HU ISSN 1785-1270

PRODUCTION SYSTEMS AND INFORMATION ENGINEERING

A Publication of the University of Miskolc

VOLUME 7 (2015)

MISKOLC UNIVERSITY PRESS

HU ISSN 1785-1270

PRODUCTION SYSTEMS AND INFORMATION ENGINEERING

A Publication of the University of Miskolc

VOLUME 7 (2015)

MISKOLC UNIVERSITY PRESS

Secretariat of the Vice-Rector for
Research and International Relations
University of Miskolc
Responsible for publication: Prof. Dr. Tamás Kékesi
Published by the Miskolc University Press
under the leadership of Attila Szendi
Responsible for duplication: Erzsébet Pásztor
Number of copies printed: 70
Put to the Press on April 4, 2016
Number of permission: TNRT 2016-100-ME
HU ISSN 1785-1270

Production Systems and Information Engineering
Volume 7 (2015), pp. 5–14

AUTOMATIC GESTURE GENERATION

László Czap
University of Miskolc, Hungary
Department of Automation and Infocommunication
czap@mazsola.iit.uni-miskolc.hu

Roland Kilik
University of Miskolc, Hungary
Department of Automation and Infocommunication
kilik@mazsola.iit.uni-miskolc.hu

[Received September 2015 and accepted November 2015]

Abstract. The usage of virtual agents make necessary to produce natural motions and gestures. The paper introduces a novel solution which generates vertical head motion from the speech. The method uses neural networks for estimating the positions of the head. The outputs of the proposed algorithm have validated by human feedback. The results show that the naturalness of the generated motions is similar to the original ones.

Keywords: gesture generation, neural networks

1. Introduction

Our initial assumption, according to which both the intensity of the sound, and its pitch have a given time domain relationship with head gestures, was proved by correlation methods in our prior publications [3].

At the phase of the research that described in this paper our goal was automatic head gesture generation from the sound, based on the prior results. We tried to give a solution not only for the automatic generation of vertical and horizontal movements, but the blinks as well.

The most advanced part of the research is the field of generating the vertical movements, and the paper mostly describes its methods, procedures and results.

The teaching and testing samples for our two neural networks used for the vertical gestures, were alltogether 100, one sentence video files, what were downloaded from the web, or made by the authors with starring different subjects.

The topic of the paper is automatic gesture generation. After mentioning the results of the related researches, the paper describes the methods, procedures and current results of generating vertical movements of gestures from the sound, that developed and reached by the authors. After showing the current stage of the validation, the research's other phases are mentioned in the paper.

2. Related work

A partial basis of our research was the result of other works, that stated for example that there is a relationship between different gestures (for example movements of the hands), or the movments of the lips and the speech features [1, 2]. However, these researches did not examine the time-domain relationships between those, and produced head movements with very different approach and with restrictions, furthermore did not produce blinks from the sound.

In [2] the authors use *Hidden Markov Model*. Our proposed solution is based on *Artificial Neural Network* and this way it is able to avoid some post processing steps. In the case of HMM implementation the algoritm results motion segments and necessary to join them in a further step. The neural network is a better choice in this sense because it can determine the positions while processing the given segment of the speech.

3. Teaching and testing samples

The teaching videos for the first network were fully spontaneous, while the ones for the second network were not, but still mainly performed by the announcer's own words. The testing videos were spontaneous speeches.

4. Preprocessing

Both the neural network teachings, and testings were performed with sentences, that were created from the above mentioned video files.

The movement vectors of the sample sentences (for the testing and the training) were produced partially by a Java program that created by the authors that followed the eyes of the subjects, and by manual determination of the eye positions as well.

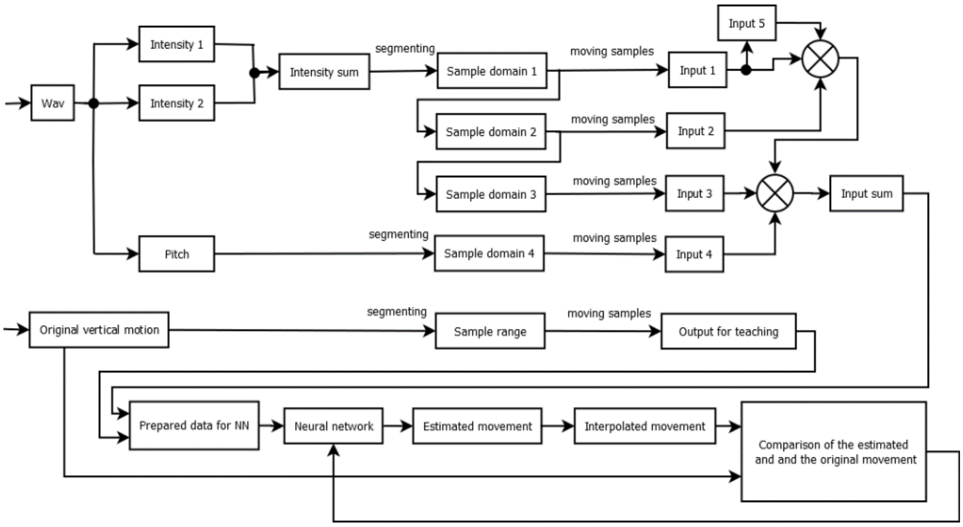

Figure 1. The teaching procedure and the input structure of the first neural network

The pitch vectors of the sound were produced by Praat software, and the intensity vectors (intensity1 on the figure 1) were calculated by the following formula with MATLAB:

```
intensity1(i) = sqrt(sum(window .^ 2));
```
(4.1)

The elements of the vector are the square root of the sum of the square of every element of the window, where the window was always a frame-sized part of the amplitude vectors.

As the first figure shows, the initial inputs of our first neural network were mainly derivatives of the intensity vectors of the given teaching samples, and the input of the other neural network was only a pitch vector.

As it can be seen on the Figure 1, a second form of the intensity was also used (`intensity2`), which was calculated on a vector that we got by a 25 window size smoothing mean on the amplitude vector. The calculating program code was:

```
intensity2(i) = mean(window .^ 2);
```
(4.2)

Every element of the vector is the mean of square of every element of the window, where the window here was a frame-sized part of the previously smoothed and meaned amplitude vector.

Finally, `intensity1`, and `intensity2` were summed by the following formula (after simplification):

```
intensity_sum = intensity1 * 0.735 + intensity2;
```
(4.3)

The segmented version of this summary vector (sample domain 1) was one of the five for neural network no. 1. The second (sample domain 2) vector was produced by subtracting the values of the previous element from each element's value in the sample domain 1 vector. The third (sample domain 3) vector was calculated by subtracting the values of the previous element from the value of each element in the sample domain 2 vector. The fourth sample domain vector was the segmented version of the pitch vector. The segmentation was carried out by having the syllable starting times in each sentences (that were manually marked), and calculating the segmentation time by having the time of the maximum amplitude between every two syllable starts in the voice.

As teaching and testing movement vectors, vertical movements were used. These movement vectors were the differences between the vertical position coordinates of the left eye on every frame, finally divided by the mean of them. These were also segmented with the same procedure as the sound-originated vectors, and used as sample range vectors.

Atfer preparing the sample domain vectors, corresponding input vectors were created from them, and from the segment range vector, by a moving sample procedure. This procedure prepared the data for prediction, by pairing the following element from the movement vector (sample range), to every consecutive two elements from the given sample domain vector. The procedure made 4 input vectors from the 4 sample domain vectors, and a 5 th input, which was created by elements made from shifting elements of the input1 by one to the left). An output vector was also created by the procedure from the sample range vector for teaching the neural network.

For the second neural network, the first sample domain vector was the pitch vector (Figure 2). At this stage there was only one input vector, which was created by the same moving sample function as used for the first neural network.

The neral networks were nonlinear autoregressive networks with feedback [4], trained in open loop mode, and then closed for the testings and used in closed loop state. The structure of the networks can be seen on Figure 3 and Figure 4.

As can be seen from the figures, the mainly intensity-based first neural network has an input delay 4, and a layer delay 4, while the second, pitch-based network has input delay 2, and a layer delay 2.

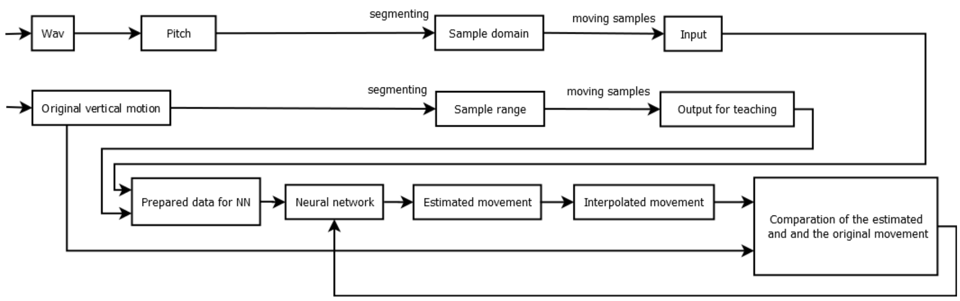

Figure 2. The teaching procedure and the input structure of the second neural network

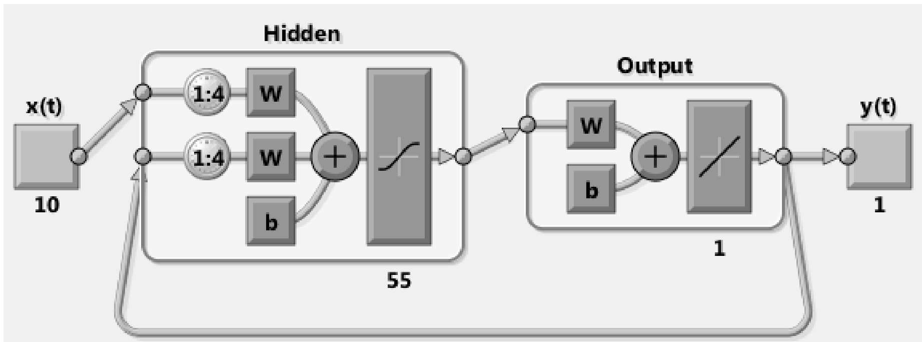

Figure 3. The structure of the first neural network

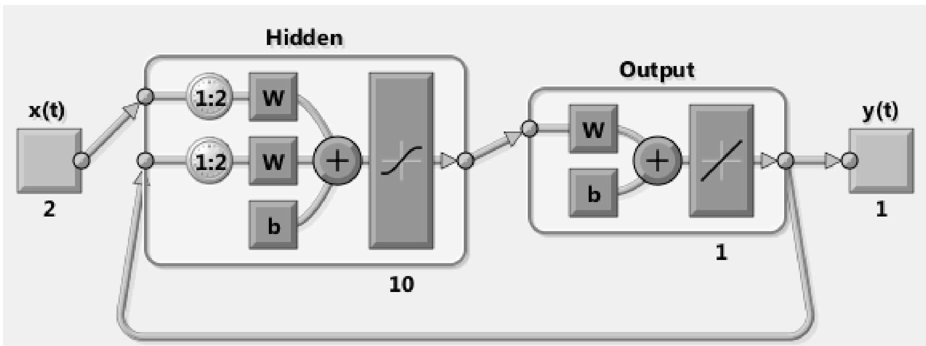

Figure 4. The structure of the second neural network

Not only does MATLAB'S data preparing function order the input and output due to the delays of the network, but it also cuts the same number of elements from the input and the target as the number of the delay. Therefore, at first, we supplemented the beginning of our vectors by zeros for these to be cut, in order to avoid the shortening of our useful sample sizes. This also made possible to have predicted motions even at the beginning of the sentences.

5. Postprocessing

After the predictions of the two networks, the amplitudes of the outputs of the neural networks were corrected. At this stage it was accomplished only by constants. The values of the multipliers were determined by the volunteers, as they saw results produced by different multiplier values, and chosen those for the best. The outputs of the intensity-input neural network were multiplied by 2 (and resulted in a high amplitude, frequent movement), and the outputs of the pitch-based neural network were multiplied by 4 (and resulted a lower amplitude, less frequent movement). This method was suitable for most of the videos, but for others, the intensity-based neural network's results had to be divided by 4 in amplitude, and the results of the other network were not modified. There is an ongoing procedure that could predict two things from some of the sound features. One of them is about which of the 2 neural network's output would be more natural by the opinion of the viewers .The other procedure is concerned with how much amplitude correction is would be the best for the given example.

As both the inputs and outputs were segmented vectors, the outputs of the neural networks had to be finally interplolated to have the movement for every frame.

The corrections of the amplitude, the inputs of the networks, and the suitability of the two selected networks (from the plenty of trained and tested ones) were mostly determined by the authors and users. The method was to compare the naturalness of the generated movements with the original and random movement, they could saw on an announcer's picture being moved vertically by those movements. Before this, as a precondition, an acceptable similarity had to exist with the selected network between the original and generated movement vectors.

On Figure 5 some of the segmented, and concatenated teaching sample's movement vectors and the first neural network's predicted output vectors can be seen. It can be observed, that the similarity is not tight, but the goal of the research is to produce a result movement for sounds that seems natural to the viewer rather than producing a movement vector that is similar to the original one.

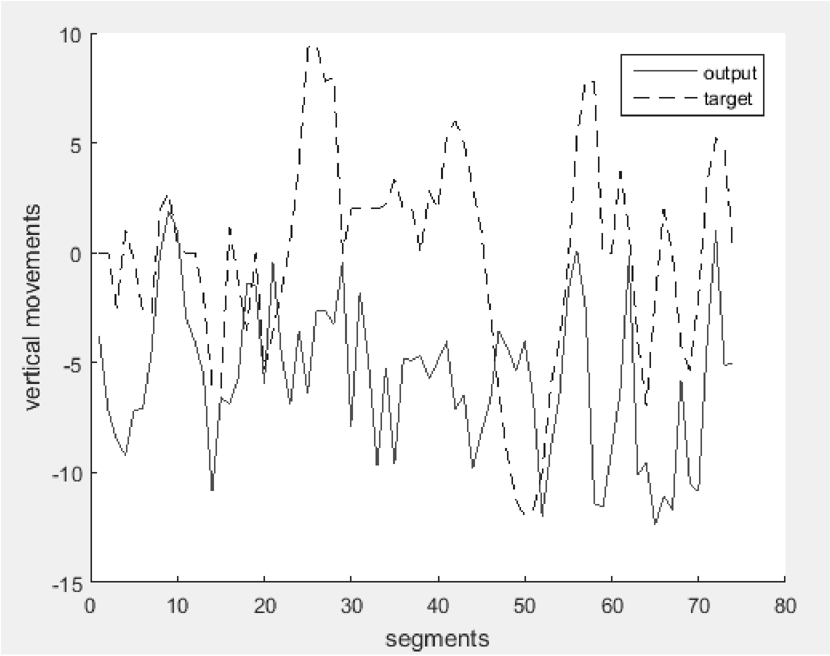

Figure 5. Output and target for a part of teaching samples for neural network 1.

On Figure 6, what is the original movement and target movement of some of the second neural network's teaching samples, the above statements are also true.

6. Testings and results

After the authors had chosen some of the previously created networks performing better in the objective similarity than others, 10 test videos were tested by 6 volunteers at this stage. At this phase of tests, only a still frame of an announcer was moved with the given frame rate while playing the sound of the sentence at the same time. For every test, a moving picture was created this way with the original movement, with random movement with the same co-domain as the original, and two with the neural network generated outputs.

These tests resulted in choosing the two best neural networks, refinements of the amplitude corrections, a predicted value correcting function (with an output-correcting neural network), and hand-made corrections in the results, each according to the authors' and the volunteers' opinions. Having summed

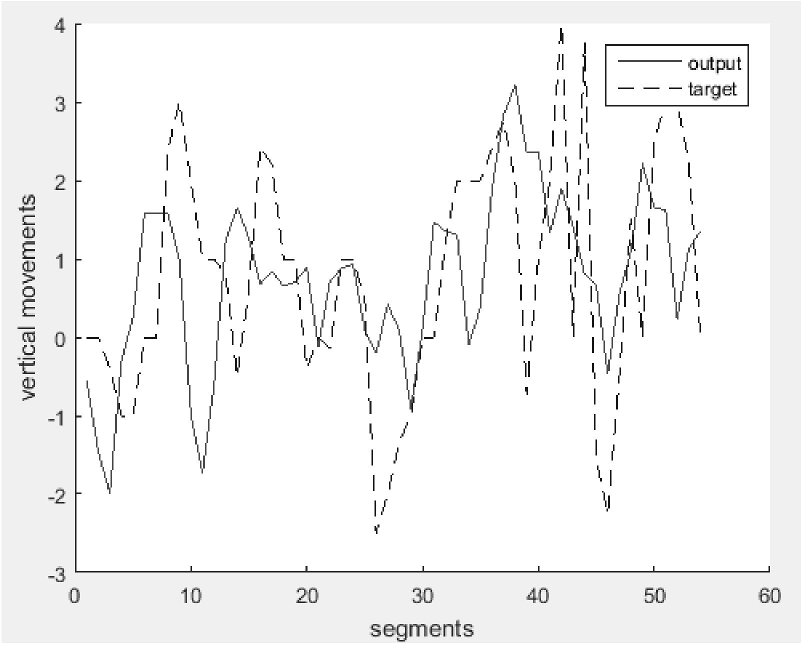

Figure 6. Output and target for a part of teaching samples for neural network 2.

those corrections, a neural network retraining was carried out with the modified teaching target diagram to make the output similar to this, without having the correctional network and hand-made corrections.

As a result of this, 9 from the 10 testing samples described both of the two final neural network's result as more natural than the random movement in almost every cases. About half of them, most of the volunteers stated that the video with neural network generated movement was even more natural, than the one with the original movement.

On Table 1 some of the results can be seen.

In the table each cell contains the rank of the naturalness of the given movement variant for the video, by the (A, B, C or D) volunteer. Examining the first table, it can be seen for example for the first test video, that first neural network's result described as the most natural one by A, B, and D volunteers, and the least natural (4-th place) by one volunteer D. Neur. 2/a means the results with the older amplitude corrections to the results of the second neural network (which were related to the original amplitude), and 2/b version means the results with an amplitude correction that is independent of the original movement's amplitude.

	random				original				neur 1.				neur 2./a				neur 2./b			
volunt.	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D	A	B	C	D
video 1	4	4	1	4	2	1	2	2	1	1	4	1	3	3	3	3	-	-	-	-
video 2	4	4	3	4	3	1	2	2	2	3	4	1	1	2	1	3	-	-	-	-
video 3	4	1	4	4	2	2	1	2	3	4	3	3	1	3	2	1	-	-	-	-
video 4	4	4	4	4	3	3	3	3	2	2	1	2	-	-	-	-	1	1	2	1
video 5	3	5	5	5	2	3	3	2	1	4	1	4	5	1	3	1	4	2	2	3
video 6	5	4	5	5	2	1	3	4	1	2	1	1	4	5	2	2	3	3	4	3

Table 1. Some user opinions on some videos

Since the purpose of the tests was mainly to promote the improvement of the methods of the generations and corrections, the volunteers were not asked to give their opinions at the same time, and not every video was required to face opinions. The table contains only those videos that the most of the volunteers were asked to give opinion about.

As stated above, the similarity of the original and generated movements was not a goal. A verification of this approach is for example the result of a video that can be seen on figure 7, (video 6 on Table 1) where there are many differences between the original and the generated version of movement, however the generated one was even more natural according to the opinions.

7. Future works

It is important to note that the purpose of these opinion tests of users was mainly to help in choosing the proper networks and creating the necessary corrections. The real tests, with a larger number of volunteers, and using a virtual speeching head, will be the scope of the next phase of the research, as also the correction and the test cases of the horizontal movements and blink generation.

Acknowledgements

The research has beeen carried out within the framework of Mechatronics and Logistics Centre of Excellence operating as one of the strategic research

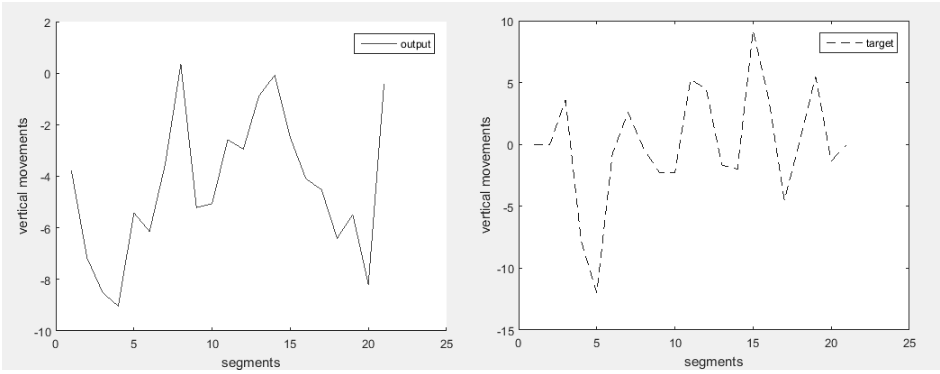

Figure 7. Neural network's estimated movement and the original movement for a given test video

area of the University of Miskolc and as the part of the TÁMOP-4.2.2.C-11/1/KONV-2012-0002 project funded by the European Unio, cofinanced by the European Social Fund.

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] Busso C.; Deng C.; Zhihang, Narayanan, S.; Neumann, U., : *Learning expressive human-like head motion from speech, Data-Driven 3D Facial Animation*, Springer, 2008.

[2] Rubin, P; Yeiha, H.; Vatikiotis-Bacon, E.: *Quantitatice association of vertical tract and facial behavior*, Speech Communication 26., 23-43., 1998.

[3] Czap, L. and Kilik R.: *Prelimenary Examinations For Automatic Gesture Generation*, Alkalmazott Nyelvészeti Közlemények, Miskolci Egyetemi Kiadó, 105-113., ISSN 1788-9979, 2015.

[4] Billings, S. A.: *Nonlinear System Identification: NARMAX Methods in the Time, Frequency, and Spatio- Temporal Domains*, Wiley, ISBN 978-1-1199-4359-4, 2013.

Production Systems and Information Engineering
Volume 7 (2015), pp. 15–31

THE EFFECT OF THE SUPPLY ACCURACY ON THE INVENTORIES

János Korponai
József Hatvany Doctoral School in Information Sciences
janos.korponai@gmail.com

Ágota Bányainé Tóth
University of Miskolc, Hungary
Institute of Logistic
altagota@uni-miskolc.hu

Béla Illés
University of Miskolc, Hungary
Institute of Logistic
altilles@uni-miskolc.hu

[Received September 2015 and accepted November 2015]

Abstract. Our theories applied in the case of the economic order quantity are influenced by the fact that disturbances can occur in our internal and external processes that together create an uncertainty regarding the arrival date of purchased parts and the development of the stock level. By walking around the continuous review model during our analysis, we will present the relations between the fluctuation of supply accuracy and the safety stocks, and we will define the formula for the quantification of the safety stock. The definition of the service level provides an opportunity to handle the fluctuation of replenishment period with a predetermined safety, and to quantify the stock level required for the operation.

Keywords: supply accuracy, replenishment period, safety stock, standard normal distribution, random variable

1. Introduction

All production companies require for operation the acquisition of items from external resources, and their arrival by the specified deadline. When the stock replenishment can be exactly on time realized, the level of the closing stock, the quantity of the placed order and the actual date of the order can be clearly defined with the knowledge of the initial stocks, the production demand and the restocking of supplies. However, this initial condition is very rare in practice. There are several unpredictable factors influencing the supply accuracy that will affect the operation of the production company. The objective of the logistics management is to guarantee the stock level required for the adequate handling of production at the lowest possible level of costs [1].

2. The relationship between the deterministic supply accuracy and the stock level

The importance of supply accuracy on the inventories and supply chain efficiency is growing. The operation research and the management sciences literature includes a huge number of methods and models to support the design, optimisation and control of purchasing processes from the point of view supply accuracy [2].

The integration is a core problem of supply in the field of logistics, because the processes of purchasing and production have bidirectional effects. The integration of the first two functional parts of logistics (purchasing, production) is especially important in the case of multistage supply [3].

The supply chain problem includes a wide area of decisions: supplier selection [4], payment management of supply processes [5], coordination of multi-level, vendor managed inventory systems [6], consignment stock management [7], risk management of supply chain and stock management [8]. The modelling of supply processes can be described from the point of view analysis and optimization of warehouses, and there are important sources proposing approach to reduce inefficiencies using value stream mapping [9]. The supply chain accuracy is influenced by the size, operation and scheduling of fleet [10]. It means, the control of the supply processes includes external and internal factors to be optimised. Within the frame of this paper authors are focusing on the supply accuracy on the inventories.

As a simplified model, Figure 1 shows the inventory mechanism and the modification of stocks in case of a deterministic demand and stock replenishment. The initial conditions of the model include the continuous use at a steady pace, a steady t periodic supply, the minimum stock level $Q_{\min}$ equal to zero starting from the inadmissible stock shortage and the zero safety stock, and the Q reorder quantity until the admissible $Q_{\max}$ maximum stock level. The calculable τ replenishment time and the t_i date of receipt determine the latest t_r date, when the order must be placed [11]. The Q_r stock level assigned to time t_r represents the minimum stock level that covers the usage within the time period necessary from the placement of the order until the actual receipt of the ordered quantity [12].

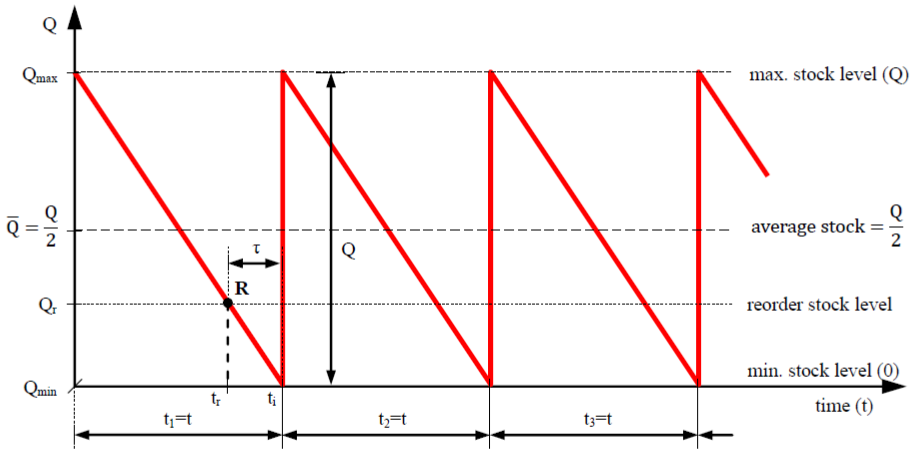

Figure 1. Development of stock level in case of deterministic demand and replenishment time

Furthermore, the initial conditions of the model include the fact that the time demand of stock replenishment is shorter than the periodic time ($\tau < t$), meaning that maximum one supply can be in progress at a given moment. This simplified model shows that the maximum stock level is equal to the quantity per order ($Q_{\text{max}} = Q$), the stock drops to zero at the end of the period ($Q_{\text{min}} = 0$), thus the average stock can be defined with the following formula [13]:

$$\overline{Q} = \frac{Q_{\text{max}} - Q_{\text{min}}}{2} = \frac{Q}{2} \tag{2.1}$$

where

$\overline{Q}$ – average stock,
Q_{max} – maximum stock level,
Q_{min} – minimum stock level,
Q – ordered quantity.

3. The relationship between the stochastic supply accuracy and the stock level

By using the principle of ceteris paribus, we carry out the analysis by emphasizing one initial condition of the model, the development of the arrival time of purchased parts. In this case, the model changes to the extent that we allow a deviation in any direction and of any extent between the lengths of the different periods, by specifying that the demand of the respective period is continuous and constant within the period (almost stationary) [11]. The practical applicability of the model justifies this kind of modification of the conditions, since it is hard to imagine for the operation of producing organizations to respect the delivery deadline with complete accuracy at every moment. This change requires the actual supply of the planned demand, thus the model reflects only the replenishment period and not the quantitative deviations from the planned demand.

In case a disturbance occurs in the process of at one of the supply chain's actors, it can happen that the actual arrival date of goods will deviate from the specified deadline. By continuous use, the delay of the arrival of goods would lead to the shortage of stocks. Figure 2 shows that, compared to Figure 1, in case of a delayed goods arrival the stock level can drop below the previous level $Q_{\min} = 0$.

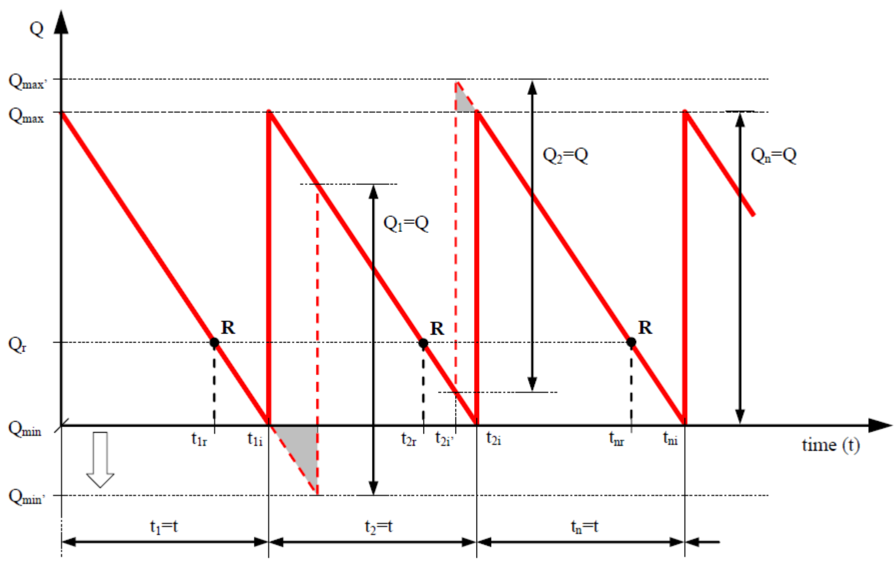

Figure 2. The relationship between supply accuracy of different extents and stock levels

In case of a supply management strategy that does not allow a stock shortage, this means in practice a shift of the Qmin level towards the value $Q_{\min'} = 0$. In order to simplify the model and by applying the principle of ceteris paribus once again, the stock replenishment quality Q remains constant for every order, so it can lead to a maximum stock level $Q_{\max'}$ of different periodical values in relation to the actual date of arrival. The fields marked with grey on Figure 2 show the $Q_{\min}$–$Q_{\min'}$ stock shortage generated by the delayed arrival of goods, and the $Q_{\max'}$–$Q_{\max}$ surplus stock due to early arrival compared to the planned maximum stock level $Q_{\max}$ [14].

The average stock can be defined using two different approaches. With the first method, we divide the whole analyzed period into shorter periods by taking the planned and the actual arrival dates of goods into account. The average stock of the various shorter periods can be quantified separately, and the average stock of the whole analyzed period can be determined by using the average stocks of the individual periods and the length of the different intervals (Figure 3). According to the other method, we take the opening and closing stocks into account conform to the periods with the original length values, and the utilization assumed as uniform during the whole period.

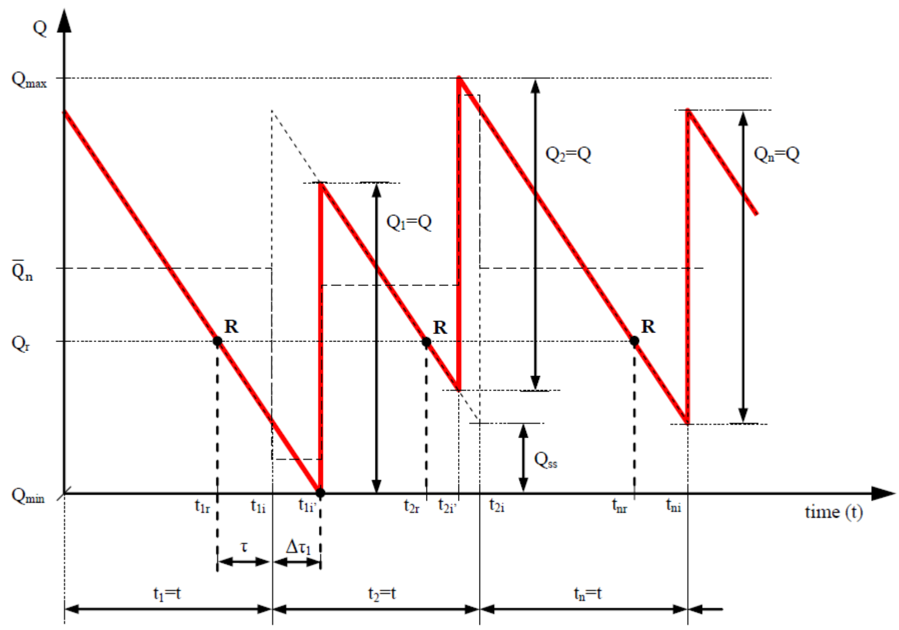

Figure 3. The average stock of the various shorter periods

In practice, the deviation from the planned arrival date can be attributed to various root causes, e.g. unexpected traffic obstacles, disturbances in the supplier's processes, cancellation of supply due to a lack of capacity, delayed or early supply due to an oversight etc.

When analyzing the direction and extent of the deviation of supply accuracy, we can start from the long-term analysis of one product, but we can analyze product groups, product scopes assigned to suppliers, and also complete product portfolios. In case the analysis is limited to the monitoring of one product, the deviations from the target must be analyzed in relation to each other for comparability and cumulability.

In case of an analysis of an adequate number of samples, the deviations will be typically situated along the probability density function of a standard normal unimodal distribution (Figure 4).

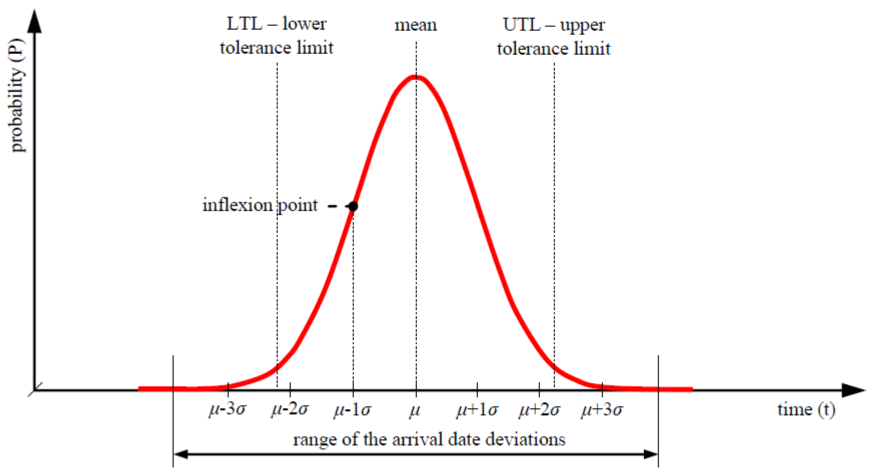

Figure 4. The probability density function of the standard normal distribution and its features

In case the partners have agreed on a maximum number of allowed deviations, the respective range of accepted deviations can be defined in the form of lower and upper tolerance values. The well-known distribution and density functions popular in production management and quality management circles for the presentation of quantity features of certain products can be used in logistics as well. With them, we can show the standard deviation of the actual arrival time from the plan, the situation of deviations compared to the target, the distribution of deviations, the intervention points or tolerances, and the range of the changing replenishment period.

Since the processes between the order placement and the actual arrival of goods are influenced by accidental factors too, the actual date of arrival can be defined only with random variables.

4. The characteristic of the deterministic stock replenishment time

The simplified stock management model showed on Figure 1 and characterized with a deterministic stock replenishment time and a utilization with constant intensity can be completed with the maximum extent of the deviation of supply accuracy in two directions, so that Figure 5 is modelling the deviations occurring compared to the planned utilization. Since the direction and extent of the deviation compared to the plan is stochastic, i.e. it cannot be exactly defined in advance, thus the development of the actual arrival date and of stocks can be determined only with random variables that are represented on the Figure 5 by the probability density functions of the standard normal distribution.

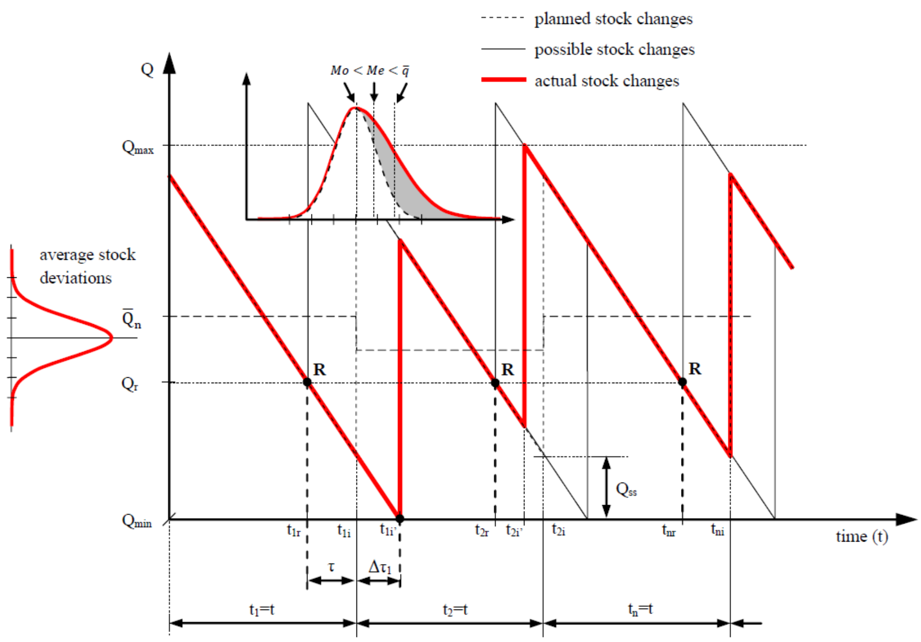

Figure 5. Continuous review in case of stochastic replenishment period

The arrival date of the ordered product is theoretically limited from below, meaning that the product ordered at the R date cannot arrive sooner than the R date of the order placement, and in general, a delay of any extent can occur in the actual arrival date by varying degrees of probability. The form of frequency curves limited from below and characterized with a left-side asymmetry resembles the density function of the F-distribution and the χ^2

distribution. However, in practice – even if by a very low probability – it can occur that the item ordered at the R date arrives before the order placement due to some kind of error, e.g. in case of a supply of identical products in every period, the supplier sends a certain amount even if an order was not placed in the respective period. This means that the date of arrivals is not limited from below in practice, a fact that justifies the application of the probability density function of a standard normal distribution.

The situation and form of the density function shows these two approaches in the Figure 5. The density function reaches a little bit over order rate R from the left side, showing the low probability of the item arriving before the order placement. The left-side asymmetry of the density function marked with grey shows the distortion of the probability of the delay and the early arrival towards the delays.

The longer the stock replenishment time τ, and the higher the performance accuracy towards the supplier (service level), the smaller the shift or asymmetry of the probability density function of standard normal distribution in any direction. Accordingly, in order to simplify the model, we will use a symmetric unimodal function without the shift of the mean value.

In case we only keep stocks to cover the demands of the planned length of the period, according to our stock management strategy, the delayed receiving of the purchased parts compared to the planned $t_{i'}$ arrival date would result in a drop to zero of the stocks before the ti date of receipt. This is inacceptable in case we start from a strategy that does not allow a stock shortage, a certain amount of safety stock (Q_{ss}) should be kept in order to avoid the risks and costs due to delayed deliveries.

Since the fluctuation of supply accuracy has a stochastic character, the possible results are random variables equivalent to the standard normal distribution. Figure 5 shows that if the date of arrival is a random variable, the occurrence of extreme deviations has a very low probability, thus the stock shortage can be most likely avoided only with a safety stock with an infinite high level. In practice however, this is not feasible due to different economic efficiency reasons, thus a compromise must be reached between the consequences caused by the stock shortage and the sacrifices brought to avoid the shortage [15].

In order to be able to define the relation between the supply accuracy and the necessary safety stock, we need to convert the delivery time to used quantity. Earlier, we measured the supply accuracy in time and not in quantity, so we have to quantify the quantity used within the period between the planned stock replenishment time and the deviation compared to the deadline. By assuming a use of uniform D intensity, the used amount due within the planned

τ stock replenishment:

$$Q_t = D \cdot \tau \tag{4.1}$$

where

Q_t – utilization during the stock replenishment time,
D – intensity of the demand,
τ – lead time of stock replenishment.

By assuming a use of uniform D intensity, the used amount due within the fluctuation by $\Delta\tau$ of the stock replenishment time:

$$Q_{\Delta\tau} = D \cdot \Delta\tau \tag{4.2}$$

where

$Q_{\Delta\tau}$ – fluctuation of the utilization caused by the fluctuating stock replenishment time,
$\Delta\tau$ – fluctuation of the stock replenishment time.

5. The effect of the safety stock level on the probability of the stock coverage and shortage

The density function detailed on Figure 6 shows the probability of the used amount due within the actual stock replenishment period, which reflects that the frequency of occurrences is more typical around the expected value, while their probability decreases towards the two ends of the function [16, 17].

The Figure 6 shows clearly that the extent of the given stock level covering the demand during the replenishment period can be defined in relation to any arbitrary q_n stock level and the actual demand during the replenishment period. The stock level above the actual demand during the replenishment period is the safety that covers the fluctuation of the arrival date of purchased parts. The safety stock level can be calculated with the following formula:

$$Q_{ss} = q_n - \mu \tag{5.1}$$

where

Q_{ss} – safety stock level,
q_n – arbitrary stock level,
μ – the actual demand or the expected demand value during the replenishment period.

The formula above demonstrates that the safety stock can have a negative level, if we allow the probability of the occurrence of a stock shortage during our stock management strategy to reach an extent where the stock we

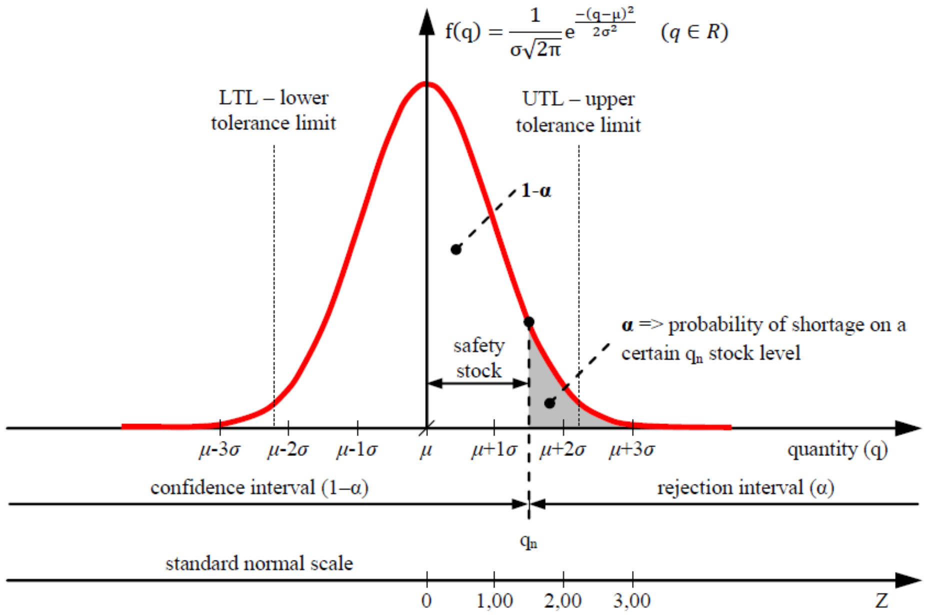

Figure 6. The effect of the safety stock on the occurrence probability of the stock shortage

keep in certain periods is not enough to cover the planned demands of the replenishment period:

$$q_n < \mu \text{ in case of } Q_{ss} < 0 \tag{5.2}$$

In the knowledge of the standard deviation and the supply accuracy, the probability of the occurrence of a shortage can be defined in relation to any replenishment period and stock level. In case we adjust the stocks only to the planned demand level, i.e. we do not keep a safety stock ($q_n = \mu$), the probability of the occurrence of a shortage in case of a supply accuracy with standard normal distribution will correspond to the probability of the non-occurrence of the shortage (Figure 7). Since the combined probability of the two possible results is 100%, the following relation is valid:

$$P_{(\text{shortage})} = P_{(\text{coverage})} = 0.5 \tag{5.3}$$

where

$P_{(\text{shortage})}$ – probability of stock shortage,
$P_{(\text{coverage})}$ – probability of stock coverage.

The distribution function on Figure 7 shows the probability of the coverage of replenishment period fluctuations in relation to the changes of the stock level. The probability of the occurrence of the shortage decreases proportionally with the increase of the safety stocks; however, the complete safety can be guaranteed only by a stock with an infinite level. The introduction of the distribution function is justified by the fact that the function reflects well the probability of the occurrence of the stock shortage in connection with the stock level. The probability can be expressed as a risk as well, to which actual costs can be assigned.

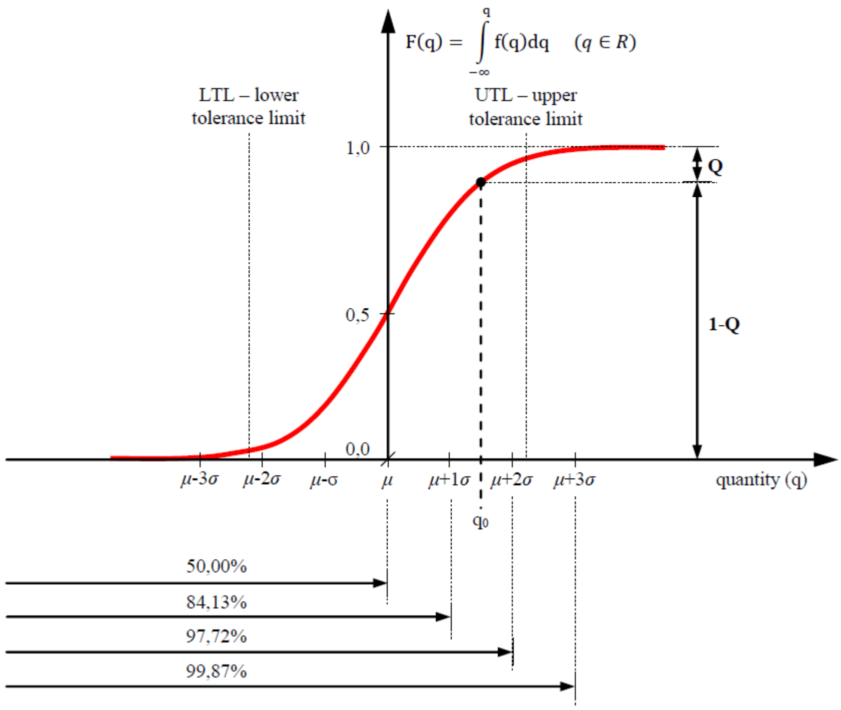

Figure 7. The probability of the coverage of supply accuracy fluctuations in relation to the change of the stock level

6. The method of the safety stock calculation

When determining the order quantity due for the next period and the stock level guaranteeing a continuous operation, five tasks must be completed.

1. The service level of our stock management strategy has to be defined, i.e. the frequency of shortage acceptance.

2. Define the planned duration of the stock replenishment.
3. Define the expected duration of the acceptable stock replenishment by the respective service level by considering the expected fluctuation of the stock replenishment time.
4. Define the stock level that covers the utilization demand due within the expected duration of the acceptable stock replenishment at the respective service level by avoiding the occurrence of a shortage.
5. Verify whether the stock available at the time of the order does cover the utilization demand defined by considering the service level and the expected fluctuation of the stock replenishment time.

Under service level we mean the extent of acceptance regarding the shortage, which can be defined in two ways [1]. It can be determined as a relation of the number of periods allowing shortages and the total number of periods analyzed, or we can define the allowed probability of the occurrence of the shortage. In case we allow 2 periods uncovered by stocks for a complete period, e.g. one year, the service level can be defined by a weekly period as follows:

$$\frac{SL \cdot t}{T} = \frac{2 \cdot 7}{360} = 0.0389 \tag{6.1}$$

where

SL – service level,
t – the length of a given period,
T – the length of the complete period.

From the standard normal distribution table we get the value $Z = 1.765$ matching the probability value 0.0389. Another way to define the service level is to define the maximum allowed level of the probability of the shortage's occurrence (e.g. 0.02) and to find the matching $Z = 2.33$ value. Irrespective of the definition of the service level, we apply the resulting value in a consistent way.

Subsequently, we must define the planned duration of the stock replenishment of the n-th period:

$$\tau_n = t_{ni} - t_{nr} \tag{6.2}$$

where

τ_n – the planned duration of the stock replenishment of the n-th period,
t_{ni} – the planned arrival date of the order belonging to the respective n-th period,
t_{nr} – the date of the order placement belonging to the respective n-th period.

The fluctuation expected for the duration of the next period can be defined based on the service level and the expected demand value during the replenishment period. By using the formula the distribution function (6.3), we can determine the $F(q)$ value of the distribution function for any arbitrary q value [18, 19, 20]:

$$F(q) = \int_{-\infty}^{q} f(q)\,\mathrm{d}q = \frac{1}{\sigma\sqrt{2\pi}} \int_{-\infty}^{q} e^{\dfrac{-(q-\mu)^2}{2\sigma^2}}\,\mathrm{d}q = \Phi\left(\frac{q-\mu}{\sigma}\right) = Z \tag{6.3}$$

where

$F(q)$ – the formula of the distribution function,
Z – the standardized value of a normally distributed random variable.

In case the value of the standard normal distribution function is equal to the service level, the stock level q barely covers the expected utilization demand μ derived from the replenishment period and constant utilization that can be characterized with an σ standard deviation by an SL reliability level:

$$\Phi\left(\frac{q-\mu}{\sigma}\right) = SL = Z \tag{6.4}$$

where

q – the demand during the stock replenishment time expired until the actual arrival of goods,
σ – standard deviation of the demand, caused by the fluctuation of the length of the stock replenishment period.

We have to define the expected duration of the acceptable stock replenishment by the respective service level by considering the expected fluctuation of the stock replenishment time. We can transform the above-mentioned formula in a way that q equals the demand during the $t_{ni'}$ stock replenishment time expired until the actual arrival of goods accepted by the given SL level, and μ equals the demand during the $\tau_n = t_{ni} - t_{nr}$ planned stock replenishment period, it means:

$$q = D \cdot (t_{ni'} - t_{nr}) \text{ and } \mu = D \cdot (t_{ni} - t_{nr}) = D \cdot \tau \tag{6.5}$$

where

$t_{ni'}$ – stock replenishment time expired until the actual arrival of goods.

From the formula above it follows:

$$F(q) = \Phi\left(\frac{q-\mu}{\sigma}\right) \to \Phi\left(\frac{D \cdot (t_{ni'} - t_{nr})}{\sigma}\right) = \Phi\left(\frac{D \cdot \Delta\tau_n}{\sigma}\right) = SL \tag{6.6}$$

Formula (6.5) and (6.6) show that the Q_{ss} safety stock level belonging to the SL service level is the difference of the stock level and during the planned replenishment period expected demand level $q - \mu$, which can be defined with the help of the standard normal distribution $\Phi(q)$ table, it means:

$$Q_{ss} = q - \mu \tag{6.7}$$

The extent of the expected demand due by the end of the period between the ordering point and the receipt of products (replenishment period) can be quantified similarly to formula (6.5):

$$\mu_\tau = \tau \cdot \overline{D}_\tau \tag{6.8}$$

where

> μ_τ – expected value of demand due within the stock replenishment period,
> $\overline{D}_\tau$ – average demand of the period between the ordering point and the period's end.

During the quantification of the safety stocks, it is not enough to cover the utilization demands expected for the remaining τ period of the ordering period; the temporal deviations during stock replenishment must be taken into account as well. Figure 8 demonstrates the above-mentioned relations [1].

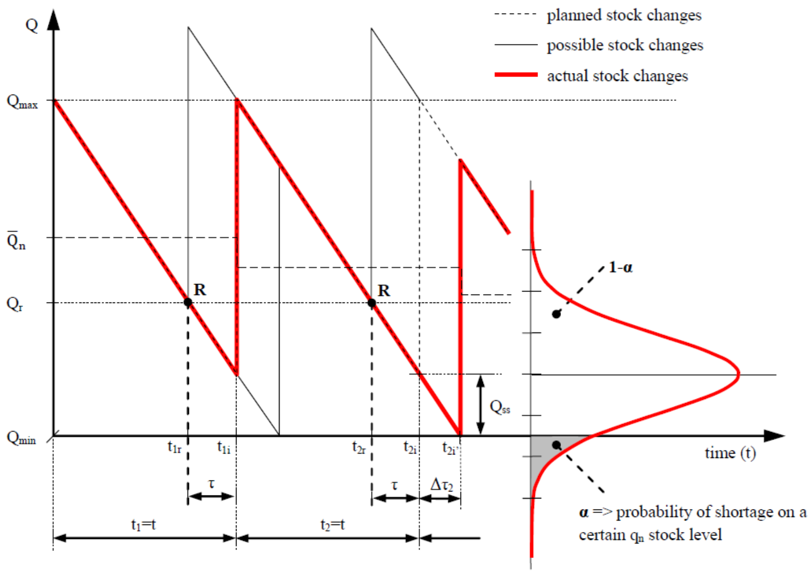

Figure 8. The relations of the continuous review

7. Summary

Several unpredictable factors can occur in the supply chain, that influencing the operation of the production company. The objective of the logistics management is to guarantee the stock level required for the adequate handling of production at the lowest possible level of costs. The more stringent expectation we define towards the frequency of stock shortages in our stock management strategy, the higher the expenses will be for the avoidance of the stock shortage. Fundamentally, we can guarantee the achievement of the target level in two directions:

- with a stricter supplier qualification, we can reduce the uncertainty of supplies,
- we can increase the safety stocks to cover the fluctuating arrival date of purchased parts.

The management of unpredictable supply accuracy urges the management of the production company to reach a compromise. The shortage of stocks can cause serious disturbances in the production supply and customer service, the costs of which are often unquantifiable. The avoidance of stock shortages is an important objective for every organization; however, it would be possible only with the management of an infinite stock level due to the stochastic nature of demands.

Acknowledgements

This research was partially carried out in the framework of the Centre of Excellence of Mechatronics and Logistics at the University of Miskolc.

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] KOLTAI T.: *Production Management* (in Hungarian: *Termelésmenedzsment*), Published by Typotex Kiadó, Budapest, 2009, ISBN 978 963 279 035 0, ISSN 1787-9655

[2] WANG, Y, WALLACE, S. W., SHEN, B, CHOI, T-M.: *Service supply chain management: A review of operational models.* European Journal of Operational Research, Volume 247, Issue 3, 16 December 2015, pp. 685-698

[3] MASOUD, S. A., MASON, S. J.: *Integrated cost optimization in a two-stage, automotive supply chain.* Computers & Operations Research, Volume 67, March 2016, pp. 1-11.

[4] Cardenas-Barron, L. E., Gonzalez-Velarde, J. L., Trevino-Garza, G.: *A new approach to solve the multi-product multi-period inventory lot sizing with supplier selection problem.* Computers & Operations Research, Volume 64, December 2015, pp. 225-232.

[5] Zhou, Y.-W., Wen, Z.-L., Wu, X.: *A single-period inventory and payment model with partial trade credit.* Computers & Industrial Engineering, Volume 90, December 2015, pp. 132-145.

[6] Lee, J.-Y., Cho, R. K., Paik, S. K.: *Supply chain coordination in vendor-managed inventory systems with stockout-cost sharing under limited storage capacity.* European Journal of Operational Research, Volume 248, Issue 1, 1 January 2016, pp. 95-106.

[7] Zahran, S. K., Jaber, M. Y., Zanoni, S., Zavanella, L. E.: *Payment schemes for a two-level consignment stock supply chain system.* Computers & Industrial Engineering, Volume 87, September 2015, pp. 491-505.

[8] Cedillo-Campos, M. G., Cedillo-Campos, H. O.: *Security risk level classification of stock keeping units in a warehouse.* Safety Science, Volume 79, November 2015, pp. 358-368.

[9] Dotoli, M., Epicoco, N., Falagario, M., Costantino, N., Turchiano, B.: *An integrated approach for warehouse analysis and optimization: A case study.* Computers in Industry, Volume 70, June 2015, pp. 56-69.

[10] Sha, M., Srinivasan, R.: *Fleet sizing in chemical supply chains using agent-based simulation.* Computers & Chemical Engineering, Volume 84, 4 January 2016, pp. 180-198.

[11] Cselényi J., Illés B.: *Planning and controlling of material flow systems* (in Hungarian: *Anyagáramlási rendszerek tervezése és irányítása*), Published by Miskolci Egyetemi Kiadó, Miskolc, 2006, ISBN 963 661 672 8

[12] Prezenszki J.: *Warehousing-logistics* (in Hungarian: *Raktározás-logisztika*), Published by Ameropa Kiadó, Budapest, 2010, ISBN 978-963-06-8670-9

[13] Szegedi Z., Prezenszki J.: *Logistics Management* (in Hungarian: *Logisztika-menedzsment*), Published by Kossuth Kiadó, Budapest, 2003, ISBN 963 09 4434 0

[14] Prezenszki J. (ed.): *Logistics I.* (in Hungarian: *Logisztika I.*), Published by Kossuth Kiadó, Budapest, 2006, ISBN 963 431 796 0, ISSN 0865-3313

[15] Vörös J.: *Production and service management* (in Hungarian: *Termelés- és szolgáltatásmenedzsment*), Published by Akadémia Kiadó, Budapest, 2010, ISBN 978 963 05 8835 5, ISSN 2061-6430

[16] Kulcsár B.: *Industrial logistics* (in Hungarian: *Ipari logisztika*), LSI Oktatóközpont, Budapest, 1998, ISBN 963 577 242 4

[17] Kummer, S., Grün, O., Jammernegg, W.: *Basics of purchasing, production, logistics (in German: Grundzüge der Beschaffung, Produktion, Logistik)*, Published by Pearson Studium, München, 2009, ISBN 978-3-8273-7351-9

[18] CSERNYÁK L. (ED.): *Probability theory – mathematics for business administration* (in Hungarian: *Valószínűségszámítás – matematika üzemgazdászoknak*), Published by Nemzeti Tankönyvkiadó, Budapest, 1998, ISBN 963 18 8413 9

[19] CSERNYÁK L. (ED.): *Analysis – mathematics for business administration* (in Hungarian: *Analízis – matematika üzemgazdászoknak*), Published by Nemzeti Tankönyvkiadó, Budapest, 1998, ISBN 963 19 0307 9

[20] KORPÁS A.-NÉ (ED.): *Elementary statistics II.* (in Hungarian: *Általános statisztika II.*), Published by Nemzeti Tankönyvkiadó, Budapest, 1997, ISBN 963 19 5508 7

Production Systems and Information Engineering
Volume 7 (2015), pp. 33–42

ROLE OF WORD REPRESENTATION FORMAT IN MORPHOLOGY LEARNING

László Kovács
University of Miskolc, Hungary
Department of Information Engineering
kovacs@iit.uni-miskolc.hu

Daniel Hládek
Technical University of Kosice, Slovak Republic
Faculty of Electrical Engineering an Informatics
daniel.hladek@tuke.sk

[Received September 2015 and accepted November 2015]

Abstract. This paper provides a survey on the available methods for induction of morphology grammars using statistical methods. The selected word and character representation forms play a crutial role in the efficiency of the generated grammar rule system. The performed analysis covers a wide range of methods where the main focus is set on the decision tree based rule trees which provides an efficient approach for large problems too. Regarding the word representation formats, the paper shows the main benefits of the attribute-based approach.

Keywords: morphology analysis, grammar induction, word representation formats

1. Morphology

The morphology is a linguistic discipline analyzing the internal components of words in expression of syntactical roles. In the case of syntetic languages, the morphemes are the primary tools to determine the meaning of the words. The most widely used implementation forms of morpheme units are the affixes. In concatenative morphology, the different components (morphemes) can be composed into a sequence and there is a clear boundary between the different components. On the other hand, some languages use also non-concatenative morphology rule too. The past tense of irregular verbs in English language is based on this kind of transformation. In the Ngiti [4] language, the plural of a noun is generated by replacing the last two syllables with high tone syllables:

singular: kama, plural: km.

Current morphology analyzer, like the Humor analyser for the Hungarian language [1] uses a lexicon on the possible surface forms. The implemented program performs a search on the input word form for possible analyses. It looks up morphs in the lexicon the surface form of which matches the beginning of the input word (and later the beginning of the yet unanalyzed part of it). The lexicon may contain not only single morphs but also morph sequences. These are ready-made analyses for irregular forms of stems or suffix sequences, which can thus be identified by the analyzer in a single step, which makes its operation more efficient [7, 8].

A transformed word has usually a stem part which relates to the base lexeme word. The inflectional part denotes the modified parts of the word. Considering inflectional part, it can be subcategorized based on the position of the modification: prefix, suffix and infix part. The prefix part occurs at the beginning of the word, suffix is at the end of the word and infix part is within the stem word. The circumfix denotes the case where both prefix and suffix parts are used. Based on the degree of morphology transformations in the words, there are isolating and polysynthetic languages. The isolating languages do not use any morphological transformation, while polysynthetic languages allow considerable number of inflectional morphemes for a lexeme word. Another categorization system is based on the meaning of the affix parts. In case of fusional languages, the same morpheme unit can express many different semantic roles at the same time. For example, the morpheme s in the word walks denotes the following properties: present tense, singular and 3rd person. The other group of languages is called agglomerative languages as they use a unique morpheme for every semantic role. These morpheme units are usually well separable from each other.
Due to the big variety and irregularity of morphology, it is a big challenge in computational linguistic to develop an efficient learning method for induction of morphology rules. In the literature, current tools are primary tailored only on the simple concatenative morphology using no fusional components. The TASR method [6] for example builds up a tree of suffix rules based on the training samples. The method provides efficient execution cost characteristics but it has been proven inefficient in generalization of the rules for untrained data.

2. Learning methods

2.1. TASR - a decision tree method

A widely used tree-based method is the TASR method [6] which is based on a tree of suffix rules. A suffix rule is defined with a rule $LHS \rightarrow RHS$, where LHS is the left-hand suffix and RHS is the right-hand suffix. This means that LHS from the input will be transformed to RHS in the output. For example, the training word pair alma → almt indicates the following suffix rules:

- alma → almt
- lma → lmt
- ma → mt
- a → t

A dependency can be observed among these rules based on the alignment relationship. The $LHS \rightarrow RHS$ suffix rule is aligned with the $LHS' \rightarrow RHS'$ suffix rule iff $LW = con(s, LHS) = con(s', LHS')$ and $RW = con(s, RHS) = con(s', RHS')$ where symbol *con* means string concatenation and s, s' are prefixes and LW, RW are arbitrary words.

The child suffix rule $LHS \rightarrow RHS$ is subsumed by the parent suffix rule $LHS' \rightarrow RHS'$ iff $\exists x \in \Sigma : (LHS = con\,(x, LHS') \ \wedge RHS = con\,(x, RHS'))$.

The suffix rule tree based on the alignment relationship of the left hand suffixes, meets the following properties:

- The root's $|LHS|$ is minimal.
- Each node contains the rule with the highest frequency, where that rule is not subsumed by the parent's rule.
- For every child node, $LHS' = con\,(x, LHS)$ where $x \in \Sigma$ and LHS is the parent rule.

The tree building starts with an empty root node, by extracting all the aligned suffix rules according to the definitions above. For every node we have to find the rule with the highest frequency and check whether or not it is subsumed by the parent rule. In the latter case the found rule is the winning rule and we can append the new node to the tree.

After this recursive building algorithm, we can start using the tree, this time in a bottom-up fashion. We traverse the tree for each input word, and try to find the first matching rule from the deepest level. After we found it, we can apply the transformation on the input word and return the result.

Of course this method depends heavily on the input training set, and we can only use TASR in case of special languages, because only the suffixes are considered by the algorithm. However, if the language meets the requirement of TASR, it is by far the fastest method.

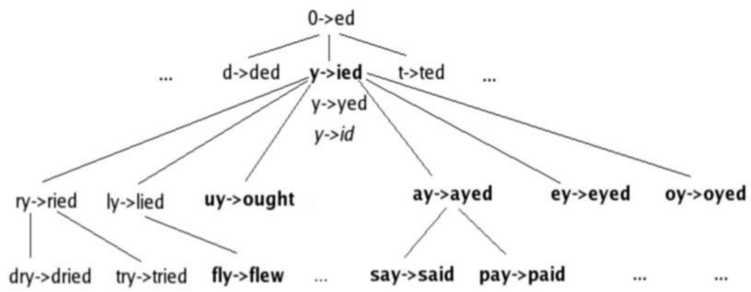

Figure 1. Sample figure (source:[6])

2.2. Ostia - FST method

Another but slightly more complex morphological method is OSTIA or the Onward Subsequential Transducer Inference Algorithm [5] with which belongs to the family of FST (finite state transducer) methods.

A transducer is defined as $t \subseteq \Sigma^* \times \Gamma^*$, $t = \{(s_1, s_2) \mid s_1 \in \Sigma^*, s_2 \in \Gamma^*\}$ where s_1 is the input word and s_2 is the output word.

A rational transducer is a tuple $T = (Q, \Sigma, \Gamma, q_\lambda, E)$ where

- Q is the set of states
- Σ and Γ are the input and output alphabets, respectively
- q_λ is the start state
- $E \subseteq Q \times \Sigma^* \times \Gamma^* \times Q$ is the set of transitions

The sequential transducer is a rational transducer such that $E \subset Q \times \Sigma^* \times \Gamma^* \times Q$ and $\forall (q, a, u, q'), (q, a, v, q'') \in E \rightarrow u = v \wedge q' = q''$

The subsequential transducer is defined as $T = (Q, \Sigma, \Gamma, q_\lambda, E, \sigma)$ such that the first five components make a sequential transducer and $sigma : Q \rightarrow \Gamma^*$ is the state output function.

The onward transducer meets the following condition

$$\forall q \in Q, a \in \Sigma : lcp\left(\left\{u \mid \left(q, a, u, q'\right) \in E\right\} \cup \left\{\sigma\left(q\right)\right\}\right) = \lambda,$$

where *lcp* returns the longest common prefix of the given word set.

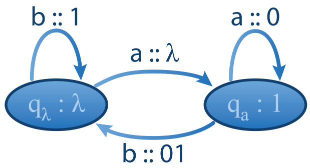

Figure 2. A sample onward subsequential transducer

During production mode, we start from q_λ (start state) and for every input character we change our state and append the output characters on the edge to the output word. After we reach the end of the input word, the end state's output characters are also appended and thus the output word is produced and the algorithm is finished.

There are many applications of the FSA methods in the literature. It can be used among others for learning of word formation too. In the article [2], the formation of noun + noun compounds by computational morphological means is investigated by FST in order to understand how this process should be formalised, modelled and subsequently implemented. In [3], Allomorfessor model is presented using a probabilistic context-free grammar (PCFG). The model extends the morphological segmentation method, Morfes- sor Baseline, to model allomorphy. The proposed unsupervised method discovers common base forms for allomorphs from an unannotated corpus.

2.3. Markov models

In the statistical approaches of NLP problems the Markov model is a dominating method. The Markov processes are based on the assumption that the state value at given time point, depends only on the previous states. In the case of 1-MM, the following assumption is used for every t index:

$$P(s(t) = x|s(t-1)s(t-2)...s(1)) = P(s(t) = x|s(t-1))$$

The main parameters of the Markov model are the

- the probabilities of the initial states
- the probabilities of the state transition

After learning the probability parameters, the probability of the different state sequences can be calculated.

The base Markov model [10] is suitable for problems where the state values can be observed directly. In many problems of morphology and grammars, this condition is not met, as we cann't observe the state, only some related labels are visible. In the case of Hidden Markov Models (HMM), the following components are included into the problem description:

- set of hidden states
- state transition probabilities
- initial state probabilities
- set of output symbol
- symbol transmission probabilities for the states

The goal of the Viterbi algorithm is to determine the hidden state sequence for a given symbol sequence. The algorithm uses a dynamic programming approach, i.e. the global optimum is calculated via sequence of local optimum states. The algorithm determines the most likely state transition path for the given observation sequence O and probability parameters M:

$$argmax_X P(X|O,M)$$

3. Character Representation Formats

In the morphological analysis of the words, the representation form plays a crucial role. The morphology rules are usually based on latent physical and mental parameters and the written form of the words is a result of several transformation steps. The process of morpheme analysis is used in many different application areas having very different process models. In the different environments, different transformation processes can be observed. For example, in NLP interface, the voice to text transformation has a central role, while in a word spelling application, the position of keyboard buttons is an important factor.

3.1. Alphabet representation

This base model is the most widely used model where the alphabetical symbols are applied as building blocks. Most of the proposals like [9] use this approach. This model is the most general one, and it can be used for every application areas. This model is based on the surface level, the processes are working only with the visible symbols. As there are many complex dependencies in the background, the work at the surface level is considered as a more complex problem, it is harder to detect the dependencies. The following table summarizes the benefits and the drawbacks of this model.

Table 1. Alphabet representation

Benefits	**Drawbacks**
- simple representation - general - many application areas - many supporting methods	- complex relationships - not very efficient

3.2. Metric space approach

In a spell checking application, the core measure is the similarity between the words. There exists a lexicon of valid words and the words found in the text should be tested for validity. If the word is not found, the best matching words from the lexicon should be suggested for corrections [11].

The metric space is based on a distance function meeting the following conditions

$$d(a,b) \geq 0,$$
$$d(a,b) = d(b,a),$$
$$d(a,c) \leq d(a,b) + d(b,c).$$

In this problem domain, the word distance metric is used to measure the similarity between the words. The most usual approach is the edit distance, where a dynamic programming algorithm is used to calculate the minimum transformation cost. The most widely used approach is the application of unit costs, i.e unchanged

- costinsert = 1
- costdelet = 1
- costreplace = 1
- costunchanged=0

In the case of special application, like spell check of text written on keyboard, a more specific distance model can provide a better solution. For example, in the case of keyboard, the distance matrix of the letters is based on the position of the letter keys. Using this mode, the implemented spell checker preferences the lexicon words having characters in the neighboring positions.

Table 2. Metric space representation

Benefits	**Drawbacks**
- simple representation - better accuracy for special domains - many supporting methods	- complex relationships - not very efficient - domain specific distances

3.3. Vector representation

For description of the label parameters, the label should be considered as a compound object. This model enables a deeper investigation and a more accurate representation. For example, in the case of NLP applications, the letter is a representations of the tones and the tones have special physical parameters. Thus the letter symbols can be described with the help of these physical symbols. This representation can improve the efficiency of the rule induction processes as it takes also some of the latent parameters behind the visible letters. It can be assumed namely, that there exists some relationship between the physical parameters of the tunes in the word inflection system.

Using a vector space model, the feature vectors are used to represent the letters where the dimensions correspond to the attributes. The main benefit of the vector model, that the usual calculation methods like Euclidean distance or cosine similarity can be used on a standard way. On the other hand, the vector model is a relatively rigid model, as the dimensionality is fixed. That restriction makes some reasoning operations like generalization more difficult to implement.

Table 3. Vector representation

Benefits	**Drawbacks**
- more accurate representation - standard model - more accurate inference - many application areas - many supporting methods	- complex relationships - higher calculation costs - rigid representation

3.4. Attribute set representations

The attribute set representation has a similar goal as the feature vector representation but here, the attributes are considered as elements of a set instead of dimensions. This model also enables a deeper investigation and a more accurate representation. This representation can improve the efficiency of the rule induction processes as it takes also some of the latent parameters behind the visible letters.

This approach is based on an object representation widely used in many information models, namely on the application of a relation context. The main benefit of the set based approach that it enables the application of the standard methods in conceptual modelling. For example, in the FCA (Formal Concept Analysis) model where the objects are described with attribute sets, the generalization can be used on a natural way. During the generalization, the intersection of the corresponding attribute sets yields the description of the generalized concept. Using sets instead of vectors can extend the set of operations in the rule induction methods.

Table 4. Attribute set representations

Benefits	**Drawbacks**
- more accurate representation - standard model - more accurate inference - many application areas - many supporting methods - flexible representation	- complex relationships - higher calculation costs

3.5. Syllable representation

In some proposal on morpheme analysis, a model best matching the speech format is used to represents the words. The syllable is an atom of the speech and this unit can be used in word analysis too. As the syllable units are closer to the physical level than the letter description, it can provide a more accurate analysis of the hidden relationships. To use a syllable based description, the words should be transformed first into a syllable format. This step is on the other hand, not a trivial task.

Table 5. Syllable representations

Benefits	**Drawbacks**
- more accurate inference - direct representation	- complex relationships - higher calculation costs - domain specific - few supporting methods

4. Conclusion

In this paper, a survey on the main statistical methods for morphology rule induction was presented with especial highlight on the role of word representation formats. From the viewpoint of generality and cost efficiency, the methods using decision tree structure and an attribute set representation format provide the best option for implementation.

Acknowledgements

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003. The presented research work was partially supported by the grant TÉT_12_SK-1-2013-0019.

REFERENCES

[1] Prószéky, G. and Novák, A.: *Computational Morphologies for Small Uralic Languages, Inquiries into Words*, Constraints and Contexts. 2005, pp.116-124

[2] Pretorius, L., Viljoen, B., Pretorius, R. and Berg, A.: *Towards a computational morphological analysis of Setswana compounds*, Literator 29(1), 2008, pp.1-20

[3] Kohonen, O., Virpioja, S. and Klami, M.: *Allomorfessor: Towards Unsupervised Morpheme Analysis*, Proc. of Evaluating Systems for Multilingual and Multimodal Information Access, 2009, pp. 975-982

[4] Booij, G. : *Construction Morphology*, Oxford press, 2010

[5] Higuera, C.: *Grammatical Inference Learning Automata and Grammars*, Cambridge University Press, 2010

[6] Shalonova, K., Flach, P.: *Morphology Learning Using Tree of Aligned Suffix Rules*, ICML Workshop on Grammatical Inference, 24th International Conference on Machine Learning, 2007.

[7] Akane, T., Mitsuyuki, S. and Akiko, O,: *Development of a Dictionary for the Morpheme Analysis of Agricultural Information*, Proceedings of WCCA 2008, pp. 449-455.

[8] Frakes, W. and Fox, C.: *Strength and Similarity of Affix Removal Stemming Algorithms*, ACM SIGIR Forum, 37(1), 2003, pp. 26-30.

[9] Tordai, A. and Rijke, M.: *Four Stemmers and a Funeral: Stemming in Hungarian at CLEF*, Accessing Multilingual Information Repositories, NLCS 4022, 2006, 179-186.

[10] Manning, C. and Schtze, H.: *Foundations of Statistical Natural Language Processing*, MIT Press, 2003

[11] Kovacs, L.: *Efficient dictionary matching of character streams*, Proc. of SISY 2010, 2010, pp. 103-107

Production Systems and Information Engineering
Volume 7 (2015), pp. 43–53

UTILIZING APACHE HADOOP IN CLIQUE DETECTION METHODS

László Kovács
József Hatvany Doctoral School for Computer Science and Engineering
University of Miskolc, Hungary
Department of Information Engineering
kovacs@iit.uni-miskolc.hu

Gábor Szabó
József Hatvany Doctoral School for Computer Science and Engineering
University of Miskolc, Hungary
Department of Information Engineering
szgabsz91@gmail.com

[Received August 2015 and accepted November 2015]

Abstract. There are many areas in information technology and mathematics where we have to process large graphs, for example data mining based on social networks, route problems, etc. Many of these areas require us to explore the connections among nodes and find all the maximal cliques in the graphs, i.e. all the node sets whose members are mutually connected with each other. One possible and widely used clique detection method is the so-called Bron-Kerbosch algorithm. However, this technique alone might be too slow for big graphs, thus porting the method into a massively parallel system can reduce the overall runtime. This paper introduces some possibilities and starting points in utilizing the open source Apache Hadoop framework that can help in using the resources of multiple computers. The so-called MapReduce architecture makes it possible to divide and conquer the big task into smaller chunks and eventually solve the problem faster than the equivalent sequential methods.

Keywords: graph algorithms, clique detection, mapreduce architecture, Apache Hadoop, parallel systems

1. Applications of Clique Detection

In the era of Internet of Things or IoT [1] for short, sensors are creating unprecedented amount of new data continuously. The measured values can often be structured into hierarchies or networks. These graphs need to be processed quickly to extract the information out of them, as most of the values are useless in themselves, we need to clean and interpret them, which would be very difficult, often impossible without proper automatization.

However, IoT is only one area that we can apply graph algorithms. A more popular territory of Information Technology where we can work with huge graphs is the databases and services of different social media sites like Facebook, Google+ or Twitter just to mention a few. These sites are excellent examples of world-wide distributed databases that provide almost uninterrupted services 24/7 all around the globe. Facebook Graph API for instance helps us query different parts of their graph structure that contains not only people and their properties, but also places, locations and all the connections among these graph nodes. One of the more popular widely-known area of social network analysis [2] is answering the question *who knows who*, where we can also apply clique detection if we want to find those people who mutually know each other.

A classical mathematical problem, the symmetric travelling salesman problem can also be solved with the help of clique trees [3] which are optimal data structures for storing cliques of a graph. Although this solution requires more advanced mathematical theory as well, clique detection and clique tree building introduces a new method for this well-known problem.

If we want to search for application areas other than IT, we can find many-many methods and models in different disciplines like biochemistry [4]. So we can see in the literature that clique detection is part of many scientific studies, and thus solving it well and quickly is usually crucial. There are, of course, many models that try to achieve the same goal optimally.

One model that we can use deals with neighborhood relationships. The algorithm presented in [5] tries to solve the clique detection problem in the least iteration steps, analyzing the adjacency of the input graph. A more modern approach is given in [6] where genetic programming is used to solve the same problem. As we can expect, genetic programming or any other artificial intelligence method can be utilized in clique detection algorithms, but these tools are often blind or mostly-blind due to the fact that not the specific problem is solved, but the problem is transformed to the domain of the AI model.

One of the classical models is the so-called Bron-Kerbosch algorithm that solves the clique detection problem with classical mathematical tools. This method was first introduced in [7] and since its first appearance in 1972, many other Bron-Kerbosch variations have appeared that tried to amend the original ancestor in different areas or subproblems.

In this paper the original Bron-Kerbosch algorithm will be used to demonstrate how to apply the old concepts in the fairly new area of MapReduce architecture and Apache Hadoop in order to process huge graphs in parallel.

2. The Bron-Kerbosch Algorithm

Let $G = (V, E)$ be an undirected graph of $v \in V$ nodes and $e = (v_i, v_j) \in E$ edges.

Definition 1 (Clique). *$C(G)$ is a clique of n nodes if* $C(G) = \{v_i \in V \mid 1 \leq i \leq n \wedge \forall v_j, v_k \in C(G) : (v_j, v_k) \in E\}$.

Definition 2 (Maximal clique). *$C(G)$ is a maximal clique if $\nexists C_2(G)$ for which $C(G) \subset C_2(G)$ is true.*

The Bron-Kerbosch clique detection algorithm returns all the maximal cliques for the given graph, effectively returning a set of node sets.

The Bron-Kerbosch method is a recursive algorithm that requires three sets as its input:

- P: the set of nodes that have not been processed yet.
- X: the set of nodes that are not part of the currently investigated clique.
- R: the set of nodes that are all members of the currently investigated clique.

The first step of the algorithm is $\text{BronKerbosch}(V(G), \{\}, \{\})$, thus providing the whole graph as the set of yet-to-be-processed nodes.

After every recursive call we check if the current clique is a maximal clique, and if so, we return it. Otherwise we process every member of P and call ourselves recursively, slightly modifying the input sets according to the neighbors of the currently processed node.

Let $N(v)$ be the neighbors of an arbitrary node $v \in V(G)$. With these notations, a simple pseudocode of the Bron-Kerbosch method looks like listing 1.

Listing 1. The Bron-Kerbosch algorithm

```
BronKerbosch(P, X, R):
        if length(P) = length(X) = 0:
                yield R

        for v in P:
                P2 = intersect(P, N(v))
                X2 = intersect(X, N(v))
                R2 = union(R, {v})
                BronKerbosch(P2, X2, R2)
                P = P \ {v}
                X = union(X, {v})
```

As noted above, there are multiple variations on the Bron-Kerbosch algorithm that try to mend it in different ways. Since the method is recursive, by eliminating some of the recursive calls we can make the algorithm faster.

Another way of speeding up the algorithm in case of large graphs is applying parallel programming. In the next sections some of the existing approaches are introduced, then some thoughts on utilizing Apache Hadoop in clique detection using the original Bron-Kerbosch algorithm.

3. Parallel Programming Approaches

In the history of computer science multiple different approaches have come to life in the need of executing code in parallel. The three most popular areas of them nowadays are native threading, GPGPU and distributed systems.

3.1. Native Threading

One of the first methods for parallel execution that everybody considers is native threading. All the currently used desktop operating systems and modern universal programming languages provide a way to fire up a separate thread and load some of the work to that one. Classical example is the mighty POSIX threading [8], but recently C++11 got its own threading types as well, moreover modern languages like Java and C# provide higher level APIs to work with threads like Java's Fork/Join framework and parallel stream API.

However, there are some commonly known drawbacks of using native threading: first of all, the number of parallel threads cannot be higher than the number of logical CPU cores in our system. That's why we need to take care of this fact, because starting more threads can actually slow down our application if we're not cautious. Another issue is that the number of CPU cores is rather low even in the more modern computers, so any threads started above this number will be executed in the old sequential order by the operating system.

3.2. GPGPU

To solve the second issue of native threading, we can utilize graphical processors in our computers to solve general problems. This is called GPGPU [9] which can be used in general purpose applications but is currently applied more widely in computer graphics and games.

One problem with GPGPU is that there are two major uncompatible approaches as well: CUDA [10] which is developed by NVidia and provides a higher level interface to work with the GPU, and OpenCL [11] which is slightly lower level, but platform independent, meaning that it works on both NVidia and AMD graphics cards.

However, we must make sure that the integration of GPGPU in our application is well suited for these APIs, because implementing some algorithms directly on the CPU and some of them on the GPU won't achieve the expected results, as copying data back and forth between the two memory spaces causes a lot of overhead.

3.3. Distributed Systems

The most widely used approach for parallel programming in data mining algorithms – such as our clique detection sample – is using distributed systems, where the parallelism doesn't occur inside a computer, but instead across multiple computers.

With this approach, we eliminate the current hardware limits and build a virtual supercomputer that contains the combined power of the member machines. Of course the delay and overhead cannot be eliminated, but this is not the goal with such systems. The purpose of distribution is to build an elastic and scalable parallel network so that we can add or remove computers anytime we want.

Although distributed systems have their own history and subcategories, we only deal with the MapReduce architecture and Apache Hadoop in this paper.

4. Apache Hadoop

The main goal of MapReduce [12] and Apache Hadoop [13] is to split the input data to multiple nodes and process them locally. Therefore the distributed system contains mutliple mapper and reducer tasks that can be located on different machines and get input from the core framework.

A mapper task's goal is to produce key-value pairs, grouping the input into multiple subcategories. Each key gets processed by the same reducer task that tries to return an aggregated return value for that key. Data flows and data persistence is controlled by Hadoop that uses the distributed Hadoop File System, or shortly HDFS, that is an open source variant of GFS (see e.g. [14]). HDFS is responsible not only for distributing data to the nodes, but also persisting each chunk on multiple nodes to defend against node failures and data loss. Figure 1 illustrates the logical components and layers of Apache Hadoop 1 and HDFS. Apache Hadoop 2 is slightly different after the introduction of the Yarn framework, but the core concepts have remained the same, only the configuration and flexibility have been improved.

As we can see, Hadoop works in a master-slave fashion where a core component on master controls the slave components. Of course the complexity of the distributed nature and the necessary redundancy is not visible in the rather simplified figure.

5. Using MapReduce to Solve the Classical Bron-Kerbosch Algorithm

When porting a classical algorithm like the Bron-Kerbosch method to the MapReduce architecture, we must solve the following issues:

1. How to split the input data among the nodes?
2. How to test our approach without much overhead?

The next two subsections will answer the above two questions.

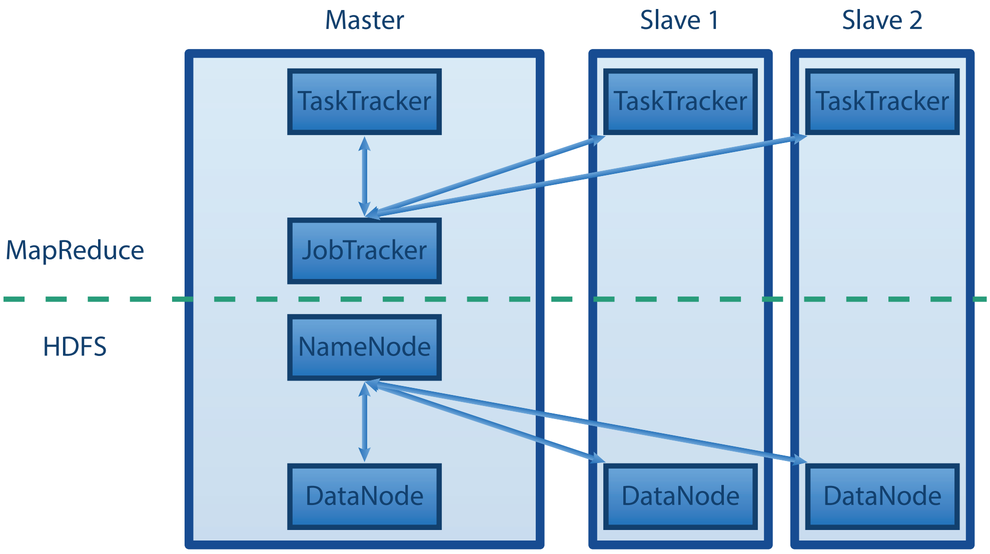

Figure 1. Architecture of Apache Hadoop 1

5.1. Splitting the Input Among the Nodes

When we work directly with Apache Hadoop, we must implement different interfaces that will be called by the framework during execution. These main interfaces are:

- *WritableComparable*: all domain data model classes must implement this interface if they need to be copied among the nodes.
- *Mapper*: this class must be extended by our custom mapper.
- *Reducer*: this class must be extended by our custom reducer.

Every mapper and reducer has four generic type arguments as well: the type of received keys, the type of received values, the type of produced keys and the type of produced values. We have chosen a very intuitive type argument subdivision, that can be seen in table 1.

Table 1. Generic Type Arguments

	Input		**Output**	
	key	**value**	**key**	**value**
Mappers	Null	Graph	Integer	Clique
Reducers	Integer	Clique	Null	Clique

The reason behind this is that the mappers will receive only a graph, it doesn't need to have a key at all. After processing the input graph, they will produce key-value pairs consisting of the hashcode of the resulting clique and the clique itself. The reducer will then receive a clique hash code and the list of every occurrence of the same clique. Since these objects will be equal, the only logic in the reducer is that it returns the first occurrence without a key.

Only one question remains: how to split the big input graph into smaller chunks? Our method was to iterate over the nodes of the graph and extract a smaller graph consisting of the selected node and all of its neighbors. This way we got as many small graphs as there are nodes in the big graph. It's easy to see that if such a small graph contains a maximal clique, it will be a maximal clique in the original graph, too.

Of course there are other approaches for divide and conquer in this example as well. For instance, we could pass the whole graph to every mapper and they would select the appropriate cliques by node index. However, in case of big data this method proved to be slower because the initialization of such distributed systems tend to be slower, since the whole graph has to be copied to every single mapper node, and the mappers must wait until every copy is ready locally.

5.2. Test Environment for Testing Hadoop Based Solutions

When testing a distributed system, usually we have to differentiate a development and a production mode, as maintaining multiple machines for testing purposes is not very remunerative. In our case we had three possibilities beside using a physical distributed system.

The first, and probably the most sophisticated method – that can also be used for production systems – is using a 3rd party cloud service. There are many different providers like Microsoft, Amazon, Oracle, etc. who provide easy-to-use, configurable and scalable solutions. We can pay for any number of virtual machines that we can maintain and use as we wish. However, proof of concepts tend to be more like experimentations that usually take longer time than the available trial of these systems.

Another cheaper method is using local virtual machines. VirtualBox is a free tool that can help us in maintaining arbitrary Linux boxes that we can use for our test distributed system. Usually one physical machine with enough CPU and memory resources can execute even three or four virtual machines that are more than enough for testing purposes. However, writing automated scripts for VirtualBox is not a very easy task, while maintaining them by hand is a bit error-prone.

A perfect match for our purpose is a relatively new tool called Docker, that also helps us to maintain virtual Linux boxes, however the method of virtualization is faster and more configurable. Moreover, Docker provides a way to write automated scripts that can download, initialize, install and configure systems. Our script uses two custom virtual Debian Linux containers and consists of the following steps:

1. It downloads the base Debian Linux container.
2. It installs all the required software for the machine like Java, Maven, Hadoop, ssh, etc.
3. It copies the required Hadoop configuration files to the appropriate folders based on external XML templates.
4. It builds the test application with Maven.
5. It executes the test application and shuts down the machines.

To keep DRY (don't repeat yourself) principles, two custom Linux Docker containers were used: the first one inherits from the base Debian container and configures a base Hadoop slave machine, while the other one inherits the first one and adds support for Hadoop master nodes. This way the installation steps of Java, Maven, Hadoop, etc. are only stored once.

6. Summary

Hadoop provides an excellent way to make classical algorithms run faster by making them parallel. It provides a way to run parts of the algorithm on different computers, thus eliminating the limits of a single machine. The only thing we must be prepared for, is how we want to divide and conquer the input data. One classical graph algorithm is the Bron-Kerbosch method that returns all the maximal cliques of a graph. The original algorithm is sequential, so we decided to port it to Hadoop by splitting the graph by its nodes. The mappers receive a subgraph as input and return key-value pairs of the hashcode of the found cliques and the maximal cliques themselves. After that the reducers return the first occurrence of the mapper outputs because every reducer receives all the mapper output with the same key (hashcode). To test our proof of concept, we used randomly generated input graphs and Docker as the virtualization technology because of its scalable and configurable nature. The next step will be to use real physical machines bound in a real cluster and real-life input data, so that we can properly test not just the validity, but the performance of this algorithm compared to the original sequential method.

Acknowledgements

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] DaCosta, Francis: *Rethinking the Internet of Things: a scalable approach to connecting everything*, 2013, Apress

[2] Scott, John: *Social network analysis*, 2012, Sage

[3] Grötschel, Martin and Pulleyblank, WR: *Clique tree inequalities and the symmetric travelling salesman problem*, Mathematics of operations research, 11, No 4, 1986, pp. 537–569

[4] Butenko, Sergiy and Wilhelm, Wilbert E: *Clique-detection models in computational biochemistry and genomics*, European Journal of Operational Research, 173, No 1, 2006, pp. 1–17

[5] Osteen, Robert E and Tou, Julius T: *A clique-detection algorithm based on neighborhoods in graphs*, International Journal of Computer & Information Sciences, Vol 2, No 4, 1973, pp. 257–268

[6] Haynes, Thomas and Schoenefeld, Dale A: *Clique detection via genetic programming*, 1995, Citeseer

[7] Bron, C and Kerbosch, JAGM and Schell, HJ: *Finding cliques in an undirected graph*, 1972, Technische Hogeschool Eindhoven

[8] Pacheco, Peter: *An introduction to parallel programming*, 2011, Elsevier

[9] Eberly, David H: *GPGPU Programming for Games and Science*, 2014, CRC Press

[10] Cheng, John and Grossman, Max and McKercher, Ty: *Professional Cuda C Programming*, 2014, John Wiley & Sons

[11] Scarpino, Matthew: *OpenCL in Action: How to Accelerate Graphics and Computation. NY*, 2011, USA: Manning Publcations

[12] Dean, Jeffrey and Ghemawat, Sanjay, *MapReduce: simplified data processing on large clusters*, Communications of the ACM, Vol 51,No 1, 2008, pp. 107-113

[13] Wadkar, Sameer and Siddalingaiah, Madhu and Venner, Jason: *Pro Apache Hadoop*, 2014, Apress

[14] GHEMAWAT, SANJAY AND GOBIOFF, HOWARD AND LEUNG, SHUN-TAK: *The Google file system*, ACM SIGOPS operating systems review, Vol 37, No 5, 2003, pp. 29-43

Production Systems and Information Engineering
Volume 7 (2015), pp. 55–65

WEB BASED DESKTOP ENVIRONMENT

Imre Piller
University of Miskolc, Hungary
Department of Applied Mathematics
piller@iit.uni-miskolc.hu

Sándor Fegyverneki
University of Miskolc, Hungary
Department of Applied Mathematics
matfs@uni-miskolc.hu

[Received August 2015 and accepted November 2015]

Abstract. The web browser has become the natural platform of the graphical applications. This paper introduces a desktop environment concept which makes the development and deployment of the applications easier. The approach combines the thin client technology and locally installed servers. This hybrid solution is proper for the common use cases. The paper primarily analyzes two aspects of the desktop environment; the rendering of the desktop and the synchronization of the user data. The proposed solutions are compared with the available similar alternatives.

Keywords: web desktop, thin client, display manager, version control

1. Introduction

There are usually some misconception of the user about the importance of underlying software stacks. We can found many debates on the Internet about various operating systems and kernels. Most of them only compare the thin top layers namely the desktop environment. This recognition suggests that from the viewpoint of the user the underlying layers are irrelevant because they provide non-functional features.

The graphical user interface toolkit (near the top of the software stack) has huge impact to the user experience. The commonly used components have already standardized but their behaviour and appearance are not defined exactly. It results inconsistent look and feel when we use applications from a different kind. It is the source of some further problems. The inconsistent terminology and graphic elements make the usage of the software harder. The various dependencies of the widget toolkit libraries cause unnecessary redundancy and sometimes make the communication between software problematic. (For example special clipboard and IPC implementations.)

The personal computing means nowadays that the user typically works with at least two computers. We can assume that the computers are different in some aspects. (The user works with more computers because all of them have a benefit for example the location, the performance, the storage size or the weight is more appropriate in a given situation.) The data are usually shared between the machines. It makes necessary to synchronize the data.

The thin client technology is able to solve the previously mentioned problems. The client is only responsible for displaying the contents and for sending the commands to the servers. The applications are provided by the server therefore the set of applications can be collected carefully by the system administrator. The synchronization is not a problem because all data is stored on the server. However there are some disadvantages also. For instance the thin client conceptually has only a limited offline functionality.

The paper concentrates on the consistent visual representation of the user interface and the problematic of the synchronization. It shows compromised solutions between the thin client and offline workstation.

2. Related Works

The term "web based" here means that the desktop application is a web application where the server can be local or remote. We can found some similar architectures in the contemporary operating systems.

The Bell Laboratories has developed an operating system for distributed servers [1]. Its main design goal is to became graphical, portable and networked. The system's programming language is the Limbo. For portability it uses machine independent bytecode and the Dis virtual machine. (The concept of the project is close to the Java Environment. Basically this project was the concurrent technology when the Java project started.)

The FirefoxOS is an operating system designed for the web [2]. It uses open standards and provides a low cost alternative for smart devices. It's layered architecture consists the following parts.

- *Gonk*: A common layer for the Linux kernel and the Hardware Abstraction Layer.
- *Gecko*: Web browser engine. Its main task is to render HTML pages.

- *Gaia*: User interface implementation written in HTML, CSS and JavaScript. It communicates over the Open Web API therefore the applications can be works in other web browser.
- *HTML5 applications*: The FirefoxOS is able to run web applications. It provides access to device drivers.

We can use applications in online or offline mode. The online mode is the same as in the case of desktop web browsers. For offline usage we need to install the application. In this case, instead of nativ application we can run the application server on the given device.

The Chrome OS (and its open source version called Chromium OS) is also a browser oriented operating system [3]. Technically this is a Linux distribution with some unique features and hardware support. Primarily it is a thin client which provides access to Google services.

There are other operating systems which are designed directly to the web [4]. Their interfaces resembles to the frequently used desktop environment. The most notable representatives are the *eyeOS*, *Glide OS* and the *Joli OS*.

As we can see the browser has become a platform in the Web 2.0 era. The application of thin client is not a trivial decision. The main differences of the architectures originated from the measure of network dependency.

3. Interaction with the User

3.1. Desktop as Primary Interface

The desktop environment is an essential part of the interaction with the user. We can see the main steps of interaction cycle in Figure 1. The input and output devices are physically available for the user. In the case of web browser the typical input devices are the keyboard and mouse, the output devices are the screen and the speaker.

We can notice the *Model-View-Controller* pattern in the cycle. The controller here responsible for registering and preprocessing the input. The application logic is stateful in most cases. The layout arrangement calculates the positions and sizes of the visual elements. In the rendering step from the tree-like component structure an image will be created.

3.2. Online and Offline Works

The online services have many advantages. For instance, they require only a minimal standard installation because most of the applications are installed on the remote server.

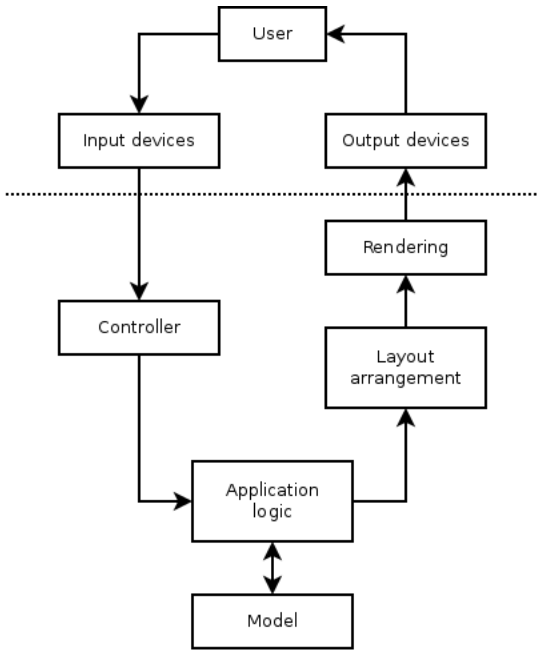

Figure 1. The main steps of the interaction.

The platform of this type of software is the web browser. It results that the applications are portable. We can use the software on any device which can run web browser.

The reliable and fast network connection is a natural requirement in information technology. In the case of online services these requirements are essential. When the network connection is slower or we have no connection at all these online services are unable to work.

The thin client technology often assumes that the client device has only a limited resources and cannot run the application locally. The computational power of smart devices is enough for managing file systems and databases. The thin clients are still preferred because intensive calculations requires more energy and the battery becomes the bottleneck.

4. Layered Architectures

4.1. Client types

We can distinguish different client types as we can see on Table 1. The clients are classified by their components. The literature use the *light*, *rich* and *heavy* names as synonyms in general [5]. Here these names reflect to its complexity. In the following paragraphs their advantages and disadvantages will be summarized.

type	Model	App. L.	Controller	Layout Arr.	Rendering
thin					
light					×
rich			×	×	×
heavy		×	×	×	×
desktop	×	×	×	×	×

Table 1. The types of the clients.

thin client: The client only display the previously rendered image. It requires only a minimal amount of resources on the client side but assumes that the network is able to stream the view of the desktop real-time. This solution is very flexible on the client side but the rendering requires graphical capabilities on the server. For example the VNC client and remote desktops belong to this category.

light client: In this case the server send document object, vector graphics or only the difference of the desktop image from the previous one. Most of the web pages can be considered as light client because the browser waits the response from the server and render graphical elements from the HTML source code.

rich client: The web browser usually able to manage the user input events and calculate the layout arrangement. In web environment the rendering provided by the HTML rendering engine. As *rich* suggests the *Rich Internet Application*s are in this type.

heavy client: The application runs in the client. The persistent storage still on the server side. This construction is beneficial for example the case of calculation intensive applications. The drawback is that the client must have enough resource for running the application but unable to work offline. The clients of online games are heavy clients in this sense.

desktop client: The desktop client means that all components and data are available for offline working. The *client* in the name is intentional. The model is available on the machine but the data are shared or the application partially depends on online services. It can be for instance the file sharing or mail client.

4.2. Function Locations

The client-server model is a simple distributed architecture. The reliable network connections give a choice to locate some functionality either the client or the server side. We have possibility to separate some elements of the server side to remote machines. On Figure 2 we can see the two preferred construction of the layered architecture. At this step we have already assume that the user works with web browser on the client side.

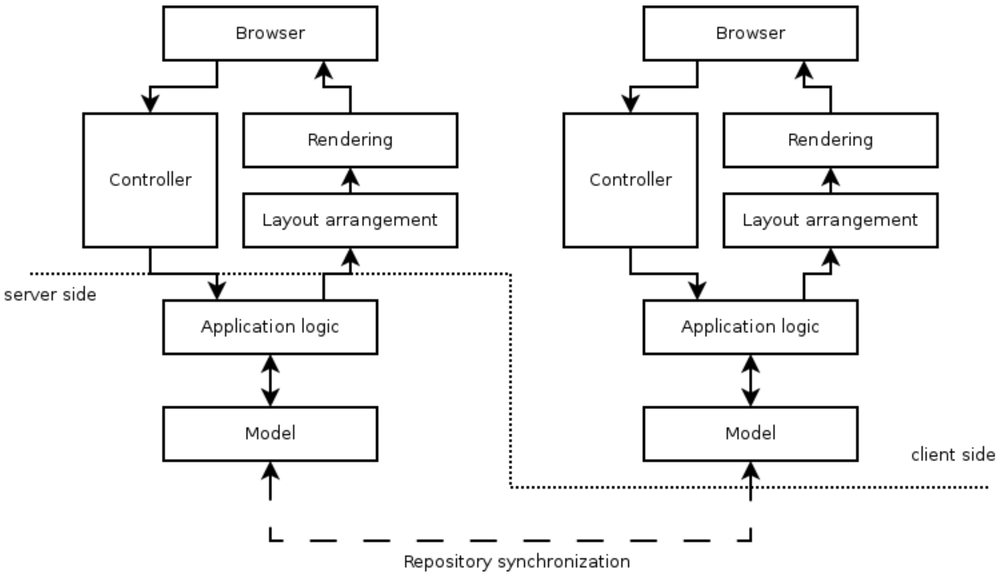

Figure 2. The preferred layered architectures.

The two configuration nearly the same because they use the same interfaces. The difference between them is the location of the application and the model. The left is a rich and the right is a desktop client. We prefer these types because in the case of desktop applications the hardware requirement of the application engine and the model are similar. (In many applications the logic and the model are not separated.)

At the bottom of the figure we can see that the models are connected and able to synchronize the data. The model basically has two parts: *file storage* and *metadata*.

The file is a unit of data. We use files from many reasons, for example,

- the data is unstructured,
- we cannot edit with online editor,

- we are trying to be compatible with other systems,
- the file is large and we cannot store it in the database,
- the filesystem is common, available and supported.

The metadata stores structured data. It provides consistency checking and flexible queries. The type of the database is not important at this point because the choice is highly depends on the given application.

From the user's viewpoint the application is a web application. Many rich clients can use the same web application while in the case of desktop client it presumably dedicated to the user. Therefore the web application is not necessarily installed by the user but enables to install locally for offline work.

The controller, layout arrangement and rendering can be implemented in the browser. It is usually developed in JavaScript as one of the most dominant programming language of the Internet. (In the web browser we have only some other alternatives, because the support of other programming languages is rare.)

5. Optimal selection of configuration

The proposed architecture makes possible to choose between configurations. The decision is based on the following factors.

- *Network availability, and reliability*: We can configure the application for online or offline work. Most of the applications are sensitive to network problems. The periodically updated local repository helps to smooth the differences between the real time online work and offline workstation-like usage.
- *Local resources*: We need to examine the impact of locally installed applications and stored shared files. In general, the synchronization of all files from the repository is not an appropriate solution.

The length of the updating process in the case of distributed configuration based on the speed of network and from the amount of data. It does not affect the behaviour of the application after the synchronization is ready.

We can consider the model as the part of the application from the aspect of performance. Therefore we need to compare the cost of local configuration $(t_l + c_l)$ and the cost of remote configuration $(t_r + c_r)$, where

- t_l, t_r: average response time for locally and remotely installed application,
- c_l, c_r: average time cost of local and remote data exchange.

The $(t_r + c_r) - (t_l + c_l)$ value shows the average response time advantage of the locally installed application.

6. Desktop Interface

The proposed concept help us to create application to this distributed environment. In the following sections the part of the toolkit will be presented.

6.1. User input and the controller

The considered desktop environment uses the keyboard and the mouse as the input devices. It defines the following events.

- *key down/up*: Uses keycode for identify the key.
- *mouse move*: It contains the position (x, y) of the mouse cursor.
- *mouse down/up*: It identifies the mouse buttons similarly to the keys.

The controller is responsible for creating or abtract the event. For instance instead of sending the raw input to the application it can send request for starting or stopping a process, clear the input field or other context dependent commands. It has an internal state but it is limited and not overlapped with the application logic.

6.2. Layout arrangement

The application only sends responses to the requests. In the layout arrangement step the following informations are combined:

- the data which should be presented for the user,
- the rules of the layout arrangement,
- the previous arrangement.

It is hard to divide the functionality of layout arrangement and rendering. As a guideline we should consider only with the structure of the user interface and leave tasks related to the appearance for the rendering.

6.3. Display renderer

The mentioned simple interface can be implemented on any hardware or software platform. The trends suggest that the most appropriate tool nowadays is the HTML5 [6]. It is able to listen the keyboard and mouse events and to display the rendered image.

We can trust the design decesions of HTML5 canvas. It inherits the primitives of widely used graphical toolkits. When we build our display manager on the top of this primitives we can assume that it will have hardware support in the future.

The main advantage of HTML5 is the web browser support. The HTML5 canvas is available in all major browser. It has no standardized widget toolkit yet. For a higher level functionality (for example windowing) it is necessary to use a third party Javascript library or creating a new one. The portability of the applications remains but the consistent look and feel is not guaranteed.

7. Repositories and synchronization

The web application is responsible for the synchronization. It must make available the current data automatically [7]. For synchronizing the data the web applications require the change sets from other web applications instances. From this aspect we can consider the application as a version control system. The database behind the desktop environment is practically a repository [8].

We must distinguish two cases in synchronization. At the simple case the user works with only one browser at the same time. It is a plausible assumption, because the environment is designed for multitasking and for supporting the all required feature. As an advanced scenario, the user is able to work with many desktops parallel. In this case it is hard to avoid collisions and inconsistent states. The login interface helps us to restrict the user to work with a single instance of the desktop.

7.1. Log based synchronization

The structure distinguish the data and the metadata. Ideally one database should enough but the management of binary files make necessary to store the files without knowing its internal format. It require different approach for synchronizing the metadata and the file storage part.

The log based synchronization is beneficial, because

- the calculation of deltas is straightforward since all changes are discrete events with timestamp,
- makes merging easier; the problematic steps can be detected and corrected,
- helps to follow the growth of the repository.

We can group the changes into named sets of changes, but this is optional. The commit messages help to make the modifications easier to follow but the conflicts are not necessary separated by revisions.

7.2. Repository types

The local repository often contains only the subset of the managed data. We can create repository for working and for backup. We can define further types also by considering the following properties of the files.

- *file size*: We expect that the synchronization of the repository will be fast and the size of local copy is small as possible. We can avoid large local storage size via on-demand synchronization.
- *last access or modification date*: Presumably we will work on a set of files on a given period. From this reason we can remove the old files from the local repository.
- *importance*: This is a subjective factor. Some cases it does not turned out from the access or modification dates.

As we can see, the repository types are the results of adaptation to the user behaviour. It is a complex optimization problem which try to achieve small repository size and prompt access at the same time.

8. Conclusion

The paper considers two difficulties in providing convenient desktop environment, namely the graphical representation and the repository synchronization. The preferred solution is based on contemporary web technology (standards, software, tools and methodology). It has showed a possible categorization of client types according to the functions of the client. We prefer the rich and desktop clients because the comparisons show that these configurations are able to adapt to the available hardware and software environment properly. These represent a compromise in the aspect of low resource requirement and the possibility of offline work.

Acknowledgements

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] S. Dorward, R. Pike, D. L. Presotto, D. M. Ritchie, H. Trickey, P. Winterbottom: *The Inferno Operating System*, Bell Labs Technical Journal, Vol. 2, No. 1, Winter 1997, pp. 5-18.

[2] Network, Mozilla Developer, and Individual Contributors. "*Firefox OS architecture*." (2014).

[3] Azad, S. "*Chrome OS and System Architecture.* Retrieved November 3, 2014." (2012).

[4] Lawton, George. "*Moving the OS to the Web.*" Computer 3 (2008): 16-19.

[5] Satyanarayanan, Mahadev. "*Pervasive computing: Vision and challenges.*" Personal Communications, IEEE 8.4 (2001): 10-17.

[6] W3C, *HTML5 Recommendation*, `http://www.w3.org/TR/html5/`, 2014.

[7] Karlson, Amy K., Greg Smith, and Bongshin Lee. "*Which version is this?: improving the desktop experience within a copy-aware computing ecosystem.*" Proceedings of the SIGCHI Conference on Human Factors in Computing Systems. ACM, 2011.

[8] Constantin, Camelia, et al. "*A desktop interface over distributed document repositories.*" Proceedings of the 15th International Conference on Extending Database Technology. ACM, 2012.

Production Systems and Information Engineering
Volume 7 (2015), pp. 67–80

DATA MODEL FOR HYBRID INDOOR POSITIONING SYSTEMS

Zsolt Tóth
University of Miskolc, Hungary
Department of Information Engineering
zsolt.toth@iit.uni-miskolc.hu

Péter Magnucz
University of Miskolc, Hungary
Department of Information Engineering
peter.magnucz@uni-miskolc.hu

Richárd Németh
University of Miskolc, Hungary
Department of Information Engineering
richard.nemeth@iit.uni-miskolc.hu

Judit Tamás
University of Miskolc, Hungary
Department of Information Engineering
judit.tamas@iit.uni-miskolc.hu

[Received September 2015 and accepted November 2015]

Abstract. Indoor Positioning Systems have gained attention over the last decade. Despite of numerous attempts and solutions, there is no common standard indoor positioning system such as the GPS for outdoor environment. Smart phones seem to be a good candidate for mobile devices of indoor positioning systems due to their low cost and incidence. The sensor sets of the most popular mobile platforms are analyzed from the point of view of indoor positioning and a formal data model has been created in this paper. The presented model allows the development of hybrid indoor positioning algorithms. The data model is used to design and implement a Java library which will be used to create an indoor location and navigation framework.

Keywords: indoor positioning, mobile platforms

1. Introduction

Location based services became the part of our life. Global Navigation Satellite Systems [1, 2] are essential in transportation, logistics, military, aviation, etc. Augmented reality has become a hot topic in the last years. Smart phones, –watches and glasses– are possible candidates to make augmented reality available to the masses. Google Indoor Maps makes it possible to share the map of buildings on Google Maps, moreover big airports, railway stations, malls and other public areas already have been modeled.

Despite of the numerous efforts to create indoor positioning system [3, 4], there is no single commonly used solution on the market. These systems have different complexity, installation and maintenance costs, performance and requirements. Developers usually have to make a trade–offs between performance and costs. For example a system with high accuracy usually requires special hardware and infrastructure thus its cost is high. Moreover the special hardware requirements limit the usage and the spread of the system. The systems which use the already established infrastructure have low installation cost but they have lower accuracy and precision than the special–purpose systems. Hence the smart phones seem to be good possible candidates for client of indoor positioning systems due to their availability, easy programmability and wide sensor set.

The analysis of the widely available sensors is necessary to design the data model for indoor positioning systems [5]. The three dominant mobile platform were analyzed in this work. The analysis of the sensor sets makes it possible to create a mathematical description and a formal data model for hybrid indoor positioning methods. The presented data model was implemented in Java. The library supports JSON serialization which is essential for web applications with multi–platform mobile clients.

2. Mobile Platforms

Smart phones are excellent candidates for client devices of indoor positioning system due to their availability, popularity and wide range of built–in sensors. The smart phones became popular in the early 2000's and these days hundreds of them are produced by four major producers. According to the statista.com [6], the smart phone market was shared between the Google, Apple, Microsoft and Blackberry in the United States. Google's Android operation system has slightly more than half of the market. The Apple's iOS system is the second dominant and owns approximately 40% of the market. Microsoft have acquired only 5% of the market over the years. The Blackberry produces devices for Enterprises so they have less potential customers than the previous producers.

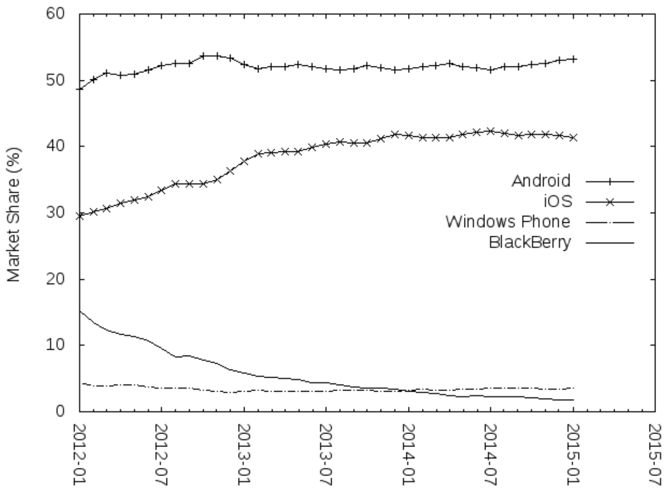

Figure 1. Market Share of Mobile Platforms

Android is the most popular and most widely used mobile platform. Numerous manufacturers offer huge range of devices so that there is huge variety in hardware, computational capabilities and performance of Android devices. Applications have to be implemented in this heterogeneous hardware which is standardized by the Android operation system. Although the API allows to implement the same application on various Android devices, but precision and accuracy of the mobile sensors depend on the quality of the device. Thus the mobile device affects on the measurement due to the heterogeneity of the Android platform.

Although iOS can be run on only iPhone and iPad devices Apple still has about 40% of the market share in the United States. The limitation of the possible architectures allows to optimize the operation system and applications to the supported ones. Moreover this limitation increases the robustness and quality of applications and simplifies the development. On the other hand, the iOS development has special requirements for the developers such as specific operation system which limits its applicability in research projects.

Both Windows Phone and BalckBerry have only 5% of the market of smart phones so they are not leading but they are relevant. Windows Phone has joined the most recently to the market of smart phones. Microsoft provides efficient tools for developers. Moreover Windows Phones can be programed in C# and .Net Libraries are also available for the developers. Blackberry produces smart phones for enterprises thus these products are bought by companies. Moreover Blackberry provides various services to handle and manage the devices remotely and other producers do not have similar services.

3. Sensors

Most of the available sensors are detailed and evaluated from the point of view of indoor positioning in this section. These sensors are categorized based on their purpose so motion, environment and radio frequency sensors are discussed. The applicability for indoor positioning was the main selection criteria. Availability, platform support, incidence are also important features.

3.1. Motion

Motion sensors measure the movement of the device. Numerous popular applications are built on the usage of these sensors such as step counters, fitness application or simulators [7]. From the point of view of indoor positioning these sensors can be used for tracking devices [8]. Calibration is required for these systems which limits the applicability of these sensors in real life situations. On the other hand, motion sensors based measurements can support positioning systems and can be applied in tracking services.

Accelerometer measures the acceleration of the device in a three dimensional space. Equation 3.1 shows the calculation of acceleration vector $\overline{a}$ where $\overline{g}$ stands for gravity, $\overline{F}$ denotes the forces acting upon the device and m is the mass of the device.

$$\overline{a} = -\overline{g} - \frac{\sum \overline{F}}{m} \tag{3.1}$$

Gyroscope and Rotation Vector give data about the spinning of the device via three axes. However these values could be applied to determine the orientation of the device, but they are not related to positioning directly. Proximity sensor determines the distance of the device from other objects. Its result can be given in centimeter or logical value. Proximity sensor is not suit for indoor positioning tasks. Thus our data model is based on neither gyroscope nor proximity sensors.

3.2. Ambient Environment

The characteristics of the environment can be used to determine the location. Location and their functions are often recognized by their characteristics and signs. For example a lecture hall can be recognized based on the organization and it can be identified by its number. Temperature and humidity also can suggest the function of place. There are systems that creates unique environment characteristics for mobile devices. For examples the Active badges [9] uses infrared sensors which are installed at entrances. Visible light communication based indoor positioning offers high accuracy [10] but it requires a specific infrastructure thus its installation cost is high. Ultrasonic signals are also widely used [11, 12].

The usage of magnetic field for indoor positioning was suggested by various researchers [13, 14]. Mobile platforms provide interface for the measurement of the magnetic field. Android has a built–in model to convert the results of magnetometer into GPS coordinates and vice-versa. This model was created by the United States National Geospatial-Intelligence Agency and it was valid until 2015. The indoor environment affects on the magnetic fields. Electronic devices and large metal objects can generate noise for the magnetometer.

Smart phones usually have numerous built–in sensors to measure the environment such as temperature, relative humidity, or pressure. These sensors are meant to give information about the environment for further processing. Temperature sensor can be used to warn the user about the possible overheating of the device. These kind of measurement can vary in the building but their values can be used to distinguish certain positions or functionality. For example an ice–room can be identified by its temperature and a greenhouse has unique relative humidity parameters. But generally, these parameters do not change within a building. Moreover, in industrial facilities like an electronic product line these parameters are fixed or limited. Hence these sensors are not included in the proposed model.

3.3. Radio Frequency Signals

Radio frequency signals are widely used in wireless communication solutions. Short–range wireless protocols such as WLAN [15, 16, 17], RFID [18, 19, 20] and Bluetooth [21] are used for indoor positioning so only these sensors will be discussed. Moreover WLAN and Bluetooth are supported by all of the dominant platforms and RFID readers also became available in the modern smartphones.

Bluetooth is a short ranged communication protocol. Its range is typically less than 10 meters in mobile devices. Bluetooth modules has unique identifiers which can be used for tracking objects. Both three major mobile platform provides API for Bluetooth programming. The Received Signal Strength Indication of Bluetooth devices were used to create a triangulation based indoor positioning method [21]. Two mobile devices can communicate via Bluetooth, if they supports the same protocol and they are within the range. Equation 3.2 shows the formal description of Bluetooth scan with a protocol version v.

$$B_v = \{id_0, id_1, \ldots, id_n\} \tag{3.2}$$

RFID tags are widely used in industrial environment, logistics and security systems. Near Field Communication is a subcategory of RFID technology which allow short range wireless data exchange among devices. Due to the popularity gain of NFC technology, the recent smart phones have built–in RFID readers. Typical range of NFC is less than 10 cm which limits its applicability. However the current built–in NFC sensors are not capable for indoor positioning, hand–held RFID readers can be connected via Bluetooth. Landmark [20] and SpotOn [18, 19] are RFID based indoor positioning systems. The RFID readers detect the tags that are within a given range. The range depends on the setup of the reader and tags. Thus the result of an RFID measurements can be described as Equation 3.3:

$$RFID = \{tag_0, tag_1, \ldots, tag_n\} \tag{3.3}$$

WiFi Card is a common interface of mobile devices. The first fingerprinting based Indoor Positioning System, RADAR [15], was based on the Received Signal Strength Indication of WLAN Access Points. Experiments on the Horus system showed that the RSSI values fluctuate, so the usage of RSSI vector instead of a single value was proposed [16, 17]. The measurements of a WiFi RSSI can be formalized as a set of access points and RSSI values which is shown by Equation 3.4 where ap stand for the SSID of an Access Point and $rssi \in R$.

$$WiFi = \{(ap, rssi)\} \tag{3.4}$$

4. Formal Model

The analysis of the available sensor set allowed to create a formal data model for hybrid indoor positioning methods. The position is also included into the formal model because it is required for fingerprinting based methods. The presented data model is used in the development of the ILONA (Indoor LOcation and NAvigation) system.

4.1. Selected Sensors

Table 1 gives a brief summary of different sensors on various platforms. The main selection criteria were the applicability for indoor positioning, availability and platform support. Bluetooth, GPS, magnetometer, RFID reader and WLAN network interface card were selected because they are vital components of smart phones. RFID reader was chosen because it is integrated into modern Android devices and the RFID technology is significant in the research of indoor positioning systems.

Table 1. Commonly Available Mobile Sensors

Sensor	Android	iPhone	Windows Phone	Chosen
Accelerometer	✓	✓	✓	
Bluetooth	✓	✓	✓	✓
Camera	✓	✓	✓	
Gravity	✓			
GPS	✓	✓	✓	✓
Gyroscope	✓	✓	✓	
Light	✓	✓	✓	
Magnetic Field	✓	✓		✓
Microphone	✓	✓	✓	
Orientation	✓		✓	
Pressure	✓			
Proximity	✓	✓	✓	
Relative humidity	✓			
RFID				✓
Rotation Vector	✓			
Temperature	✓			
WiFi Card	✓	✓	✓	✓

4.2. Mathematical Description

The measurements of the selected sensors can be modeled formally as a tuple which is shown in Equation 4.1. Measurement $\mathcal{M}$ consists of data that is read from the selected sensors ($\mathcal{S}$) and a time stamp which is used to distinguish measurements and a position ($\mathcal{P}$) which is set by the user. The timestamps and the position together determine a measurement because it is possible to perform multiple measurements in the same location.

$$\mathcal{M} = \langle \mathcal{S}, timestamp, \mathcal{P} \rangle \tag{4.1}$$

Let $\mathcal{S}$ be a tuple of the read sensor data and given in Equation 4.2. Missing values are allowed in $\mathcal{S}$ due to the variety of the mobile devices. Moreover the existing indoor positioning systems usually rely on a single sensor. The

$\{bt_{tags}\}$ represents the set of available Bluetooth devices. GPS coordinate is unavailable in indoor environment but it can be used in campuses and hospitals which contain many buildings. The magnetometer returns with a vector (x, y, z) which point the North. It can be used to determine orientation of the user. The sensed RFID tags are represented as set in the model. Finally, the WLAN interface can be used to scan available access points. The measurement yield key–value pairs where the key is the SSID of the access point and the value is the measured RSSI.

$$\mathcal{S} = \langle \{bt_{tags}\}, (lat, lon, alt), (x, y, z), \{rfid_{tags}\}, \{(ssid, rss)\} \rangle \tag{4.2}$$

Position can refer to either a symbolic zone of the building or relative coordinates. Zones are considered as well defined disjunct parts of the building. Thus zone can denote a room, a corridor or even a part of the building. Zones are used by symbolic or room level positioning methods. The relative coordinates denotes the distance from a fixed point which is the origin of the coordinate system. Absolute positioning methods use the relative coordinates and the GPS coordinates can be calculated.

$$\mathcal{P} = \langle zone, (x, y, z) \rangle \tag{4.3}$$

This mathematical description allows to handle the measurements formally and create operations among them. However this abstract model cannot be used directly by indoor positioning systems because technical details have been omitted. For example, the mathematical model do not define data types nor constrains.

4.3. Data model

Entity-Relationship model has been created based on the mathematical description in order to simplify the implementation of the formal model. Figure 2 shows the entity–relationship model which describes the measurements and the positions. The data model contains unique identifiers for both measurements, positions. Zone identifier was created because the name of the zone is not necessarily unique. For example there are ground floor in every building or each floor could have the same room numbering. Because either the zone and the coordinate can be unknown the position identifier has to be defined. In the mathematical model, a measurement could be identified by the position and the time stamp but also a unique identifier has been defined for each measurement, due to the following reasons: Firstly, the positions also identified by identifier. Secondly, the position is unknown during the positioning and the identification of measurements is necessary to monitor the positioning algorithm.

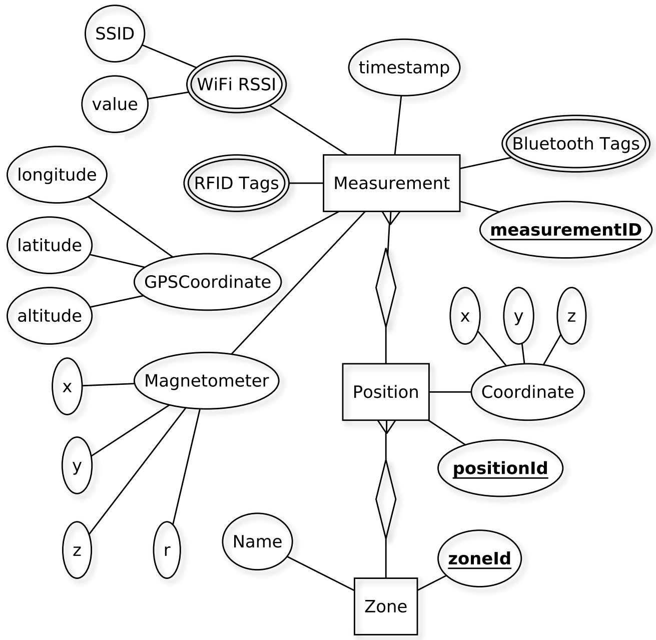

Figure 2. ER model of Measurements

5. Implementation

The above detailed model has been implemented in Java. Maven was used to build the component and to manage its dependencies. JUnit and EasyMock were used for testing and Jackson JSON Processor Library was used to serialize and deserialize objects.

The classes of the components are shown in Figure 3. The `Measurement` class is used to wrap the values of the sensor. The measurements of the selected sensors are represented by corresponding classes such as BluetoothTags, `Magnetometer` and `WiFiRSSI`. These classes were implemented as beans because this was required for Jackson.

The instantiation of `Measurement` object is difficult because it is a wrapper class. The number of the possible valid constructors is the combination of the wrapped types. So the number of the constructors grows rapidly. The Builder pattern [22] was used to simplify the instantiation of `Measurement` objects.

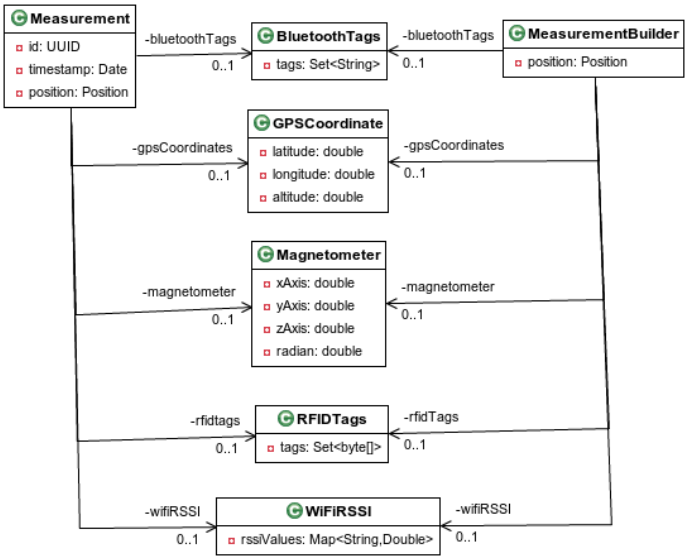

Figure 3. Measurement Implementation

Due to its protected constructor, `Measurement` objects can be instantiated only from its package and the `MeasurementBuilder` can be used to create them.

5.1. Serialization

Jackson JSON Processor was used to serialize the objects. The JSON format was chosen due to its readability and compactness. Moreover, REST based web application are popular nowadays and JSON format is required by them. Restful web applications communicate via HTTP protocol and send object in JSON format. So that platform-independent, widely available, reusable, component based web applications can be developed rapidly.

```
{
"coordinate":{"x":0.0,"y":0.0,"z":0.0},
"zone":{
  "id":"47aa248b-5852-43a6-af34-21b9771517c8",
  "name":"Lab 115"},
"uuid":"a014dee1-ea28-4c66-b160-87a58becf9ff"
}
```

Figure 4. Position JSON

The `Position` objects can be represented by coordinates, a `Zone` object or both of them. Figure 4 shows the JSON format of a `Position` object. The coordinate consists of zeros because it was created during the tests. The zone element shows the structure of the `Zone` objects which have an identifier and a name. Development of a refined zone model could be necessary for navigation services.


```
{
"bluetoothTags":null,
"gpsCoordinates":null,
"id":"99631985-b254-4598-886a-33033521d4b8",
"magnetometer":{
  "xAxis":0.34035778045654297,
  "yAxis":-0.04757712036371231,
  "zAxis":0.0158716831356287},
"position":{
  "coordinate":null,
  "zone":{
    "id":"47aa248b-5852-43a6-af34-21b9771517c8",
    "name":"Lab 115"},
  "uuid":"a014dee1-ea28-4c66-b160-87a58becf9ff"},
"rfidtags":null,
"timestamp":1444932266450,
"wifiRSSI":{
  "rssiValues":{
    "TP-LINK_517E90":-90.0,
    "teszt":-78.0
    }
  }
}
```


Figure 5. Measurement JSON

Figure 5 shows the serialized form of a `Measurement` object which contains a time stamp, an identifier, magnetometer and WiFi RSSI measurements. The time stamp is serialized as a long number which is the standard form of UNIX times stamp. The magnetometer contains a vector which points to North. The WiFi RSSI contains a set of key value pairs, where the key is SSID of the access point, and the value is the RSSI value.

6. Conclusion

Formal model for hybrid indoor positioning systems have been proposed. Analysis of the sensor sets of the popular mobile platforms was performed to select the widely available, relevant and useful sensors for indoor positioning. The existing indoor positioning systems were reviewed from the point of view of their applied technologies. Mathematical description were given for the measurements which can be used to develop positioning algorithms. A data model has been created based on the mathematical model which introduces domain constrains for the measurements which would be hard to define in the mathematical model. Based on the data model a Java Library was designed and implemented. The implementation support JSON serialization due to the increasing popularity of restful web applications. The presented library is used in the development of a indoor location and navigation system which performs the positioning calculations on the server and communicate with the client via HTTP.

Acknowledgements

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] Lechner, W. and Baumann, S.: Global navigation satellite systems. *Computers and Electronics in Agriculture*, **25**(1), (2000), 67–85.

[2] Hofmann-Wellenhof, B., Lichtenegger, H., and Wasle, E.: *GNSS–global navigation satellite systems: GPS, GLONASS, Galileo, and more.* Springer Science & Business Media, 2007.

[3] Liu, H., Darabi, H., Banerjee, P., and Liu, J.: Survey of wireless indoor positioning techniques and systems. *Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, **37**(6), (2007), 1067–1080.

[4] Deak, G., Curran, K., and Condell, J.: A survey of active and passive indoor localisation systems. *Computer Communications*, **35**(16), (2012), 1939–1954.

[5] Kaemarungsi, K.: Design of indoor positioning systems based on location fingerprinting technique, 2005, URL `http://d-scholarship.pitt.edu/6395/`.

[6] Statista.com: Market share held by smartphone platforms in the united states from january 2012 to january 2015. URL `http://www.statista.com/statistics/266572/market-share-held-by-smartphone-platforms-in-the-united-states/`.

[7] Kecső, D.: Okostelefonnal vezérelhető repülőgép szimulátor. 2014.

[8] Tajti, T.: Indoor localization with mobile phone. In *Proceedings of the 1st International Conference on Future RFID Technologies*, 2014.

[9] Want, R. and Hopper, A.: Active badges and personal interactive computing objects. *Consumer Electronics, IEEE Transactions on*, **38**(1), (1992), 10–20.

[10] Zhou, Z., Kavehrad, M., and Deng, P.: Indoor positioning algorithm using light-emitting diode visible light communications. *Optical Engineering*, **51**(8), (2012), 085009–1.

[11] Ward, A., Jones, A., and Hopper, A.: A new location technique for the active office. *Personal Communications, IEEE*, **4**(5), (1997), 42–47.

[12] Borriello, G., Liu, A., Offer, T., Palistrant, C., and Sharp, R.: Walrus: wireless acoustic location with room-level resolution using ultrasound. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ACM, 2005, pp. 191–203.

[13] Brachmann, F., Jenak, I., Horvath, Z., Wu, T., Xuan, C., Baranyi, A., and Jene, S.: The effects of hardware and software-based signal distortion in multi-platform indoor positioning systems. In G. Gartner and H. Huang (eds.), *Proceedings of the 11th International Symposium on Location-Based Services*, ISBN 978-1-907075-07-0, 2014.

[14] Li, B., Gallagher, T., Dempster, A. G., and Rizos, C.: How feasible is the use of magnetic field alone for indoor positioning? In *Indoor Positioning and Indoor Navigation (IPIN), 2012 International Conference on*, IEEE, 2012, pp. 1–9.

[15] Bahl, P. and Padmanabhan, V. N.: Radar: An in-building rf-based user location and tracking system. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 2, Ieee, 2000, pp. 775–784.

[16] Youssef, M. and Agrawala, A.: Handling samples correlation in the horus system. In *INFOCOM 2004. Twenty-third AnnualJoint Conference of the IEEE Computer and Communications Societies*, vol. 2, IEEE, 2004, pp. 1023–1031.

[17] Youssef, M. and Agrawala, A.: The horus wlan location determination system. In *Proceedings of the 3rd international conference on Mobile systems, applications, and services*, ACM, 2005, pp. 205–218.

[18] Hightower, J., Want, R., and Borriello, G.: Spoton: An indoor 3d location sensing technology based on rf signal strength. *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering, Seattle, WA*, **1**.

[19] Hightower, J., Vakili, C., Borriello, G., and Want, R.: Design and calibration of the spoton ad-hoc location sensing system. *unpublished, August.*

[20] Ni, L. M., Liu, Y., Lau, Y. C., and Patil, A. P.: Landmarc: indoor location sensing using active rfid. *Wireless networks*, **10**(6), (2004), 701–710.

[21] Feldmann, S., Kyamakya, K., Zapater, A., and Lue, Z.: An indoor bluetooth-based positioning system: Concept, implementation and experimental evaluation. In *International Conference on Wireless Networks*, 2003, pp. 109–113.

[22] Gamma, E., Helm, R., Johnson, R., and Vlissides, J.: *Design patterns: elements of reusable object-oriented software*. Pearson Education, 1994.

Production Systems and Information Engineering
Volume 7 (2015), pp. 81–92

INVENTORY MANAGEMENT IN NETWORKED SERVICE SYSTEMS

Péter Veres
József Hatvany Doctoral School in Information Sciences
eauchleveier@freemail.hu

Tamás Bányai
University of Miskolc, Hungary
Institute of Logistic
alttamas@uni-miskolc.hu

Béla Illés
University of Miskolc, Hungary
Institute of Logistic
altilles@uni-miskolc.hu

[Received September 2015 and accepted November 2015]

Abstract. The globalization of economy and market leaded to increased networking in the field of manufacturing and services. The processes of these manufacturing and services including logistics became more and more complex. The design and operation of these complex processes can be described as NP-hard optimization problems. These problems can be solved using sophisticated models and methods using metaheuristics based algorithms. Much of the researches in this area are focusing on manufacturing. This paper aims to report a firefly metaheuristics based optimization method, by the aid of which it is possible to support the solution of design and control problems of networked service processes. The authors describe a general model and present a new metrics to measure permutation distances used in the algorithm.

Keywords: optimization, firefly algorithm, route planning, heuristic

1. Introduction

Beside production processes, logistics plays an increasing role in the field of service activities. There are plenty of literatures in planning of production processes, but there are significant gaps in the area of design and operation of service activities. In this research topic, there are perspectives that deliver significant benefits for the development and operation of service processes [1, 2].

This work presents a general model based on a metaheuristic algorithm, which is suitable for optimizing production processes and service support systems.

In the literature several research work discuss the planning and management of logistics processes related service activities. Some of this literatures examine the optimal development of a properly functioning network for logistics processes [3], but only on a concept level, while others have specific models and algorithms used to solve the optimization problems [4, 5, 6].

One of the richest topics of logistics activities in the service area is the city logistics, where distribution problems involve social, cultural and economic effects. Optimizing this particular segment is very important in many aspects [7, 8].

The optimal design of logistics services affects not only the traditional areas of logistics (purchasing, production, distribution, recycling). Significant research is being conducted with the aim to develop methods and algorithms that could improve the logistics service system and greatly reduce the total environmental load [9].

The quality assurance and risk assessment is an interesting aspect in an operating logistic service network, because most of the literature just examine the risk of each participant of a properly functioning network system, There is little research work found in the literature, that manages the risk of a network-based logistics service system [10]. The environmental impact inevitable when planning logistic activities, and this is especially true in the case of logistics processes where service activities supported with high transport costs and performance arise due to the nature of the activities [11].

2. The Model

The present research work aims to create a model to describe logistics processes related to network-based service systems. During the development the aim was to create a common model, from which specialized branch models can be created taking into account various restrictions. The model includes three groups of elements, which is absolutely necessary in logistic processes:

- central warehouses,
- logistic resources,
- objects to be served.

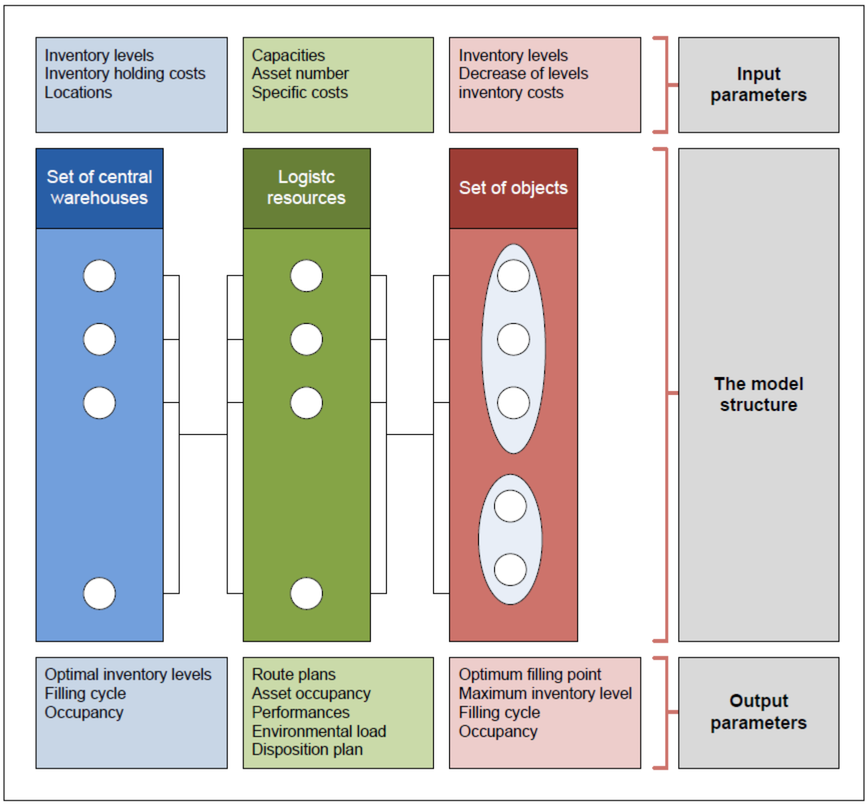

Figure 1. The general model of network-based service systems

The model in Figure 1 explains the relationships among each elemental groups, that can be a subject of optimization.

First, the objective functions need to be described based on Figure 1. The main objective function is to minimize the distance in every route in a time-frame. A route length can be described as a length between the central warehouse and the first object, plus the distances among each objects, and add the length between the last object and the central warehouse:

$$L = \sum_{t=1}^{\tau} \sum_{\alpha=1}^{\delta(t)} \left(l^{KR}_{f(t,\alpha),p_{t,\alpha(t),1}} + \sum_{s=1}^{m(t,\alpha)-1} l_{p_{t,\alpha(t),s},p_{t,\alpha(t),s+1}} + l^{KR}_{p_{t,\alpha(t),m,f(t,a)}} \right) \to min. \tag{2.1}$$

Notations	Explanation of notations
τ	the number of cycles of the investigation period
α	the route ID
$\delta(t)$	the number of routes in the t^{th} investigation period
$f(t,\alpha)$	the central storage ID in the t^{th} investigation period visited by the α^{th} route
β	the objects number in a given route
$p_{t,\alpha,\beta}$	the objects ID number in the t^{th} investigation period visited by the α^{th} route in the β^{th} place
$m(t,\alpha)$	the number of objects in the t^{th} investigation period visited by the α^{th} route
c^L	the specific cost depending on the routes length
$K^{max}_{p_{t,\alpha,\beta}}$	the maximum storage capacity of the β^{th} object
$K^{act}_{p_{t,\alpha,\beta}}$	actual filled capacity of the β^{th} object
K^J_α	the α^{th} route assigned vehicle capacity
T^C	the central storage replenishment cycle time
c^{SZ}	the order-processing cost related to the central storages refill process
β_0	maximum attraction value of the Firefly algorithm
Y	absorption coefficient: usually the value is between 0.1 and 10
r_{ij}	the i^{th} and j^{th} firefly distance
α_t	size of the steps in the algorithm
ϵ_t	randomization parameter

Table 1. Explanation of notations

From the route length an objective function for minimum handling costs can be clearly defined:

$$C = L \cdot c^L \rightarrow min. \tag{2.2}$$

The occupancy rate for each route will define the next objective function. In this case we summarize each vehicle occupancy in the given time period for each objects, depending on the quantities of the delivery.

$$\eta^J = \sum_{t=1}^{\tau} \sum_{\alpha=1}^{\delta(t)} \frac{\sum_{\beta=1}^{m(t,\alpha)} K^{max}_{p_{t,\alpha,\beta}} - K^{act}_{p_{t,\alpha,\beta}}}{K^J_\alpha} \rightarrow max. \tag{2.3}$$

It is also important to minimize the investment costs in terms of the number of routes.

$$\alpha_{max} \rightarrow min. \tag{2.4}$$

The central warehouse plays an important role in the system. During the optimization process we have to take into account the transportation and storage costs because these affect the optimization process of the supply chain's qualities. We can specify the cost of transport, which is defined as the order processing cost in this case:

$${}^{KR}K^{MF} = \frac{\tau}{T^C} \cdot c^{SZ} \tag{2.5}$$

The storage costs in the given timeframe can also be calculated:

$${}^{KR}K^{T} = \sum_{t=1}^{T^C} \sum_{\alpha=1}^{\delta(t)} \sum_{\beta=1}^{m(t,\alpha)} K^{max}_{p_{t,\alpha,\beta}} - K^{act}_{p_{t,\alpha,\beta}} \tag{2.6}$$

Knowing these two costs the objective function for minimizing the central warehouse total cost can be written in the following form.

$${}^{KR}K = {}^{KR}K^{MF} + {}^{KR}K^{T} \rightarrow min. \tag{2.7}$$

Besides objective functions various restrictions and limitations need to be introduced. The first limitation defines that each route lengths have to be between minimum and maximum distance.

$$L^J_\alpha \leq L^J_{max} \text{ and } L^J_\alpha \geq L^J_{min} \tag{2.8}$$

Another important condition states that an object only can be served by one vehicle and one warehouse, so the P matrix is a permutation matrix.

The last function of the model is a limitation. It describes that the volume or weight that the route require cannot exceed the carrying capacity of the vehicle.

$$\sum_{\beta=1}^{m(t,\alpha)} K^{max}_{p_{t,\alpha,\beta}} - K^{act}_{p_{t,\alpha,\beta}} \leq K^J_\alpha \tag{2.9}$$

3. Distance measurement between variations

In the developed algorithm we examined the distance between the various versions of solutions defining metrics, as they affect the convergence of the algorithm. In the case of heuristic and metaheuristic algorithms, the best solution variants have to be found by optimization in different state spaces. Swarm intelligence algorithms types (ant colony algorithm, bee colony algorithm, firefly algorithm) needs different methods to measure the distance between individual solutions [12, 14]. These methods include the Hamming or the Difference distance or their normalized weighted forms and new metrics derived from them. If the solutions in the search space are binary-coded, the Hamming distance is a suitable metric. It is one of the simplest method [13], which uses the number of positions which are not identical with the elements of the two permutations:

$$d_{Ham}(s_1, s_2) = \sum_{i=1}^{n} x_i \text{ where } x_i = \begin{cases} 0, & \text{if } s_1(i) = s_2(i) \\ 1, & \text{else,} \end{cases} \tag{3.1}$$

where

- d_{Ham}: Hamming distance,
- s_1, s_2: different permutations,
- n: number of elements in permutation,
- i: positive integer.

4. The algorithm

In this chapter we will present an algorithm that can solve the above mentioned problem. The Firefly algorithm is a nature inspired metaheuristic algorithm, which has the basis of the behavior of the fireflies. The fireflies signaling system based on using a self created light at night in order to attract the opposite sex. Xin- She Yang created the first version of the algorithm in 2008, with the basic tenets of the algorithm:

- Every bug is unisex, so all of them attracted to each.
- The attraction is directly proportional to the degree of the bug's brightness. The brighter attracts less bright. However, the relative brightness is proportional with the distance.
- The brightest bug moves randomly.

For maximum search problems the brightness of the fireflies is proportional with the objective function, however a fitness function or something similar have to be used when minimum search needed.

The firefly algorithm is invented for solving continuous problems, but with special terms and interpretation it can be highly discreditable, so problems can be solved with simple permutation. The relation between two firefly bug movement (any x_i and x_j firefly movement in iteration $t+1$) can be identified as:

$$x_i^{t+1} = x_i^t + \beta_0 \cdot \exp(-\gamma \cdot r_{ij}^2) + \alpha_t \cdot \epsilon_t \tag{4.1}$$

$$\beta = \beta_0 + e^{-\gamma r} x_i = x_i \cdot (1-\beta) + x_i \cdot \beta + \alpha \left(rand - \frac{1}{2} \right), \tag{4.2}$$

where

- β_0: maximum attraction value,
- γ: absorption coefficient: usually a value between $0, 1, \ldots, 10$,
- r_{ij}: the distance between the i-th and the j-th firefly bugs,
- α_t: randomization parameter, gives us the step number,
- ϵ_t: randomization parameter, a random number.

The application of the algorithm is created with C# programming language. The program reads a data file for input parameters of the optimization problem: the coordinates of the objects or the transport route length matrix, the starting levels and daily level reductions. The steps between levels, the number of days, the number of iterations and the firefly population (versions) must be given. It is possible to set the starting percentage level. To verify the application a test run is required; we create a problem with basic parameters found in Table 1. The test case is derived from the general model and we can observe how efficient the firefly algorithm is in this case. The method now operates with one warehouse and only one vehicle to determine the optimal path and the vehicle capacity set to infinite. In the future we have a plan to expand the problem and the application to any kind of situation that we can determine with this model.

The optimization problem derived from the general model is a task where the goal is to optimize the scattered objects refilling process. In this case the objects are ATM terminals and their positioning is shown in Figure 3. The blue squares are the objects waiting for refilling and the red square is the central distribution warehouse, where every day the vehicles leaves and returns. The routes, in this case, are in flight distance to simplify the testing process. We have to insert the starting levels and the daily average reduction of each object, like we did in Table 2. When a test is made for long term we don't need the exact initial levels because it has minimal effect on the result, but in this case the test requires an accurate reduction measurement [15].

In addition, it is required to define the object's coordinates or transport route length matrix for the application. With the coordinates the program

```
Begin
Initial level (IL), daily average level reduction (DALR), coordinates
(X,Y,[Z]) or distance matrix (L) Specifying or calculation
specify: Step interval (S), iteration (I), number of days (D) and the
number of firefly population (X)
Define objective function: f(x)
for i=1:100/S
Calculate actual minimum: AM= 1+i*S
     for j=1:D
     Determination of the objects in the route:
         if (ILj<AM),
          The object is in the route
         end if
      Generate Random route versions: SOL[]
       for k=1:object number in the route+1
       Specify route distance: M[]=L(SOL[k];SOL[k+1])
       end for k
       Definition of brightness function: Mij, Mij∝f(X) or simply Mij=f(X)
       for l=1:I (iteration number)
             Select brightest (minimum) version
             Examination of the other version how many element is
             different from the brightest: A(X)
             Generate random: B=Rand(1:A(X)-1)
             In every version that different from the best, we need to
             exchange B time randomly selected numbers
             In the best version we have to exchange 2 randomly selected
             number
             Redefine the new brightness: Mijl
      end for l
      Select best daily route: BDMij
      Redefine object actual level
  end for j
  The sum of all the daily minimum route distances for a specific alarm
  level:BDMi=SUM[BDMij]
end for i
Select the alarm level with the least summarized distance
processing results, visualization;
end
```

Figure 2. The pseudocode of the application based on the modell and firefly algorithm

calculates the distance between two objects in flight path, but if we insert the actual transport matrix, the calculation will be more accurate.

After that it is possible to set the affecting parameters of the algorithm. The chosen step between percentages is 2. In this case, the program increases the critical percentage levels with two percent for each testing phase, according to the given time period. We chose the time period 1 year, so the program optimizes routes for the defined 365. The firefly population contains 10 specimens and with 100 iterations the application can easily serve such a small task. The obtained calculation results: the minimum delivery work can be achieved at 7% alarm level, but 23% and 35% also takes up local minimum of the chart.

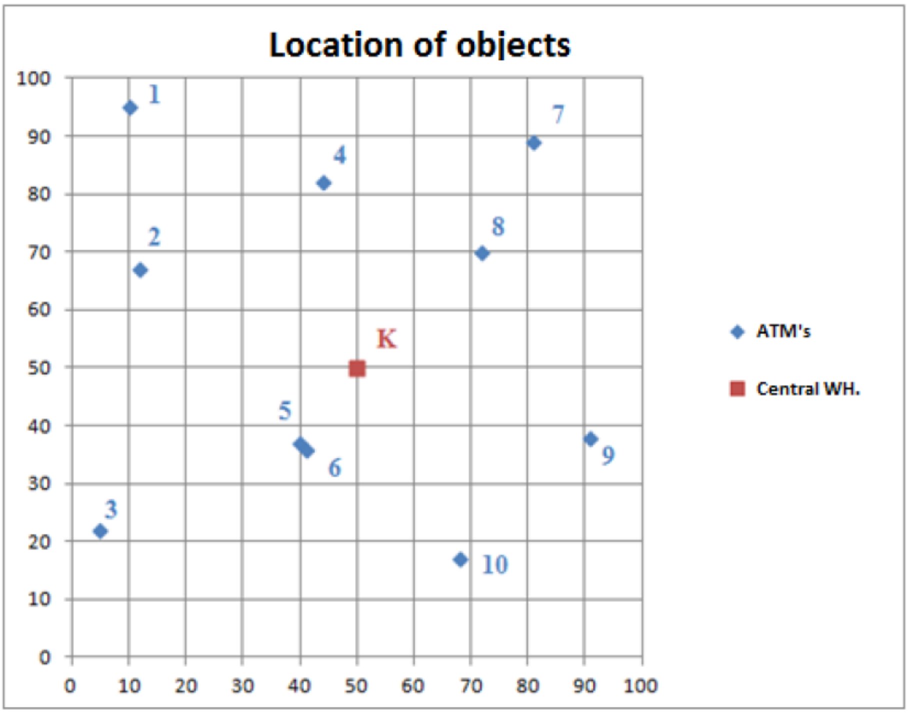

Figure 3. Location of objects

Moreover, the points can be matched very well to a polynomial curve, which shows that from 30% to 35% a low transportation work can serve the objects, and above this limit the transport distance jumps exponentially.

As the results of the processing also show the algorithm is well suited for solving the defined models problems. Of course, the model presented here is considered to be a basic model, further research is still ongoing and a number of challenges need to be taken for creating real conditions as much as possible.

5. Summary

The planning of service processes have to be solved by suitable models and algorithms. In the present research, a firefly-based heuristic optimization algorithm has been developed, that allows to optimize various network-based logistic service activities processes properly.

Objects name	X coordinate	Y coordinate	Starting level	Average daily reduction
K	50	50	100	0
1	10	95	47	9
2	12	67	12	25
3	5	22	83	11
4	44	82	75	7
5	40	37	40	23
6	41	36	61	18
7	81	89	32	22
8	72	70	47	16
9	91	38	25	14
10	68	17	56	8

Table 2. Initial parameters

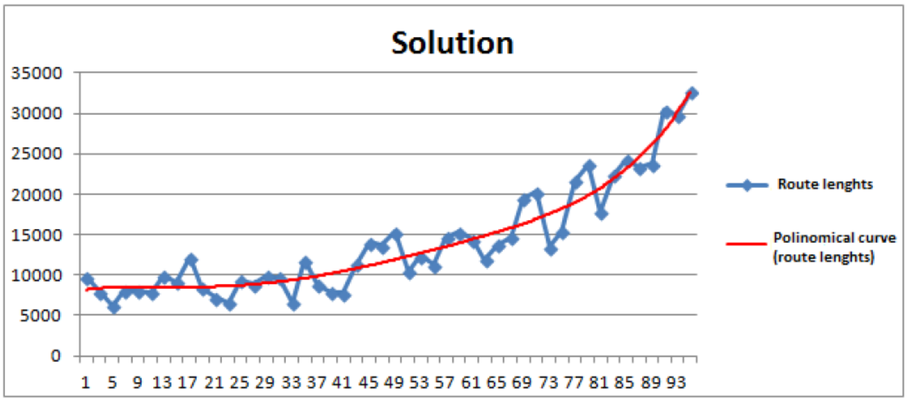

Figure 4. Transport routes lengths depending on the replenishment levels

Acknowledgements

This research was partially carried out in the framework of the Center of Excellence of Mechatronics and Logistics at the University of Miskolc.

The presented research work was partially supported by the grant TÁMOP-4.2.2.B-15/1/KONV-2015-0003.

REFERENCES

[1] SHARAD N. KUMBHARANA, GOPAL M. PANDEY: *Solving travelling salesman problem using firefly algorithm.* International Journal for Research in Science & Advanced Technologies. Issue 2. Volume 2. 2013. pp. 53-57

[2] Á. GUBÁN, Z. MEZEI, Á. SÁNDOR: *Service Processes as Logistic Workflows.* In: Branko Katalinic (ed.) DAAAM International Scientific Book 2014. Bécs: DAAAM International. 2014. pp. 485-500.

[3] S. ZHANG, C. K. M. LEE, H. K. CHAN, K. L. CHOY, Z. WU: *Swarm intelligence applied in green logistics: A literature review.* Engineering Applications of Artificial Intelligence. Volume 37, January 2015, pp. 154–169

[4] R. KÁSA, Á. GUBÁN: *Business Process Amelioration Methods, Techniques and their Service Orientation: A Review of Literature.* In: Vastag Gy (ed.) Research in the Decision Sciences for Global Business. Upper Saddle River: Pearson, 2015. pp. 219-238.

[5] Á. GUBÁN Á. Z. MEZEI, Á. SÁNDOR: *Service Processes as Logistic Workflows.* In: Branko Katalinic (ed.) DAAAM International Scientific Book 2014. Bécs: DAAAM International. 2014. pp. 485-500.

[6] W. LIU, Y. WANG: *Quality control game model in logistics service supply chain based on different combinations of risk attitude.* International Journal of Production Economics, Volume 161. 2015. pp. 181- 191.

[7] Y.H. VENUS LUN, K.-H. LAI, C.W.Y. WONG, T. C .E. CHENG: *Greening propensity and performance implications for logistics service providers.* Transportation Research Part E: Logistics and Transportation Review. Volume 74. 2015. pp. 50-62.

[8] A. DE MARCO, A. C. CAGLIANO, G. MANGANO, F. PERFETTI: *Factor Influencing Logistics Service Providers Efficiency' in Urban Distribution Systems.* Transportation Research Procedia. Volume 3. 2014. pp. 499-507.

[9] A. HOFF, H. ANDERSSON, M. CHRISTIANSEN, G. HASLE, A. LØKKETANGEN: *Industrial aspects and literature survey: Fleet composition and routing.* Computers & Operations Research. Volume 37. 2010. pp. 2041-2061.

[10] P. EVANGELISTA: *Environmental sustainability practices in the transport and logistics service industry: An exploratory case study investigation.* Research in Transportation Business & Management. Volume 12. 2014. pp. 63-72.

[11] C.-N. Liao, H.-P. Kao: *An evaluation approach to logistics service using fuzzy theory, quality function development and goal programming.* Computers & Industrial Engineering, Volume 68. 2014. pp. 54-64

[12] R. O. Large, N. Kramer, R. K. Hartmann: *Procurement of logistics services and sustainable development in Europe: Fields of activity and empirical results.* Journal of Purchasing and Supply Management. Volume 19. Issue 3. 2013. pp. 122-133.

[13] W. Li, Y. Zhong, X. Wang, Y. Cao: *Resource virtualization and service selection in cloud logistics.* Journal of Network and Computer Applications. Volume 36. Issue 6. 2013. pp. 1696-1704.

[14] Z. Shuzhu, C. K. M. Lee, H. K. Chan, K. L. Choy, Z. Wu: *Swarm intelligence applied in green logistics: A literature review.* Engineering Applications of Artificial Intelligence. Volume 37, January 2015, pp. 154– 169

[15] P. Veres, T. Bányai, B. Illés: *Supply chain optimisation of networked services*, 8th. International Doctoral Student Workshop on Logistics, Magdeburg, Otto-Von-Guericke University Magdeburg, 2015.06.23. pp:79-84

Notes for Contributors

related to the papers to be submitted
for Production Systems and Information Engineering

Aims and scope

The aim of the journal is to publish high quality research papers connected with both production systems and information engineering at the same time. Special emphasis is given to articles on the theoretical models and methods, as well as practical applications of discrete production processes including new (or partially new) software tools. Using a new term proposed in special literature in the nineties, the main profile of this journal is Production Information Engineering.

Frequency of the journal

One volume a year (80-200 pages per volume) is planned.

Submission of manuscript

Submission of a manuscript implies that the paper has not been published, nor is being considered for publication elsewhere. All papers should be written in English. Two copies of the manuscript should be submitted on pages of A4 size.

Each manuscript should be provided with an English Abstract of about 50 - 100 words, reporting concisely on the objective and the results of the paper. The English Abstract is to be followed by the keywords.

References should be grouped at the end of the paper in numerical order of appearance. Author's name(s) and initials, paper titles, journal name, volume, issue, year and page numbers should be given for all journals referenced.

For the purpose of refereeing, papers should initially be submitted in hardcopy (twofold) to the secretaries. The eventual supply of an accepted-for-publication paper in its final camera-ready form will ensure more rapid publication. As regards the detailed format requirements, please consider the next homepage: http://ait.iit.uni-miskolc.hu/~psaie.

A Short History of the Publications of the University of Miskolc

The University of Miskolc (Hungary) is an important centre of research in Central Europe. Its parent university was founded by the Empress Maria Teresia in Selmecbánya (today Banska Stiavnica, Slovakia) in 1735. After the First World War the legal predecessor of the University of Miskolc moved to Sopron (Hungary) where, in 1929, it started the series of university publications with the title Publications of the Mining and Metallurgical Division of the Hungarian Academy of Mining and Forestry Engineering (Volumes I.-VI.). From 1934 to 1947 the Institution bad the name Faculty of Mining, Metallurgical and Forestry Engineering of the József Nádor University of Technology and Economics Sciences at Sopron. Accordingly, the publications were given the title Publications of the Mining and Metallurgical Engineering Division (Volumes VII.-XVI.). For the last volume before 1950 due to a further change in the name of the Institution Technical University, Faculties of Mining, Metallurgical and Forestry Engineering, Publications of the Mining and Metallurgical Divisions was the title. For some years after 1950 the Publications were temporarily suspended.

After the foundation of the Faculty of Mechanical Engineering in Miskolc in 1949 and the movement of the Sopron Mining and Metallurgical Faculties to Miskolc the Publications restarted with the general title Publications of the Technical University of Heavy industry in 1955. Four new series Series A (Mining), Series B (Metallurgy), Series C (Machinery) and Series D (Natural Sciences) were founded in 1976. These came out both in foreign languages (English, German and Russian) and in Hungarian.

After the foundation of the Faculty of Mechanical Engineering in Miskolc in 1949 and the movement of the Sopron Mining and Metallurgical Faculties to Miskolc the Publications restarted with the general title Publications of the Technical University of Heavy industry in 1955. Four new series Series A (Mining), Series B (Metallurgy), Series C (Machinery) and Series D (Natural Sciences) were founded in 1976. These came out both in foreign languages (English, German and Russian) and in Hungarian. In 1990, right after the foundation of some new faculties, the university was renamed the University of Miskolc. At the same time the structure of the Publications was reorganized so that it could follow the faculty structure. Accordingly, three new series were established: Series E (Legal Sciences), Series F (Economic Sciences), and Series G (Humanities and Social Sciences). The seven series are constituted by some periodicals and publications, which come out with various frequencies.

PRODUCTION SYSTEMS AND INFORMATION ENGINEERING
Volume 7 (2015)

CONTENTS