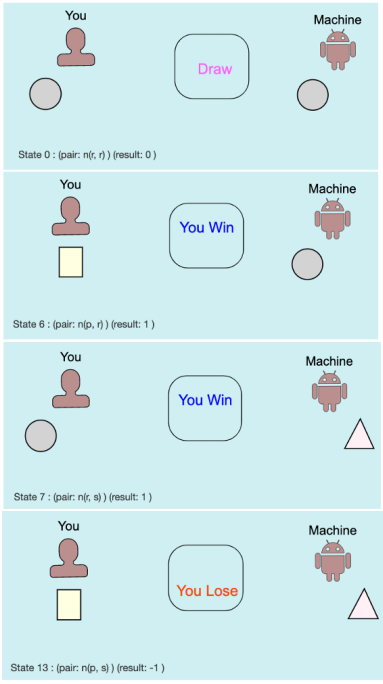


Figure 2: Four state pictures of a graphical animation for RPS

2.3. State machine graphical animation (SMGA)

State machine graphical animation tool (SMGA) is developed by Nguyen and Ogata [11]. The main purpose of SMGA is to help human users be able to recognize state patterns and perceive non-trivial characteristics of a state machine by observing its graphical animations. SMGA cannot automatically produce visual state picture designs and then it allows us to design a good state picture. As an input, SMGA basically takes a state picture design made by humans and a finite state sequence input generated by Maude. An output is a graphical animations by regarding the finite state sequence as a movie film based on the state picture design. One possible state picture design for the state machine formalizing RPS is shown in Figure 1. Figure 2 shows a four consecutive state pictures from the initial state.

3. NS(L)PK Protocols

NSPK [10] is a public-key authentication protocol designed by Needham and Schroeder and NSLPK [8] is a revised version of NSLPK by Low. NSLPK can be described as the following three message exchanges:

Init $\quad p \to q : \mathcal{E}_q(n_p, p)$

Resp $q \rightarrow p : \mathcal{E}_p(n_p, n_q, q)$
Ack $p \rightarrow q : \mathcal{E}_q(n_q)$

NSLPK uses public-key cryptography. Each principal, such as p and q, has a private/public key pair. The public key is known by all principals but the private one is only available to its owner. $\mathcal{E}_p(m)$ denotes the ciphertext obtained by encrypting a message m with the principal p's public key. n_p is a nonce (a random number) generated by principal p. A nonce is a unique and non-guessable number that is used only once in all protocol runs. The difference between NSLPK and NSPK is only the Resp message. NSPK uses $\mathcal{E}_p(n_p, n_q)$ as the Resp message.

If p wants to mutually authenticate q, p generates a nonce n_p and sends to q an Init message that consists of n_p and its ID p encrypted by the public key of q. When q receives the Init message, q tries to decrypt the ciphertext received by its private key. q then generates a nonce n_q and sends back to p a Resp message that consists of n_p, n_q, and ID q encrypted by the public key of p. On receipt of the Resp message, p tries to decrypt the ciphertext received by its private key. If the decryption is successful, p obtains two nonces and a principal ID and checks if the principal ID equals q and one of the nonces is the exact one that p has sent to q in this session. p then sends back to q an Ack message that contains n_q encrypted by the public key of q. On receipt of the message, q decrypts it, obtains a nonce and checks if the nonce is the one that q has sent to p in this session.

4. Formal Specification of NSLPK

We first introduce the following three operators (constructors) to represent three kinds of ciphertexts used in the protocol:

```
op enc1 : Prin Nonce Prin -> Cipher1 [ctor] .
op enc2 : Prin Nonce Nonce Prin -> Cipher2 [ctor] .
op enc3 : Prin Nonce -> Cipher3 [ctor] .
```

where `Prin` is the sort representing principals; `Nonce` is the sort denoting nonces; `Cipher1`, `Cipher2`, and `Cipher3` are the sorts denoting three kinds of ciphertexts contained in Init, Resp, and Ack messages, respectively. Given principals p, q and a nonce n_p, the term `enc1`(q, n_p, p) denotes the ciphertext $\mathcal{E}_q(n_p, p)$. `enc2` and `enc3` can be understood likewise. Hereinafter, let us use `Cipher1` (or `Cipher2`, or `Cipher3`) ciphertexts to refer to the ciphertexts contained in Init (or Resp, or Ack) messages.

The following operator (constructor) is used to represent nonces:

```
op n : Prin Prin Rand -> Nonce [ctor] .
```

where the third argument `Rand` is the sort denoting random numbers that makes the nonce globally unique and unguessable. Given principals p, q and a random value r, the term `n`(p, q, r) denotes a nonce created by p for q.

The following three operators (constructors) are used to represent the three kinds of messages used in NSLPK:

```
op m1 : Prin Prin Prin Cipher1 -> Msg [ctor] .
op m2 : Prin Prin Prin Cipher2 -> Msg [ctor] .
op m3 : Prin Prin Prin Cipher3 -> Msg [ctor] .
```

where `Msg` is the sort denoting messages. Given three principals c, s, r and a `Cipher1` ciphertext $ciph1$, the term `m1`$(c, s, r, ciph1)$ denotes an Init message such that c is the actual creator of the message, s is the seeming sender of the message, r is the receiver of the message and $ciph1$ is the message body. c may or may not be the same as s. If c is different from s, then the message must have been faked by the intruder. `m1` and `m2` can be understood likewise.

The network is modeled as associative-commutative collections of messages, which the intruder can use as his/her storage. Any message that has been sent or put once into the network is supposed to be never deleted from the network because the intruder can replay the message repeatedly, although the intruder cannot forge the first argument. Consequently, the empty network (i.e., the empty collection) means that no messages have been sent.

Let ms, rs, ns, and ps be collections of messages, random numbers, nonces, and principals, respectively. ps contains the intruder. Let $c1s$, $c2s$, and $c3s$ be collections of `Cipher1`, `Cipher2`, and `Cipher3` ciphertexts, respectively. To formalize the NSLPK protocol as a state machine M_{NSLPK}, we use the following observable components:

- (`nw` : ms) - it says that ms consists of all messages sent by principals and faked by the intruder;
- (`cenc1` : $c1s$) - it says that $c1s$ is the collection of the `Cipher1` ciphertexts gleaned by the intruder;
- (`cenc2` : $c2s$) - it says that $c2s$ is the collection of the `Cipher2` ciphertexts gleaned by the intruder;
- (`cenc3` : $c3s$) - it says that $c3s$ is the collection of then `Cipher3` ciphertexts gleaned by the intruder;
- (`nonces` : ns) - it says that ns is the the collection of nonces gleaned by the intruder;
- (`prins` : ps) - it says that ps is the collection of the principals participating in the protocol;
- (`rand` : rs) - it says that rs is the collection of random numbers available.

Each state in S_{NSLPK} is expressed as $\{obs\}$, where obs is a soup of those observable components. We suppose that three principals `p`, `q` and `intr` participate in the protocol, where `p` and `q` are trustable principals and `intr` is the intruder, and two random numbers `r1` and `r2` are initially available. Then, I_{NSLPK} consists of one initial state that is expressed as follows:

```
{(nw: emp) (rand: (r1 r2)) (nonces: emp)
 (cenc1: emp) (cenc2: emp) (cenc3: emp)
 (prins: (p q intr))} .
```

where `emp` denotes the empty collection.

Three rewrite rules `Challenge`, `Response`, and `Confirmation` formalize three actions when a principal sends an Init, a Resp, and an Ack message, respectively. Let `OCs` be a Maude variable of observable component soups; `P` & `Q` be Maude variables of principals; `Ps` be a Maude variable of collections of principals; `NW`, `R`, and `N` be Maude variables of collections of messages, random numbers and nonces, respectively; `Rs`, `CE1`, and `Ns` be Maude variables of collections of random numbers, `Cipher1` ciphertexts, and nonces, respectively. The rewrite rule `Challenge` is defined as follows:

```
rl [Challenge] : {(nw: NW) (prins: (P Q Ps))
(rand: (R Rs)) (cenc1: CE1) (nonces: Ns) OCs}
=> {(nw: (m1(P,P,Q,enc1(Q,n(P,Q,R),P)) NW))
(cenc1: (if Q == intr then CE1 else
(enc1(Q,n(P,Q,R),P) CE1) fi)) (nonces:
(if Q == intr then (n(P,Q,R) Ns) else Ns fi))
(rand: Rs) (prins: (P Q Ps)) OCs} .
```

The rewrite rule says that when `R` is in `rand`, a new Init message is put into the network, `R` is deleted from `rand`, the intruder gleans the nonce `n(P,Q,R)` if `Q` is the intruder and the intruder gleans the ciphertext `enc1(Q,n(P,Q,R),P)` if `Q` is not the intruder.

In addition to the three rewrite rules that formalize sending messages exactly following the protocol, we also introduce six more rewrite rules to formalize the intruder's faking messages:

- `fake12`, `fake22`, and `fake32`: a ciphertext `C` is available to the intruder, the intruder fakes and sends an Init, or a Resp, or an Ack message using `C`;
- `fake11` and `fake31`: a nonce `N` is available to the intruder, the intruder fakes and sends an Init or an Ack message using `N`;
- `fake21`: two nonces `N1` and `N2` are available to the intruder, the intruder fakes and sends a Resp message using `N1` and `N2`.

The rewrite rule `fake11` is defined as follows:

```
rl [fake11] : {(nw: NW) (nonces: (N Ns))
(prins: (P Q Ps)) OCs} =>
{(nw: (m1(intr,P,Q,enc1(Q,N,P)) NW))
(nonces: (N Ns)) (prins: (P Q Ps)) OCs} .
```

The rewrite rule says that when `N` is in `nonces`, a new intruder's faking Init message is put into the network.

5. State Picture Design of NSLPK Protocol

The network component, which consists of many messages, is the main part of the protocol that we should focus on. Initially, we try to make a design for the network in which Bui and Ogata [2] used, as shown in Figure 3. The

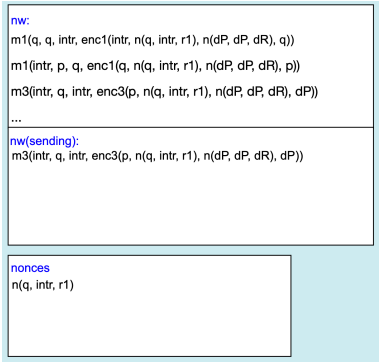


Figure 3: A simple state picture for the NSLPK protocol (1)

design, however, is hard to observe and/or analyze the messages in the network because there are many contents inside each message. As shown in Figure 3, there are three rectangles in which the first rectangle represents a network that contains all messages, the second one displays the most recent message that has been put into the network, and the collection of *nonces* gleaned by the intruder is displayed in the last rectangle. "..." is displayed whenever the content of the network is overflowed. During making a better state picture design, by observing that the number of messages increases by one after each state, we come up with an idea that displays the contents of the most recent message that has been put into the network (hereinafter, let us call such a message as the latest message).

Although there are three kinds of ciphertexts (i.e., `enc1`, `enc2`, and `enc3`), in the state picture design, we use only one form to visualize ciphertexts. The form is as follows: enc*i*(public-key, nonce1, nonce2, cipher-creator), where public-key is a principal (possibly `intr`), nonce1 for m1, m2, and m3 is in the following form: nonce1(generator, random, forwhom); nonce2 is in the following form: nonce2(generator, random, forwhom). When the ciphertext is in the form of `enc3`, cipher-creator receives a dummy principal `dP` as its value. Similarly, when the ciphertext is in the form of `enc1` or `enc3`, nonce2 receives a dummy value denoted by nonce2(`dP`,`dR`,`dP`), where `dR` denotes a dummy random number.

One possible way to observe&analyze the network is to observe&analyze each message in the network. Observing each message in the network is also equivalent to observe the latest message. Explicitly displaying the detailed content of the latest message helps us guess some non-trivial characteristics, which is discussed in Sect. 6. Furthermore, we design three sub-networks for three types of messages instead of one network that contains all messages. One network that contains all messages is another possible way to make the state picture design. Each way of design has some advantages as well as disadvantages. As shown in Figure 3, putting all messages in one place is simple but it is hard to distinguish each message. Designing three sub-networks for three types of messages helps us be able to immediately rec-

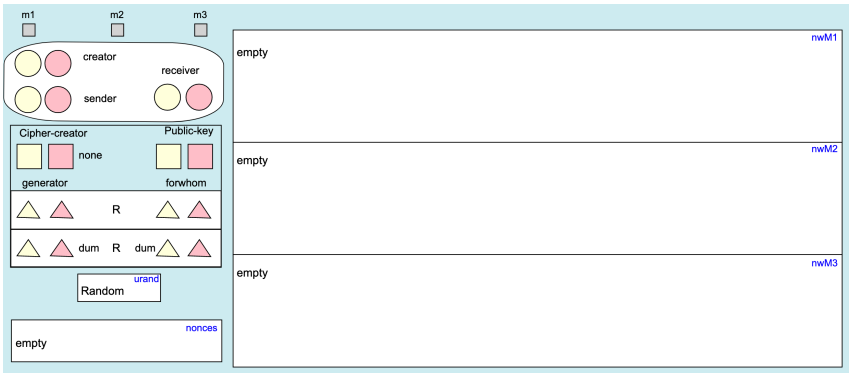


Figure 4: A state picture design for the NSLPK protocol (1)

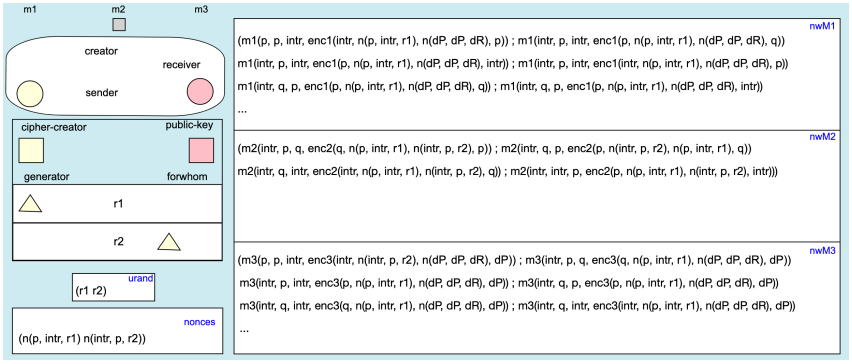


Figure 5: A state picture for the NSLPK protocol (1)

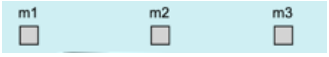
ognize the specific message type in each specific network. It makes us more transparent in our visual perception when we observe each specific message or the order/relation between messages.

In addition to the observable components presented in Sect. 4, some more observable components are introduced for visualizing the state picture design. They are as follows:

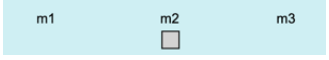
Observable components	Description
`newmsg`	The latest message (m1,m2,m3)
`m1`	Latest message `m1`
`m2`	Latest message `m2`
`m3`	Latest message `m3`
`nwM1`	Network contains messages `m1`
`nwM2`	Network contains messages `m2`
`nwM3`	Network contains messages `m3`
`urand`	Used random numbers
`nonces`	Nonces gleaned by intruder

Figure 4 depicts our state picture design. Some designs are used from state picture design tips of the work [3]. Figure 5 displays a state picture. We first divide two roles that are creators and senders into two separate places. Then, observable components are put to the corresponding place in which their roles seem to belong. For example, public-key should be put to the receiver's side because the sender uses the public-key of the receiver for encrypting. Values are displayed with different colors and shapes. For example, pink and light yellow colors represent two different principals, blank represents `intr`, triangles represent the contents of the nonce.

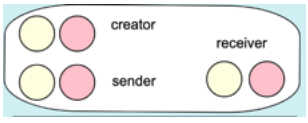
We describe the details of the state picture design. The representation of the three types of messages designed in Figure 4 is as follows:



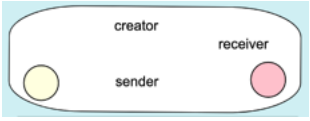
The type of the latest message is represented by a small light gray square. For example, when the latest message is a message `m2`, there is only one light gray square displayed under m2 as shown in the following picture:



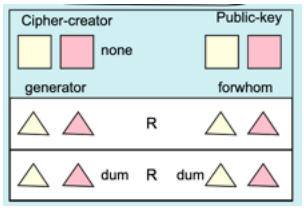
The representations of the creator, sender, and receiver of the message used in Figure 4 are as follows:



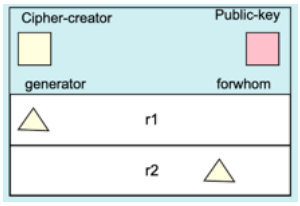
The creator of the message appears at the top-left place, pink and light yellow circles represent two different principals `q` and `p`. If the value is `intr`, nothing is displayed. The sender and receiver of the message appear at the bottom-left and bottom-right places, respectively. For example, when creator is `intr`, sender is `p`, receiver is `q`, it is displayed as follows:



The representations of the contents of the ciphertext shown in Figure 4 are as follows:



The cipher-creator of the ciphertext appears at the top-left place of the rectangle, pink and light yellow squares represent two principals `q` and `p`, respectively. If the value is `intr`, nothing is displayed. For the case the message is a message `m3`, the text "none" is displayed. The public-key of the ciphertext appears at the top-right place. If the value is `intr`, nothing is displayed. The two nonces of the ciphertext are shown with two rectangles inside the primary rectangle, where the upper rectangle visualizes the first nonce and the lower rectangle visualizes the second nonce. In the first nonce, the generator and forwhom representations appear at the left-hand side and right-hand side, respectively; pink and light yellow triangles are the principals `q` and `p`, respectively. If the value is `intr`, nothing is displayed. The random representation appears at the middle place in which the random number value used is displayed. The second nonce is represented likewise. If the message is a message `m3`, the text `dum` is displayed for the values of generator and forwhom, where `dum` denotes the dummy value `dP`. Considering the following example. cipher-creator is `p` and public-key is `q`. In the first nonce generator is `p`, random is `r1`, and forwhom is `intr`. In the second nonce, generator is `intr`, random is `r2`, and forwhom is `p`. Those values are displayed as follows:



In Figure 4, the representations of urand and nonces are designed at the left-bottom corner. The values of both urand and nonces are displayed using two rectangles as follows:

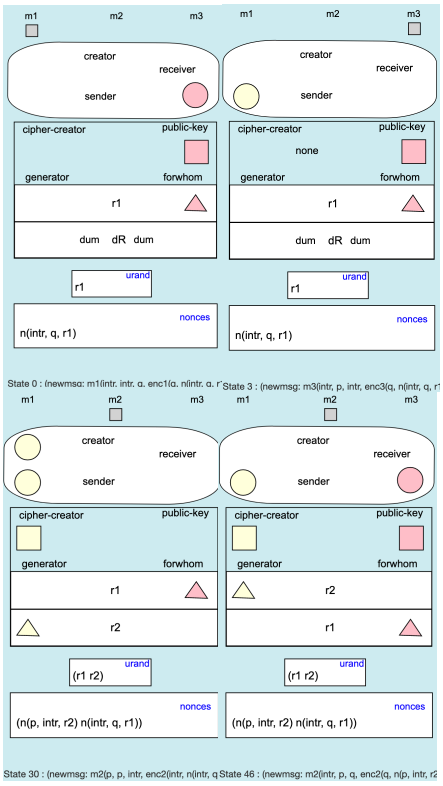
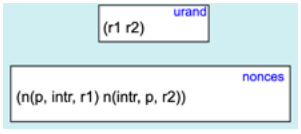
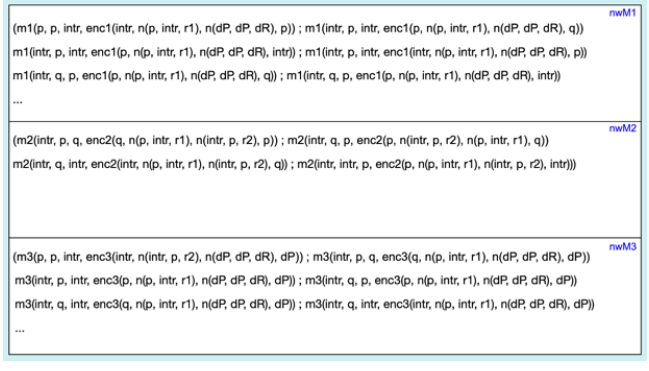


Figure 6: Some state pictures for the NSLPK protocol (1)



In Figure 4, three types of network representations are designed on the right side. "…" is displayed whenever the messages are overflowed. This can be seen in the figure below:



6. Characteristics Guessed Based on Our Design

This section presents how to guess the characteristics of NSLPK by observing graphical animations using SMGA. Observing graphical animations of a state machine allows

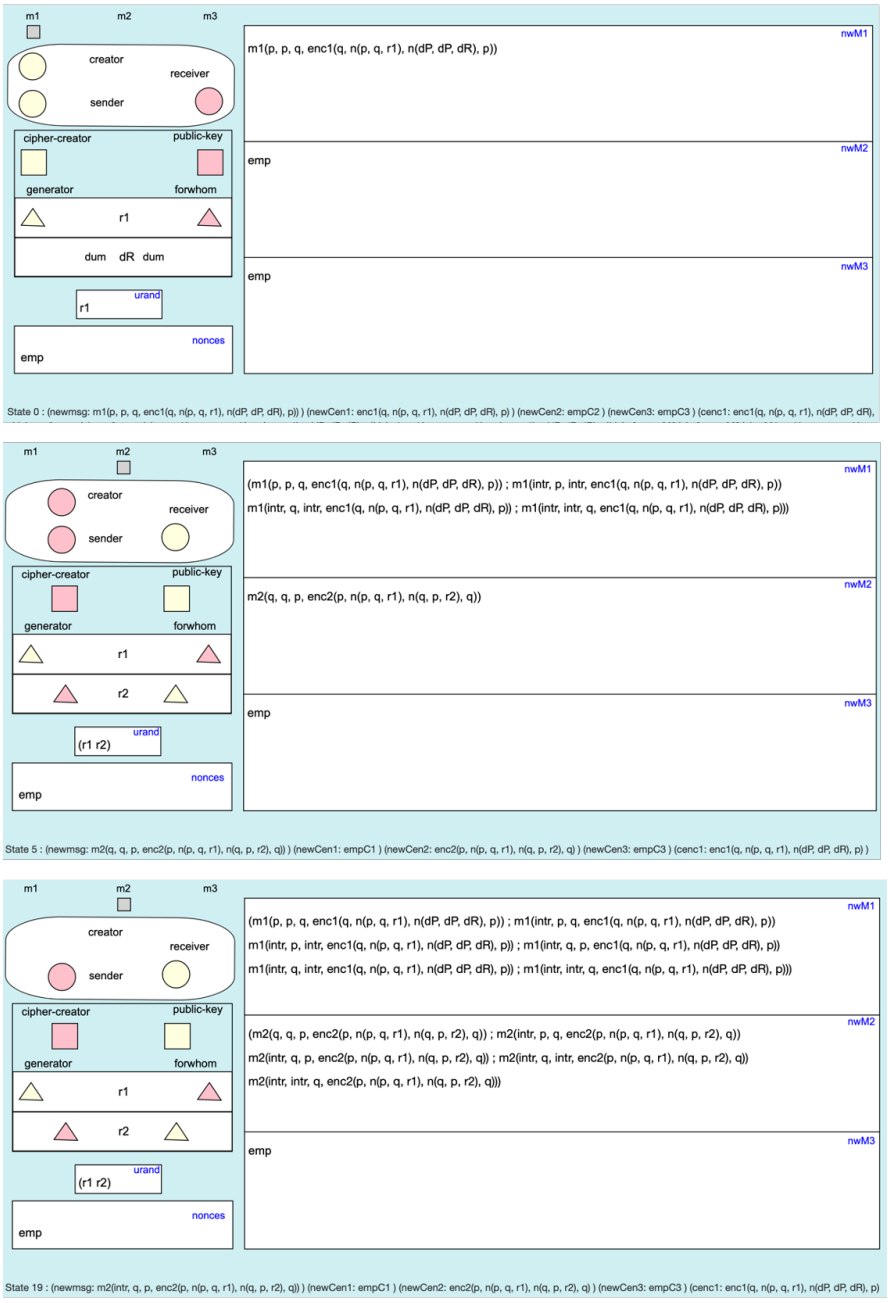


Figure 7: Some state pictures for the NSLPK protocol (2)

users to recognize some relations between observable components (OCs). Observing all OCs at the same time is less likely to recognize the characteristics since there are many OCs in the design picture.

There are some tips on how to conjecture characteristics of NSLPK by observing graphical animations with SMGA as follows:

1. By concentrating on one observable component, we may find that if the value of that observable component is `intr`, any other observable components may have some specific values, from which we may conjecture some characteristics.
2. By concentrating on two different observable components, we may find a relation between them, from which we may conjecture some characteristics.
3. By observing the order of the message in the network, we may find a relation between them, from which we may conjecture some characteristics.
4. By carefully investigating the conditions of some characteristics that have been already conjectured, we may conjecture some other characteristics.

Hence, we sometimes need to concentrate on some specific OCs when we observe the graphical animations. Characteristics of NSLPK that involve one message are straightfor-

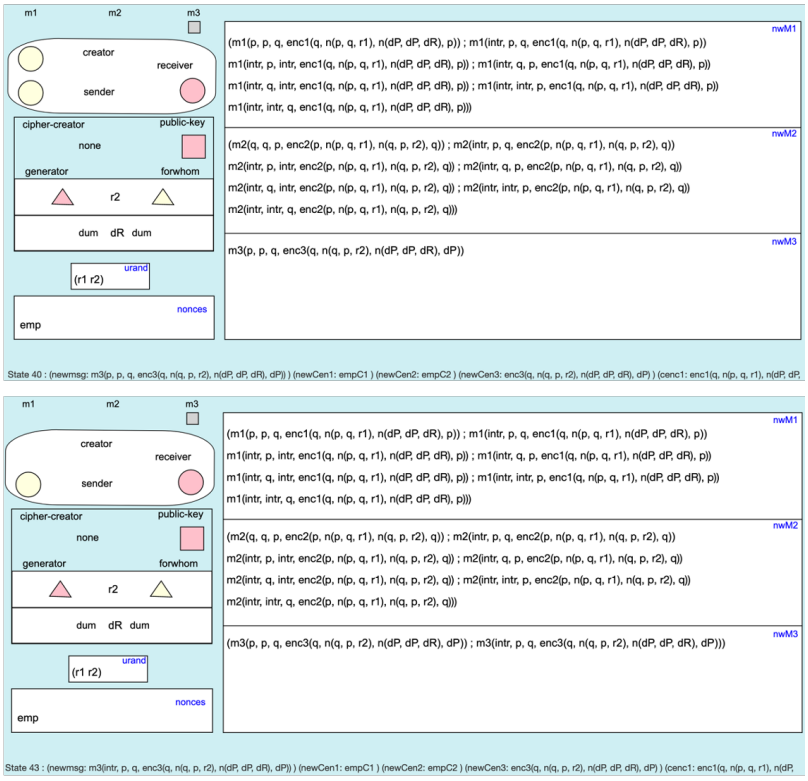


Figure 8: Some state pictures for the NSLPK protocol (3)

ward to guess by observing graphical animations. However, characteristics of NSLPK that involve two or more messages some are not.

Figure 6 shows four pictures of states for M_{NSLPK}. Taking a look at the first picture (of State 0) and the second picture (of State 3) helps us recognize that there is `n(intr,q,r1)` in `nonces` when `generator` is `intr` and taking a look at the third picture (of State 30) and the fourth picture (of State 46) helps us recognize that there is `n(p,intr,r2)` in `nonces` when `forwhom` is `intr`. Any nonce gleaned by the intruder is stored in `nonces`. Hence, observing the graphical animation of these four pictures helps us guess the characteristic such that any nonce gleaned by the intruder has been generated by the intruder or a non-intruder principal that wanted to authenticate the intruder.

Taking a look at the second picture (of State 3) and the third picture (of State 30) allows us to guess another characteristic such that whenever `receiver` is `intr` (that displays blank in the state pictures) in the latest message, then the nonce of that message is in `nonces`. Carefully observing graphical animations helps us perceive one more characteristic. Taking a look at the four pictures of Figure 6, we recognize the characteristic that when a nonce is in `nonces`, the random number used in the nonce is stored in the collection of used random numbers `urand`.

We prepare another input file that consists of a finite sequence of states so that we can guess more characteristics by observing the behavior of the protocol. To guess some non-trivial characteristics, we observe the information in which the order of the messages is mainly focused on. Carefully observing graphical animations of the order of messages in the network and expecting that a message `m2` should follow a message `m1`, we guess a characteristic which includes two messages as shown in Figure 7. Taking a look at the first picture (of State 0), there exists a message `m1(p,p,q,enc1(q,n(p,q,r1),n(dP,dP,dR),p))` in `nwM1`. After some `m1` messages are faked by the intruder based on the gleaned information, there exists a message `m2(q,q,p,enc2(p,n(p,q,r1), n(q,p,r2),q))` in `nwM2` at the second picture (of State 5). Taking a look at the third picture (of State 19), we observe that the intruder creates many faked `m2` messages including `m2(intr,q,p,enc2(p,n(p,q,r1),n(q,p,r2), q))`. Observing the order of messages in the network allows us to conjecture the following characteristic:

> If there exists a message `m1` created by a non-intruder principal and sent to another non-intruder principal, and
>
> there exists a message `m2` (either created by the intruder or a non-intruder principal) that is sent to the sender of `m1`,
>
> then the message `m2` originates from a non-intruder principal who is the receiver of the `m1`.

Similarly, we expect that a message `m3` should follow a message `m2`. There is a message `m2(q,q,p, enc2(p,n(p,q,r1),n(q,p,r2),q))` in `nwM2` at the second pic-

ture (of State 5). Taking a look at the first picture (of State 40) in Figure 8, there exists a message `m3(p,p,q,enc3(q,n(q,p,r2),n(dP,dP,dR), dP))` in `nwM3`. At the second picture (of State 43), there exists a message `m3(intr,p,q, enc3(q,n(q,p,r2),n(dP,dP,dR),dP))` in `nwM3` which is created by `intr`. Carefully observing the order of the messages in the network, we also guess the following characteristic:

If there exists a message m2 created by a non-intruder principal and sent to another non-intruder principal, and there exists a message m3 (either created by the intruder or a non-intruder principal) that is sent to the sender of the message m2,
then the message m3 originates from the non-intruder principal who is the receiver of the message m2.

Observing graphical animations of the NSLPK produced by SMGA could help us visually perceive several characteristics. The informal descriptions of the guessed characteristics are as follows:

1. If the latest message is a message `m1` and `cipher-creator` of `m1` is `intr`, then the nonce of `m1` is in `nonces` (i.e., the nonce is gleaned by the intruder).
2. If the latest message is a message `m1` that forms as `m1(p,p,q,enc1(q,n,p))` and `p` is not `intr`, then the forwhom of `n` is `q`.
3. If the latest message is a message `m2` that forms as `m2(p,p,q,enc2(q,n1,n2,p))` and `p` is not `intr`, then the forwhom of `n2` is `q`.
4. If the latest message is a message `m3` that forms as `m3(p,p,q,enc3(q,n))`, and `p` and `q` are not `intr`, then the generator of `n` is `q`.
5. If `public-key` of the latest message is `intr`, then a nonce/nonces in that message is/are in `nonces`.
6. If a nonce is in `nonces`, then either generator or forwhom of the nonce is `intr`.
   - If generator of a nonce is `intr`, the nonce is in `nonces`.
   - If generator and forwhom of a nonce are not `intr`, then the nonce is not in `nonces`.
7. If a nonce in the latest message forms as `n(p,q,r)`, and `p` is not `intr`, then `r` is in urand.
8. if a nonce is in `nonces`, then random of the nonce is in urand.
9. If message `m1(p,p,q,enc1(q,n(p,q,r),p))` is in `nwM1` and message `m2(q1,q,p, enc2(p,n(p,q,r),n,q))` is in `nwM2` and `p` is not `intr` then `m2(q,q,p,enc2(p,n(p,q,r), n,q))` is in the network and originates from `q`.
10. If message `m2(q,q,p,enc2(p,n,n(q,p,r),q))` is in `nwM2` and message `m3(p1,p,q, enc3(q, n(q,p,r))` is in `nwM3`, and `q` is not `intr` then `m3(p,p,q,enc3(q,n(q,p,r))` is in the network and originates from `p`.

Maude search command can be used as an invariant model checker to check that the NSLPK protocol enjoys the guessed characteristics. The guessed characteristics are confirmed by the search command at a specific depth (depth 5) of the state space because the reachable state space (generated by Maude) of the protocol is too huge to be exhaustively traversed. The search command does not find any counterexample at depth 5. It means that the NSLPK protocol seems to enjoy the guessed characteristics.

7. Graphical Animation in Sequence Diagram

7.1. Idea

Sequence diagram is used to model the interactive behavior system entities, which is one of the most used diagrams of UML [13]. Besides, message sequence charts (MSCs) are widely used to capture system requirements during the early design stages [6]. A variant of MSCs is also called sequence diagrams used in UML. The Alice-Bob format that is often used to describe security protocols is a kind of sequence diagrams. Therefore, sequence diagram is one possible way to visualize message exchanges between principals in authentication protocols. We develop SMGA-SD that automatically generate a sequence diagram from a sequence of states and integrate SMGA-SD with SMGA. Regarding security protocols, such as NSLPK, messages from a principal are not delivered immediately to the recipient but stored in the network so that intruders can intercept and/or replay messages. Hence, our sequence diagrams are designed slightly differently from standard sequence diagrams in that a message is not delivered immediately to the recipient. We suppose that messages in the network never be deleted. The behavior of intruders makes protocols unpredictable, which may lead to the middle-person attack [8]. To express principals (including the intruder) in NSPK and NSLPK, we draw three parallel vertical lines denoting three principals where one principal is intruder whose line is in the middle, and two others represent two trustable principals. Although we can draw as many principals as many vertical lines in SMGA-SD, for simplicity, we keep the current appearance of our sequence diagrams. Horizontal arrows represent messages exchanges between principals. The message content is displayed above in the middle position of the arrow. There are two kinds of messages in which (1) one follows the protocol and (2) the other is faked by the intruder. To distinguish the two kinds of messages, the blue color is used for (1), while the red color is used for (2). Some functionalities for animations in SMGA are applied to SMGA-SD, such as *Run*, *Stop*, *Run step*, *Back step*, so that users can control the animation of the sequence diagram.

Let us describe two main different points of SMGA-SD compared to the standard sequence diagram as follows:

1. In standard sequence diagrams, the target of arrow messages is drawn directly to principals. In our diagram, we assume that messages sent are first put into the network. Those messages are intercepted and/or replayed by intruders mentioned above. Therefore, the target of arrow messages is not drawn directly to principals except for the case in which the recipient is in-

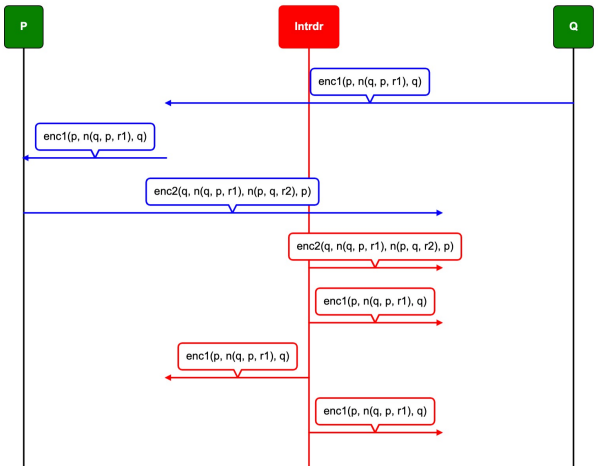


Figure 9: A snapshot of sequence diagram for NSLPK protocol

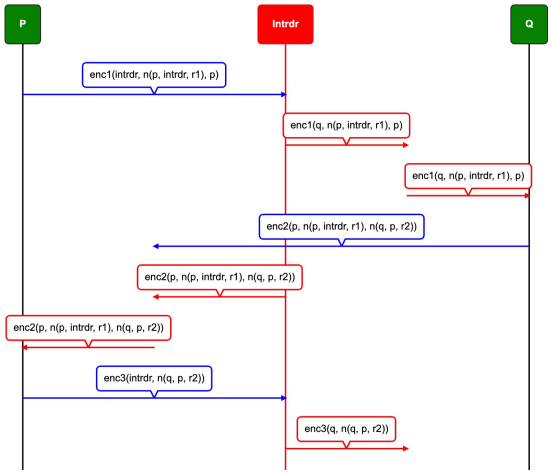
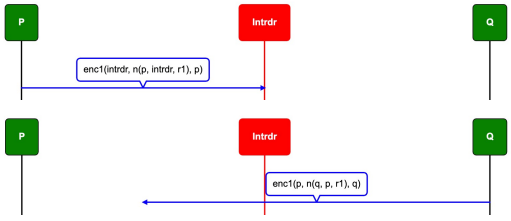


Figure 10: A sequence diagram for NSPK protocol

truder. Because the intruder can intercept all messages in the network, messages sending to the intruder can be regarded as received by the intruder without delay. The following figure shows two cases: principal `P` sends a message to intruder `Intrdr`, where the source of arrow message starts from `P` and ends at `Intrdr` at the top of the figure, and principal `Q` sends a message to principal `P` where the source of arrow message starts from `Q` and ends at the point between `P` and `Intrdr` at the bottom of the figure, which means that the message is not delivered to principal `P` yet, but it is stored in the network.



2. When a principal obtains a message (called a received message) from the network, depending on what the message is, the principal can produce a new message to reply to it. The figure below shows a case when principal `Q` sends a message to principal `P` (the first arrow) but has not delivered to `P` yet until principal `P` receives the message and produces a new message to send back to principal `Q` (in the second and third arrows, respectively). Of course, the message sent by principal `P` is not delivered to principal `Q` yet.

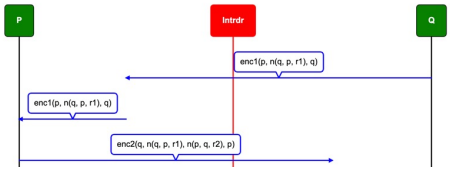
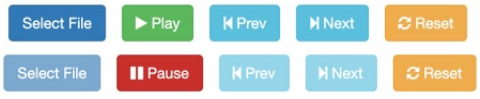


Figure 9 shows a sequence diagram of NSLPK. There are five buttons: *Select File*, *Play*, *Prev*, *Next*, *Reset* that correspond to five functionalities, which are the same as in SMGA, as follows:

- *Select File* to import a state sequence file.
- *Play* to draw a sequence diagram step by step with a speed selected by users.
- *Pause* to stop drawing a sequence diagram. When a user clicks *Play* button, the button becomes *Pause* button. The following figure shows the moment before and after clicking *Play* button.



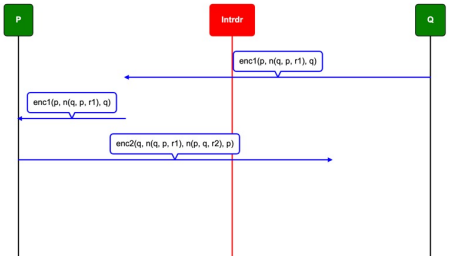
- *Next* to go forward to the next diagram in one step (one state transition).
- *Prev* to go back to the previous diagram in one step.
- *Reset* to reset the diagram to the beginning when a user just imports a state sequence.

7.2. Graphical Animations of NSLPK in Sequence Diagram

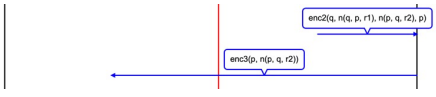
In the sub-section, we describe how to visualize NSLPK as sequence diagrams by SMGA-SD. The work flow is the same as SMGA. Firstly, we use Maude to formalize NSLPK and generate a state sequence as an input file. The input file is then imported into SMGA-SD. Apart from some existing observable components used in SMGA, we use a new observable component called `recmsg1` in the specification that represents the received message. All observable components used SMGA-SD are summarized as follows:

- (prins: (p q intrdr)): we use three values of prins to display the name of three principals on the diagram.
- (newmsg: message): the message format is same as what we defined in Sec. 4.
- (recmsg1: message): this observable component is used with newmsg to display the received message. This observable component stores the message that is m1 (in case the rl [Response] is used), m2 (in case the rl [Confirmation] is used), and empt (for the rest cases). Note that the intruder always gets the messages in the network as our assumption so we do not need to display the received message of the intruder. If recmsg1 observable component is m1, the newmsg observable component must be m2.

  An example below shows the newmsg and recmsg1 components in which newmsg is m2 and recmsg1 is m1:

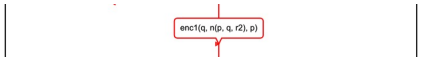


  If recmsg1 observable component is m2, the newmsg observable component must be m3. An example below shows the newmsg and recmsg1 observable components in which newmsg is m3 and recmsg1 is m2:



To be able to distinguish the messages following the protocol and those being faked by the intruder, we modify the content of messages in the specification in which a boolean value is added denoting that a message is faked by intruders or not. When drawing a message, we check the boolean value to decide the color of the message. If it is true, the red color is used. Otherwise, the blue color is used.

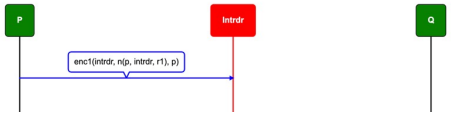
While observing the diagram for the first time, we have found that some states are meaningless, such as the intruder sends messages to himself/herself. Then, we modify the specification to avoid that situation. The following figure displays a message in which the intruder sends the message to himself/herself:



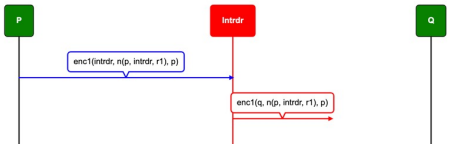
Looking at the diagram in Figure 9 makes us immediately distinguish two kinds of messages: one is message sent following the protocol and the other is those being faked by the intruder. We can recognize the sender and receiver of each message and comprehend their order.

7.3. Graphical Animations of NSPK in Sequence Diagram

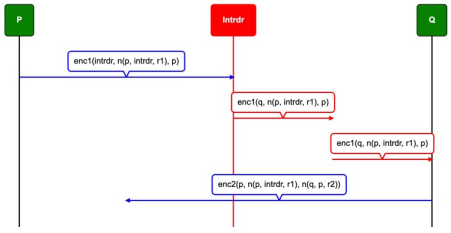
NSPK is flawed as detected by Lowe [8] and does not enjoy the nonce secrecy property (NSP). We use the Maude search command to find out a state sequence in which NSP is broken. The state sequence is then used to visualize a sequence diagram by SMGA-SD as in Figure 10. Looking at the diagram in Figure 10 makes it easier for us to understand why the property is broken. Note that by running animations step by step, we may better comprehend this flaw. Let us describe each state when using animations. Firstly, principal P sends message m1 to intruder as follows:



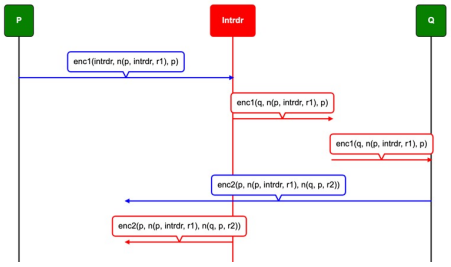
Then intruder fakes message m1 and sends it to principal Q as follows:



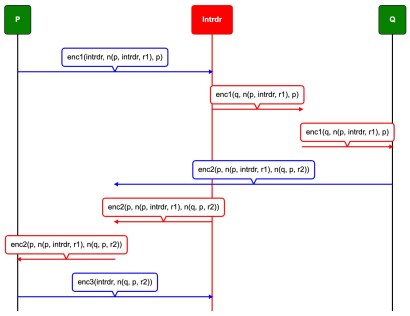
Whenever principal Q receives message m1 faked by intruder and the seeming sender is P, principal Q then sends message m2 to principal P as follows:



At this time, intruder can intercept message m2 and replay it to principal P as follows:



Whenever principal P receives message m3 faked by intruder and the seeming sender is intruder, it sends message m3 back to intruder as follows:



Finally, intruder fakes `m3` message and sends it to principal Q as shown in Figure 10.

8. Conclusion

We have graphically animated NSLPK and NSPK with SMGA and SMGA-SD. NSLPK has been visualized in a way that is different from the Alice-Bob format, while NSPK, together with NSLPK, has been basically visualized in the Alice-Bob format, although there are some differences between our sequence diagrams and standard sequence diagrams. Observing the graphical animations based on our first original design allows us to guess some (non-trivial) characteristics of the state machine formalizing NSLPK. We have checked the characteristics by Maude search command. Using the middle-person attack to NSPK, we have described how to use our sequence diagrams to help users comprehend why the attack is doable for NSPK. One piece of our future work is to graphically animate state machines that formalize other authentication protocols with SMGA, such as TLS [5].

Acknowledgment

The authors would like to thank the anonymous reviewers who carefully read an earlier version of the paper and gave them valuable comments without which they were not able to complete the present paper.

References

[1] Aung, M.T., Nguyen, T.T.T., Ogata, K., 2018. Guessing, model checking and theorem proving of state machine properties – a case study on Qlock. IJSECS 4, 1–18. doi:10.15282/ijsecs.4.2.2018.1.0045.

[2] Bui, D.D., Ogata, K., 2019. Graphical animations of the Suzuki-Kasami distributed mutual exclusion protocol. JVLC 2019, 105–115. doi:10.1007/978-3-319-90104-6_1.

[3] Bui, D.D., Ogata, K., 2020. Better state pictures facilitating state machine characteristic conjecture, in: DMSVIVA 2020, pp. 7–12. doi:10.18293/DMSVIVA20-007.

[4] Clavel, M., Durán, F., Eker, S., Lincoln, P., Martí-Oliet, N., Meseguer, J., Talcott, C. (Eds.), 2007. All About Maude. volume 4350 of *LNCS*. Springer. doi:10.1007/978-3-540-71999-1.

[5] Dierks, T., Allen, C., 1999. The TLS protocol version 1.0. RFC 2246, 1–80. doi:10.17487/RFC2246.

[6] Harel, D., Thiagarajan, P.S., 2003. Message Sequence Charts. Springer US, Boston, MA. pp. 77–105. URL: https://doi.org/10.1007/0-306-48738-1_4, doi:10.1007/0-306-48738-1_4.

[7] K. W. Brodlie, et al. (Ed.), 1992. Scientific Visualization: Techniques and Applications. Springer. doi:10.1007/978-3-642-76942-9.

[8] Lowe, G., 1995. An Attack on the Needham-Schroeder Public-Key Authentication Protocol. Inf. Process. Lett. 56, 131–133. doi:10.1016/0020-0190(95)00144-2.

[9] Mon, T.W., Bui, D.D., Duong, T.D., Ogata, K., 2021. Graphical animations of NSLPK authentication protocol, in: 27th DMSVIVA, pp. 39–45. doi:10.18293/DMSVIVA2021-005.

[10] Needham, R.M., Schroeder, M.D., 1978. Using Encryption for Authentication in Large Networks of Computers. Commun. ACM 21, 993–999. doi:10.1145/359657.359659.

[11] Nguyen, T.T.T., Ogata, K., 2017a. Graphical animations of state machines, in: 15th DASC, pp. 604–611. doi:10.1109/DASC-PICom-DataCom-CyberSciTec.2017.107.

[12] Nguyen, T.T.T., Ogata, K., 2017b. Graphically perceiving characteristics of the MCS lock and model checking them, in: 7th SOFL+MSVL, pp. 3–23. doi:10.1007/978-3-319-90104-6_1.

[13] Van Amstel, M.F., Lange, C.F., Chaudron, M.R., 2007. Four automated approaches to analyze the quality of uml sequence diagrams, in: 31st Annual International Computer Software and Applications Conference (COMPSAC 2007), pp. 415–424. doi:10.1109/COMPSAC.2007.119.

Journal of Visual Language and Computing

Volume 2020, Number 2