J. Software Engineering & Applications, **2010, 3**
Published Online February 2010 in SciRes(www.SciRP.org/journal/jsea)

CONTENTS

Volume 3 Number 2 **February 2010**

Journal of Software Engineering and Applications (JSEA)

Journal Information

SUBSCRIPTIONS

The *Journal of Software Engineering and Applications* (Online at Scientific Research Publishing, www.SciRP.org) is published monthly by Scientific Research Publishing, Inc., USA.

E-mail: jsea@scirp.org

Subscription rates: Volume 3 2010

Print: $50 per copy.

Electronic: free, available on www.SciRP.org.

To subscribe, please contact Journals Subscriptions Department, E-mail: jsea@scirp.org

Sample copies: If you are interested in subscribing, you may obtain a free sample copy by contacting Scientific Research Publishing, Inc. at the above address.

SERVICES

Advertisements

Advertisement Sales Department, E-mail: jsea@scirp.org

Reprints (minimum quantity 100 copies)

Reprints Co-ordinator, Scientific Research Publishing, Inc., USA.

E-mail: jsea@scirp.org

COPYRIGHT

PRODUCTION INFORMATION

For manuscripts that have been accepted for publication, please contact:

E-mail: jsea@scirp.org

J. Software Engineering & Applications, 2010, 3: 99-108
doi:10.4236/jsea.2010.32013 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A New Algorithmic Approach for Detection and Identification of Vehicle Plate Numbers

A. Akoum[1], B. Daya[1], P. Chauvet[2]

[1]Lab. GRIT, Institute of Technology, Lebanese University, Saida, Lebanon; [2]CREAM/IRFA, Institute of Applied Mathematics, UCO, Angers, France.
E-mail: alhussain.akoum@etud.univ-angers, b_daya@ul.edu.lb, pierre.chauvet@uco.fr

Received November 10th, 2009; revised December 16th, 2009; accepted December 20th, 2009.

ABSTRACT

This work proposes a method for the detection and identification of parked vehicles stationed. This technique composed many algorithms for the detection, localization, segmentation, extraction and recognition of number plates in images. It is acts of a technology of image processing used to identify the vehicles by their number plates. Knowing that we work on images whose level of gray is sampled with (120×180), *resulting from a base of abundant data by PSA. We present two algorithms allowing the detection of the horizontal position of the vehicle: the classical method "horizontal gradients" and our approach "symmetrical method". In fact, a car seen from the front presents a symmetry plan and by detecting its axis, that one finds its position in the image. A phase of localization is treated using the parameter MGD* (*Maximum Gradient Difference*) *which allows locating all the segments of text per horizontal scan. A specific technique of filtering, combining the method of symmetry and the localization by the MGD allows eliminating the blocks which don't pass by the axis of symmetry and thus find the good block containing the number plate. Once we locate the plate, we use four algorithms that must be realized in order to allow our system to identify a license plate. The first algorithm is adjusting the intensity and the contrast of the image. The second algorithm is segmenting the characters on the plate using profile method. Then extracting and resizing the characters and finally recognizing them by means of optical character recognition OCR. The efficiency of these algorithms is shown using a database of* 350 *images for the tests. We find a rate of localization of* 99.6% *on a basis of* 350 *images with a rate of false alarms* (*wrong block text*) *of* 0.88% *by image.*

Keywords*: *Vehicle Detection, Segmentation, Extraction, Recognition Number Plate, Gradient Method, Symmetry Method, Real-Time System

1. Introduction

Automatic identification of the vehicles used, is an effective control, in the automatic check of the traffic regulations and the maintenance of the application of the law on the public highways [1]. The identification of the vehicles is a crucial step in the intelligent transport systems. Today, vehicles play a great part in transportation. The traffic, also, increased because of population growth and human needs during the last years. Consequently, the control of the vehicles is becoming a big problem and much more difficult to solve. Since each vehicle is equipped with a single number plate, not of external charts, the tags or the transmitters must be recognizable at the number plate. Thus, many researches concerning the identification of cars involved the extraction and the recognition of number plate. Some of these works are the following: [2] proposed knowledge-guided boundary following and template matching for automatic vehicle identification. [2,3] rotted to find a presentation of six descriptors, among the most used. [4,5] found the description of the detector. [8] found the description of the Dog detector. [6] used a genetic algorithm ASED segmentation to extract the plate area. [8] used Gabor jets of projection to form a vector characteristic for the recognition of weak grey scale resolution character. [9] proposed a method of extraction of characters without preliminary knowledge of their position and the size of the image.

The general information of the system depends mainly on the general information of the number plate. If one seeks the common characteristics for various types of number plates, one finds one of the fundamental characteristics which are contrast, which means relatively large difference of color or intensity between the signs and the background of the number plate. This fact is often used in many methods of localization of the number plates [10,

12,13], and the problem of localization of text in the image [11,14,15–17].

In our work, we study the processes allowing to detect and identify the plate number, the best possible, in real time system. The database contains images of good quality (high-resolution: 1280×960 resize 120×180) for vehicles seen from the front, more or less near, parked either in the street or in a car park, with a negligible inclination.

To contribute in improving the automatic vehicle plate detection and identification systems, this work presents, in a more robust way, the detection and identification of number plates in images of vehicles stationed.

Automatic number plate recognition is a mass surveillance method that uses optical character recognition on images to read the license plates on vehicles. It can be used to store the images captured by the cameras as well as the text from the license plate, with some configurable to store a photograph of the driver.

We present two algorithms allowing the detection of the horizontal position of the vehicle: the classical method "horizontal gradients" and our approach "symmetrical method" In fact, a car seen from the front presents a symmetry plan and by detecting its axis, we find its position in the image. A phase of localization is treated using the parameter MGD (Maximum Gradient Difference) which allows the detection to locate all the segments of text per horizontal scan. These calculations must take into account the horizontal position of the vehicle so as to remove the segments which are not cut by the axis of detection. The potential segments of text are then widened or combined with possible adjacent segments of text by this line sweep (in the two directions) to form blocks of text, which will make, thereafter, the object of filtering.

The blocks of text obtained are regarded as areas of candidate text. These areas must undergo a specific technique of filtering which makes it possible to find the good block containing the plate number among different blocks obtained by the previous algorithm of detection. After locating the plate, we use four algorithms that must be executed so that our system can read a license plate. The first algorithm will adjust the intensity and the contrast of the image. The second, will segment the characters on the plate using vertical profile method. The third algorithm will extract and resize characters and the Last algorithm will allow optical character recognition OCR. During the adjustment phase, we use techniques of edge detection to increase the contrast between the characters and the background of the plate. Then a specified filter is used to eliminate the noise that surrounds the characters.

The rest of the work is organized as follows: In Section 2, a description of the real dataset used in our experiment is given. We present, in Section 3, the two methods of detection of vehicles: "horizontal gradients" and "symmetrical approach". In Section 4 we describe our approach of localization of the plate registration combining the MGD method with the symmetrical, we give in Section 5 the description of our algorithm which extracts the characters from the license plate, and we conclude in Section 6.

2. Databases

The database (Base Images with License) contains images of good quality (high-resolution: 1280×960 pixels resizes to 120×180 pixels) of vehicles seen of face, more or less near, parked either in the street or in a car park, with a negligible slope. The images being neither in stereophony nor in the form of sequence (video), they were treated consequently with methods not using information which can provide it video (movement, follow-up of vehicle) and stereophony (measurement of depth, validation).

The cameras used can include existing road-rule enforcement or closed-circuit television cameras as well as mobile units which are usually attached to vehicles. Some systems use infrared cameras to take a clearer image of the plates [18–21].

Let us note that in our system we will divide in a random way the unit of our database into two:

1) A base of Training on which we regulate all the parameters and thresholds necessary to the system so as to obtain the best results.

2) A base T is on which we will test all our programs.

2.1 Labeling of the Data

The all database was labeled in order to detect that vehicle which is not partially hidden, only on the vehicles which are in direct link with the vehicle equipped with a camera (potentially dangerous and close vehicle).

2.2 Characteristics of the Image

The images employed have characteristics which limit the use of certain methods. Very of access, the images are in level of gray. What eliminates the methods using spaces of color RGB, HSV or others? Then, the images are isolated, in the direction where they make neither started from a sequence nor of a couple of stereo image. The video can be used to make a follow-up of the vehi-

Figure 1. Some examples from the database training

cles and thus to check detections while stereophony makes it possible to have information of depth. The images have an original size of 1280×960 pixels. (Figure 1)

3. Our Approach for the Detection of the Vehicle Position

The bibliographical study led on the detection of vehicle for help to control, has brought to us a certain number of method and technique imagined for a few years. From this study, our work of development was to create and test some one of these methods.

We limited ourselves mainly to two techniques of detection. These techniques are the following ones:

- Method horizontal gradients.
- Our approach (Method Symmetry).

The goal is to detect the position of the vehicles in order to exclude false alarms, all around the vehicle that we will find in the stage of detection of the plate and to keep only those which are cut by the axis of detection. Let us note that we work with images in level of gray under sampled (120×180), which eliminates the methods using spaces of color RGB or HSV and reduces the execution of time.

3.1 Method Horizontal Gradients

The method uses knowledge in frequent appearance of the front view of a car having horizontal contours and basing on the fact that the horizontal position of the vehicle is where one finds a strong concentration of horizontal gradients [16].

Thus we calculate the profile of horizontal contours (the sum of the horizontal gradients by column) and the presence of a peak in this last makes it possible to go back to the horizontal position of the corresponding vehicle.

3.1.1 The Algorithm of the Method "Horizontal Gradients"

Extraction of horizontal contours: We calculate the image of the horizontal gradients by withdrawing from the original image its shifted copy of a line to the bottom (or upwards). One filters the preceding image by preserving only the pixels belonging to a sufficiently long horizontal segment (here the length was fixed at 11 pixels). This gives the indication of the horizontal gradients to us.

Calculation of the horizontal profile: We calculate the profile of horizontal contours by summoning for each column the horizontal gradients.

Detection of vehicle: By finding the position of the peak in the profile of horizontal contours, one finds the position of the vehicle.

Figures 2 to 7 show an example of detailed execution for our algorithm, allowing the detection of the position of the vehicle.

Figure 2. Initial image

Figure 3. Conversion of the image into level of gray

Figure 4. Subtraction enters the shifted image and itself of a line to the bottom horizontal contours

Figure 5. Image with segment length>11

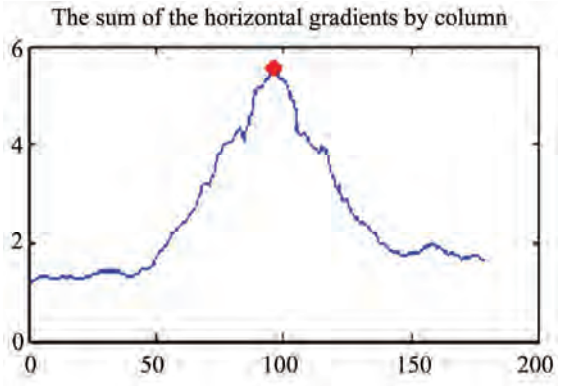


Figure 6. Summon horizontal gradients

Figure 7. Detection of the vehicle position

3.1.2 Experimental Results

The result obtained by the method horizontal gradient on a basis of 350 different images, we found the following results:

- An average execution time/image: 0.25 seconds.
- A maximum error: 49.53%.
- A minimal error: 0.08%.
- An average error: 6.91%.

To evaluate our algorithm, an error is calculated by the equation according to:

$$error = \left(\frac{pos\_fts - pos\_prog}{l} \right) \qquad (1)$$

with:

- l: width of the original image.
- pos_fts: the true horizontal position of the vehicle.
- pos_prog: the position given by the program.

Four examples of detection of vehicle position are presented in Figure 8. We notice that the right position of the vehicle (bold line) is very close to the position of the axis of symmetry (red line) given by Method Horizontal gradients. The error is about 19.63%.

We notes in the Figures 8(c) and 8(d) that bad detection corresponds to vehicles at the bottom of the image and this due to the fact that they have them also horizontal contours which come to influence our method.

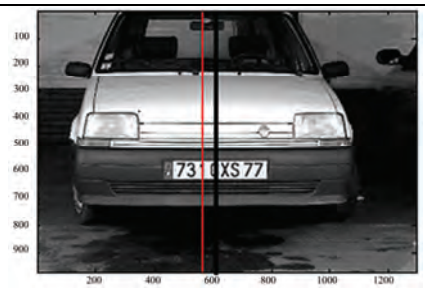
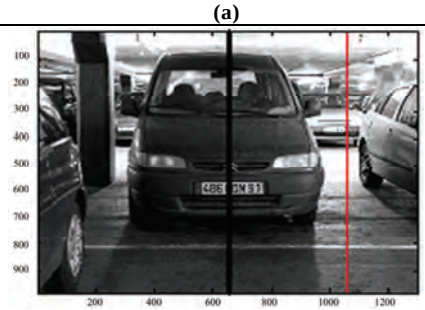
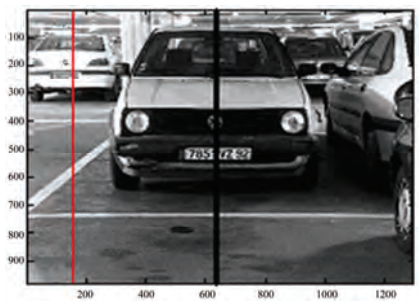
Figures	Error (%)
 (a)	3.17
 (b)	30.61
 (c)	7.45
 (d)	37.31

Figure 8. Shows the comparison between the right position (bold line) of the vehicle and the position of the axis of symmetry (red line) given by the horizontal gradient method

3.1.3 Conclusions

Thus this method, very powerful of concept of time, is found influenced by the entourage of the vehicle (presence of cars, people,…) and cannot be regarded as reliable method considering which the error can sometimes

be so large that the detected object is far from being the vehicle.

3.2 Symmetry

A car seen from the front presents a symmetry plan and in detecting its symmetric axis, we find its position in the image. This axis of symmetry is found by seeking in each line of the image, the pairs of points of the contours which have the same level of gray (with a margin of 25) and then we vote for the point medium in an initially empty matrix. The column of this matrix, having the maximum of votes, corresponds to our axis of symmetry which defines the position of the vehicle.

3.2.1 The Algorithm of Our Method "Method Symmetry"

In this method, we detect the horizontal contour of the image, and then we apply our approach of detection based on the determination of the axis of symmetry, which defines the position of vehicle.

- Detection of contours

- This detection is done by filtering:

By the following masks:

(a)

(b)

Figure 9. (a) Initial image (b) Resized image in level of gray

Figure 10. Detection of horizontal contours

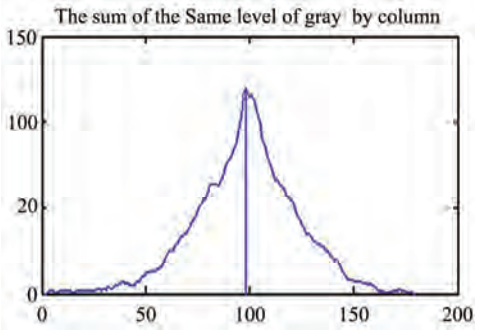


Figure 11. Index of the peak

Figure 12. Detection of vehicle

Horizontal mask: mx=[3 10 0 -10 -3]/32 (2)

Vertical mask: my=[7 63 116 67 7]/256 (3)

-Then of a thresholding.

- Detection of the axis of symmetry

One seeks in each line of the image the pairs of points of contours having the same level of gray (with a margin of 25) and one vote for the point medium in a matrix, initially empty. One makes the sum by column of the matrix and one finds the horizontal position of the maximum of votes which corresponds to our axis of symmetry.

Figures 9 to 12 show an example of detailed execution for our algorithm, allowing the detection of the axis of

symmetry of the vehicle.

3.2.2 Results

On a test basis of 350 different images, one finds:

- An average execution time/image: 0.25 seconds.
- A maximum error2.87%.
- A minimal error: 0%.
- An average error0.34%.
- Examples of detection

It is noted that with this method one always detected well the position of the vehicle independently of his limits. (see as in Figure 13)

3.2.3 Conclusions

Thus this method is reliable and effective being: it is simple, fast and the average error is 20 times smaller than that obtained with the method of the "horizontal gradients" and thus we will continue our work using this method.

4. Algorithm of Localization of the Plate Registration

Our algorithm is based on the fact that an area of text is characterized by its strong variation between the levels of gray and this is due to the passage from the text to the background and vice versa (see Figure 14). Thus by locating all the segments marked by this strong variation, while keeping those which are cut by the axis of symmetry of the vehicle found in the preceding stage, and by gathering them. One obtains blocks to which we apply certain conditions (surface, width, height, the width ratio/height,…) in order to recover the areas of text candidates *i.e.* the areas which can be the number plate of the vehicle in the image.

4.1 Detection of the Segments of Potential Text

This phase consists of elaborating one robust algorithm for the extraction of the text from the image. First, the algorithm calculates the MGD for each line (Maximum Gradient Difference) in order to keep all the segments which have a strong variation of level of gray (segments in which we find a high MGD), and which intersect with the axis of symmetry of the vehicle.

To calculate this MGD, we start by applying a mask [-1 1] to each line of the image. Then, on each site of pixel, the MGD is calculated as being the difference between the maximum and the minimum of values within a local window of size N×1 centered on the pixel. Parameter N depends on the maximum size of the text which we want to detect. A good choice for N is a value which is slightly larger than the width of the greatest character we want to detect. In our algorithm, we chose N=10.

In general, the segments of text have great values of MGD (see Figure 15).

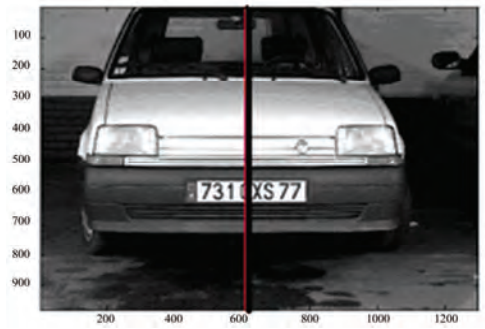
Figure	Error (%)
 (a)	0.13
 (b)	0.72
 (c)	0.81
 (d)	0.15

Figure 13. Show the comparison between the right position (bold line) of the vehicle and the position of the axis of symmetry (red line) given by the symmetrical method

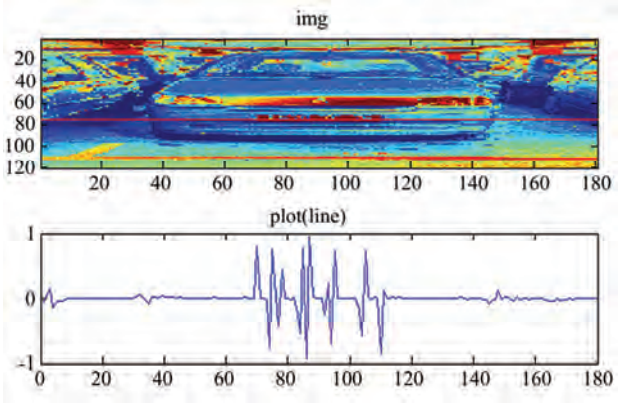


Figure 14. This figure represents the strong variations between the levels of gray and this due to the passage from the text to the background and vice versa

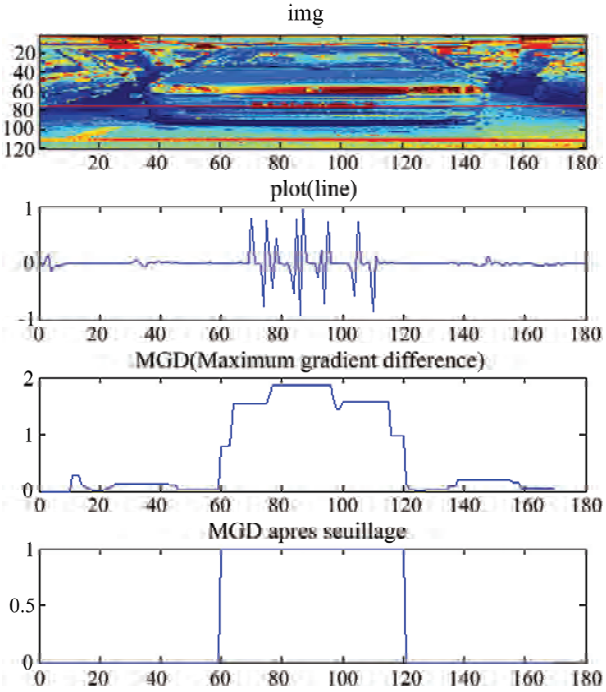


Figure 15. The image above represents the MGD in each column and we notice that there is a big rise in the zone of text (between column 70 and 130)

Then, we identify the potential segments of text by the analysis of each segment in the image, while taking into consideration the fact that the segments, marked by a strong variance between the levels of gray, are likely to contain text.

A segment of text is a continuous segment of a pixel thickness on a segment of line of sweeping made up of pixels representing of the text. In general, a segment of text passes through a character string and contains a continuation of alternation between the pixels of the text and the pixels of background pixels. This is why, if the line, containing some text, is traced, we will have a succession of positive and negative peaks in the interval including the text. These peaks are due to transitions from the text to the background (text-to-background transitions) or from the background to the text (background-to-text transitions). And also by noting that, the magnitude of the peaks corresponding to the text, is definitely larger than that of possible peaks corresponding to other elements in the image. For a segment, containing some text, there must be an equal number of transitions: background-to-text and text-to-background (with a margin of 3), and these two types of transitions must be alternated. In practice, the numbers of transitions (background-to-text and text-to- background) cannot be exactly the same and this is due to the presence of noise, but they must be almost equal. The number of peaks (negative and positive) must also be important in a segment of potential text (here, the threshold is of 5 peaks minimum).

4.2 Detection Various Blocks of Text

In the second phase, the potential segments of text are wide or amalgamated with the segments of text of adjacent lines to form blocks of text. The algorithm functions in two directions: signal-down (from top to bottom) and bottom-up (upwards): In the first direction, the group of pixels immediately in lower part of the segment of potential text is taken into account.

If the average and the variance of its levels of gray are close with those to the segment of potential text *i.e.* if:

$$\left|\mu_1 - \mu_2\right| < \theta \quad Et \quad \left|\sigma_1 - \sigma_2\right| < \theta \qquad (4)$$

with θ*=0.1*

Then, they are amalgamated. This process begins again for the group of pixels immediately in the lower part of the text, recently widened. It is stop when the block of widened text amalgamates with another segment of potential text.

In the second direction, the same process is applied but of way bottom-up with each segment of text obtained in first direction.

The isolated segments and the too long segments are removed (segment of width higher than 0.75 of that of the image). We obtain then many blocks of text with variable forms (Figure 16).

4.3 Calculation of Their Limp Including

For each block of detected text, we calculate its corner, including characteristics the four following parameters: x, y for the corner position, w for the width and h for the height (Figure 17).

Let us note that we rearrange the coordinates of this corner by making a sweep field (beginning towards the end then the fine one towards the beginning) as long as

Figure 16. Detection of plate numbers (without filtering)

Figure 17. Detection of the block of text having the same characteristic of the plate number (after filtering)

less than 40% of the pixels are white.

4.4 Filtering of the Blocks of Text

In the last phase, we filter each block detected, in function of:

1) Its geometrical properties which are:

a) Width/height ratio must be higher than 5.

b) Surface must be higher than 60 pixels.

c) Width must be lower than 60% of the width of the image.

2) Its contents:

After a numerical representation of each block, we calculate the ratio between the number of white pixels and that of the black pixels (minimum/maximum). This report/ratio corresponds to the proportion of the text on the block which must be higher than 0.15 (the text occupies more than 15% of the block).

In the experimental results of the entire process, we found a rate of detection of 99.6% on a basis of 350 images with a rate of false alarms (wrong block text) of 0.88% by image.

5. License Plate Characters Extracting

The block of the plate detected in gray (Figure 18) will be converted into binary code, and we construct a matrix with the same size block detected. Then we make a histogram that shows the variations of black and white characters [22].

To filter the noise, we proceed as follows: we calculate the sum of the matrix column by column, and then we calculate the min_sumbc and max_sumbc representing the minimum and the maximum of the black and white variations detected in the plaque. All variations which are less than 0.08 * max_sumbc will be considered as noises. These will be canceled facilitating the cutting of characters. (Figure 19)

To define each character, we detect areas with minimum variation (equal to min_sumbc). The first detection of a greater variation of the minimum value will indicate the beginning of one character. And when we find again another minimum of variation, this indicates the end of the character. So, we construct a matrix for each character detected. (Figure 20)

The Headers of the detected characters are considered as noise and must be cut. Thus, we make a 90 degree rotation for each character and then perform the same work as before to remove these white areas. (Figure 21)

Figure 18. Extracting of license plate

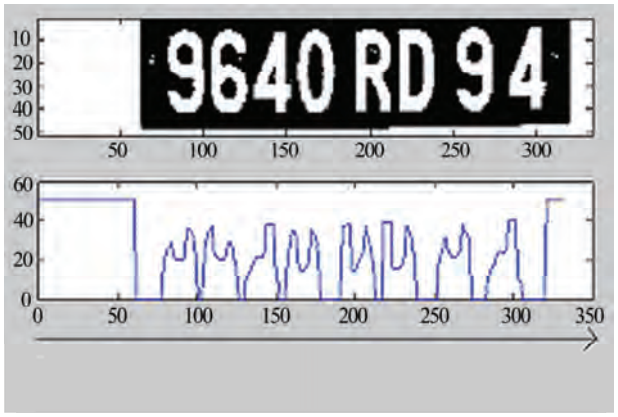


Figure 19. Histogram to see the variation black and white of the characters

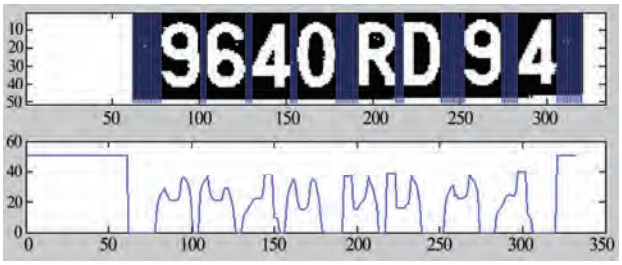


Figure 20. The characters are separated by several vertical lines by detecting the columns completely black

Figure 21. Extraction of one character

Figure 22. Rotation 90 degrees of the character

Figure 23. Representation of the histogram of the alphanumeric chain and extraction of a character from the number plate

A second filter can be done at this stage to eliminate the small blocks through a process similar to that of extraction by variations black white column. (Figure 22)

Finally, we make the rotation 3 times for each image to return to its normal state. Then, we convert the text in black and change the dimensions of each extracted character to adapt it to our system of recognition. (Figure 23)

6. Conclusions

In this paper, we presented a system of detection of the position of vehicles and the localization of their numbers plates by two robust algorithms: the classical "horizontal gradients" and our approach "symmetrical method". Then we used four algorithms for the normalization, segmentation, extraction and recognition. In our system, we apply the symmetrical approach which allows finding the symmetrical axis which will determine the position of the vehicle in the image. Then we proposed an approach of localization for the plate, which calculates the MGD (Maximum Gradient Difference) and detects the potential segments of text per horizontal scan. The blocks of texts obtained are regarded as areas of text candidates. These areas must undergo a specific technique of filtering which makes it possible to find the good block containing the numbers plates among the various blocks obtained by the previous algorithm. After detecting the plate, we use four algorithms: normalization (adjustment of the intensity and the contrast of the image), segmentation of characters (separation of character on the plate), and extraction (removal each character of the plaque) and recognition (optical identification of characters by OCR).

We tested our approach on a group of 350 images of vehicles seen from the front (taken in real situation). We obtained very encouraging results, in fact: a rate of detection of 99.6% with a rate of false alarms (wrong block text) of 0.88% by image. Moreover, the results also showed that the system is robust with respect to occlusions partial of the image.

This work could have several continuations like: 1) An adaptation to all kinds of number plates (universal dimensioning and flexible device); 2) A phase of recognition of the characters in the detected blocks of text. Thus, we could conceive our own system of automatic reading of the number plates.

7. Acknowledgements

The authors would like to thank Dr. Lionel PROVEST for his contribution to this work. This research was supported by the CEDRE project (07SciF29/L42).

REFERENCES

[1] D. G. Bailey, D. Irecki, B. K. Lim, and L. Yang, "Test bed for number plate recognition applications," Proceedings of First IEEE International Workshop on Electronic Design, Test and Applications (DELTA'02), IEEE Computer Society, 2002.

[2] E. R. Lee, P. K. Kim, and H. J. Kim, "Automatic recognition of a car license plate using color image processing," Proceedings of the International Conference on Image Processing, 1994.

[3] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool, "A comparison of affine region detectors," International Journal of Computer Vision, 2006.

[4] "Detecting salient blob-like image structures and their scales with a scale-space primal sketch: A method for focus-of-attention," International Journal of Computer Vision, 1993.

[5] T. Lindeberg, "Scale-space theory: A basic tool for analyzing structures at different scales," Journal of Applied Statistics, 1994.

[6] S. K. Kim, D. W. Kim, and H. J. Kim, "A recognition of vehicle license plate using a genetic algorithm based segmentation," Proceedings of 3rd IEEE International Conference on Image Processing, Vol. 2, pp. 661–664, 2006.

[7] D. G. Lowe, "Distinctive image features from scale-invariant key points," IJCV, 2004.

[8] H. Yoshimura, M. Etoh, K. Kondo, and N. Yokoya, "Gray-scale character recognition by Gabor jets projec-

tion," Proceedings 15th International Conference on Pattern Recognition, ICPR, IEEE Computer Society, Los Alamitos, USA, Vol. 2, pp. 335–8, 2000.

[9] H. Hontani and T. Koga, "Character extraction method without prior knowledge on size and information," Proceedings of the IEEE International Vehicle Electronics Conference (IVEC'01), pp. 67–72, 2001.

[10] H. Kwaśnicka and B. Wawrzyniak, "License plate localization and recognition in camera pictures," Gliwice, Poland, November 13–15, 2002.

[11] E. K. Wong and M.Chen, "A new robust algorithm for video text extraction: Pattern recognition," pp 1397–1406, 2003.

[12] S. Ozbay and E. Ercelebi, "Automatic vehicle identification by plate," Proceedings of Word Academy of Science Engineering and Technology, ISSN, Vol. 9, pp. 1307–6884, November 2005.

[13] N. Vazquer, M. Nakano, and H. perez Meana Autom, "Automatic system for localization and recognition of vehicle plate number," Journal of Applied Research and Technology, Vol. 1, pp. 63–77, 2003.

[14] L. M. B. Claudino, "Text fragments segmentation for license plate location," Master's Thesis (in Portuguese), Graduate Program of Electrical Engineering of Universidade Federal de Minas Gerais–PPGEE/UFMG, Belo Horizonte-MG, Brazil, 2005.

[15] L. M. B. Claudino, A. de P. Braga, and A. de A. Ara_ujo, "Text fragments segmentation for license plate location," (in Portuguese), CD-ROM Proceedings of the Workshop of Theses and Dissertations on Computer Graphics and Image Processing of the IEEE.

[16] A. Khammari, F. Nashashibi, Y. Abramson, and C. Laurgeau, "Vehicle detection combining gradient analysis and adaboost classification," In 8th International IEEE Conference on Intelligent Transportation Systems, Vienna, Austria, pp. 66–71, September 2005.

[17] A. Cornuéjols et L. Miclet. Eyrolles, "Apprentissage par combinaison de décisions," XIX Brazilian Symposium on Computer Graphics and Image Processing-WTDCGPI/ SIBGRAPI, Manaus-AM, Brazil, Extended (English) version to appear in the Brazilian Journal of Theoretical and Applied Computing, 2006.

[18] http://www.photocop.com/recognition.htm.

[19] http://vortex.cs.wayne.edu/papers/ijns1997.pdf.

[20] http://www.siemenstraffic.com/customcontent/case_studies/anpr/anpr.html.

[21] http://www.be.itu.edu.tr/Ekahraman/License plate character segmentation based on the Gabor transform and vector quantization.pdf.

[22] H. Hontani and T. Kogth, "Character extraction method without prior knowledge on size and information," Proceedings of the IEEE lnternational Vehicle Electronics Conference (IVEC'01), pp. 67–72, 2001.

J. Software Engineering & Applications, **2010, 3: 109-118**
doi:10.4236/jsea.2010.32014 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Scientific Research

The 3+1 SysML View-Model in Model Integrated Mechatronics

Kleanthis Thramboulidis

Visiting Professor Helsinki University of Technology, Electrical & Computer Engineering, University of Patras, Patras, Greece.
Email: thrambo@ece.upatras.gr

Received October 21st, 2009; revised November 6th, 2009; accepted December 1st, 2009.

ABSTRACT

Software is becoming the driving force in today's mechatronic systems. It does not only realize a significant part of their functionality but it is also used to realize their most competitive advantages. However, the traditional development process is wholly inappropriate for the development of these systems that impose a tighter coupling of software with electronics and mechanics. In this paper, a synergistic integration of the constituent parts of mechatronic systems, i.e. mechanical, electronic and software is proposed though the 3+1 *SysML view-model. SysML is used to specify the central view-model of the mechatronic system while the other three views are for the different disciplines involved. The widely used in software engineering V-model is extended to address the requirements set by the* 3+1 *SysML view-model and the Model Integrated Mechatronics* (*MIM*) *paradigm. A SysML profile is described to facilitate the application of the proposed view-model in the development of mechatronic systems.*

Keywords**:** *Systems Engineering, System Modeling, Mechatronic Component, Model Driven Development, Model Integrated Mechatronics, SysML Profile, V-Model, IEC*61499

1. Introduction

Software does not only implement a significant part of the functionality of today's mechatronic systems, but it is also used to realize their most competitive advantages. It is the evolving driver for innovations in many mechatronic systems and in general it is considered as the driving force in improving this kind of systems. However, the traditional development process is wholly inappropriate for the development of systems characterized by complexity, dynamics and uncertainty as is the case with today's mechatronic systems [1]. According to the traditional development process the constituent parts of the mechatronic system, i.e. the mechanical, electronic and software, that constitute the system are developed independently and then are integrated to compose the final system. The software development starts when the development of electronic and mechanical is already at a stage where any change in these parts is expensive and time consuming. This is why the mechanical and electronic properties impose several constrains and narrow the solution space for software development. Moreover, as claimed in [2] "the actual cooperation during the construction is less developed. There is no joint development process, no joint tool usage, no joint modeling formalism and no joint analysis. Every discipline has its own approaches". As a result of this, the current process that is traditionally divided into software, electronics and mechanics, emphasizes on domain-spanning design methods and tools and is unable to address the demand for synergetic mechatronic dependability predictions; this is why many products suffer from severe dependability problems [3].

An integrated framework for the construction of mechatronic systems is missing [2]. Such a framework should address current challenges in the development of mechatronic systems that include among others, synergistic modeling and integration, design synchronization, as well as model execution and analysis. It should provide the infrastructure required for applying a tight integration of mechanics with electronics and software in order to replace conventionally designed mechanical and electromechanical systems into smart ones where significant part of functionality will be implemented by software. It is also expected to result in massive improvements in system's Quality of Service (QoS) characteristics and allow a smooth integration of dependability predictions during the early development phases.

Model Integrated Mechatronics (MIM) [4] is a paradigm that was proposed to address the need for an integrated development in mechatronic systems. MIM supports the model-driven development of complex mecha-

tronic systems (MTSs) through the evolution of models that have as primary construct the mechatronic component (MTC). The concept of component has already been adopted in the development process of manufacturing systems by other researchers too. However, they mostly focus on the software part of the component; they do not address the whole development process; and they do not provide an architecture for the concurrent engineering of all constituent components, i.e., mechanical, electronic and software.

In this paper, the Systems Modeling Language (SysML) [5] is adopted for the system's modeling process in the MIM paradigm. An architectural view-model, the one called 3+1 SysML is proposed to address the synergistic integration of the constituent parts of mechatronic systems. The main view is the SysML view that corresponds to the mechatronic layer of the MIM Architecture. This view captures the system model that is the one constructed by the MTS developer. Each of the three views is used to describe the system from the perspective of the corresponding discipline. Specific tools of every discipline may be exploited for the model execution and analysis of the SysML models.

SysML is used to represent the models of the mechatronic system that are proposed by the MIM paradigm. A SysML profile was defined using Papyrus, an open source tool for graphical UML 2/SysML modeling. This profile supports not only the modeling of mechatronic systems using the concept of mechatronic component but also a hybrid development process that integrates the traditional approach with the MTC-based one, to allow the reuse of legacy systems. It provides a SysML based implementation of the MIM architecture that will allow the MIM paradigm to be exploited in the development process of real world mechatronic systems. An effective development process that exploits the 3+1 SysML view-model and the SysML profile is described extending the well-known in software engineering V-model [6].

The remainder of this paper is organized as follows. In the next section, a brief introduction to the MIM architecture and the SysML is given and the related work is briefly discussed. Section 3 presents the proposed 3+1 SysML view-model that emphasizes the importance of the common system-level model. In Section 4, a SysML profile that represents the artifacts used in the MIM architecture is described. Section 5 presents the proposed modifications to the widely used in software systems V-model to address the needs of the MTS development process. Future developments and research challenges are discussed in Section 6 and the paper is concluded in the last section.

2. Background and Related Work

2.1 A Brief Introduction to MIM

The upper layer of the MIM architecture that is shown in Figure 1, i.e. the mechatronic layer, was defined to systematically address complexities in the model-driven development process of component-based mechatronic systems. This layer is projected into three dimensions representing the application, the resource, and the mechanical process respectively. The controlling application software is modeled in the application layer, while the hardware, i.e., computing and communication, as well as the software resources that constitute the infrastructure required for the execution of the controlling application software, are modeled in the resource layer. Mechanical, hydraulic and pneumatic parts are modeled in the mechanical layer.

Mechatronic system integrators work horizontally in the model evolution dimension of the MIM architecture. They interactively compose the MTS using already defined MTCs without worrying on lower layers' implementation details. They go through a model-driven development process to build the MTS using descriptions of already existing MTCs. They only have to capture the application logic in application layer components, as well as to identify their required QoS characteristics from the resource layer infrastructure.

MTC builders work in the model integration dimension and apply an information integration process that crosses the boundaries between mechanical, electronic, and computer science fields. They work horizontally and vertically, either top-down or bottom-up, in the lower three layers of the architecture in a concurrent way. Constructed MTCs are stored in MTC repositories to be discovered and used by mechatronic system integrators [7].

MIM is a new paradigm that promotes model integration not only of implementation space artifacts but also of early analysis and design phase ones. It promotes

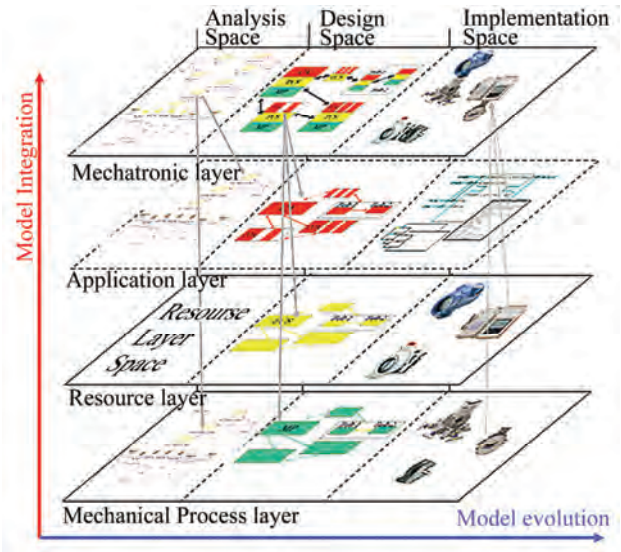


Figure 1. The Model Integrated Mechatronics (MIM) architecture [4]

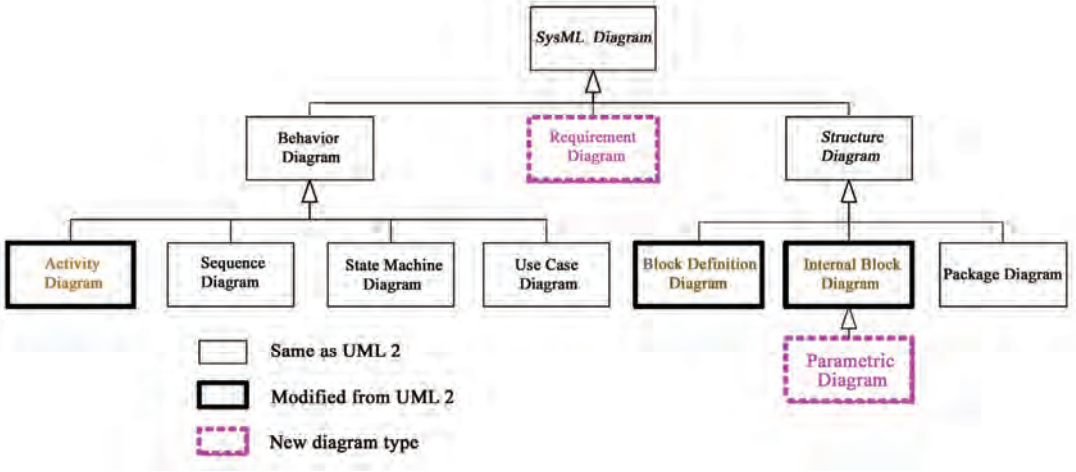


Figure 2. SysML diagrams and its comparison with UML 2 [5]

reuse at the mechatronic level and significantly decreases development and validation time of the system. MIM addresses the need for synergistic integration at the model and process levels; it facilitates the integration between the design processes of the different disciplines which is the approach considered as the most effective to improve the development process of mechatronic systems. The other two approaches being: implementation time integration and design time integration at data and model level [8].

2.2 Systems Modeling Language (SysML)

SysML was developed to support the specification, analysis, design, verification and validation of a broad range of complex systems [5]. These systems may include hardware, software, information, processes, personnel, and facilities. The objective of SysML is to unify the diverse modeling languages currently used by system engineers. SysML reuses a subset of UML 2 [9] and provides additional extensions needed to address system engineering aspects not covered by UML 2. It includes diagrams that can be used to specify system requirements, behavior, structure and parametric relationships. Requirements diagram and parametric diagrams are the new diagram types proposed by SysML, as shown in Figure 2 which presents the diagrams that are used by SysML.

SysML provides modeling constructs to represent text-based requirements and relate them to other modeling elements. The requirement diagram can be used to represent many of the relationships that exist between requirements and visualize them. It provides a bridge between traditional requirements management tools and the other SysML models. It can be used to depict the requirements in graphical, tabular, or tree structure format and highlight their relationships, as well as to capture the relationships between requirements and other model elements that satisfy or verify them.

The other new diagram, i.e. the parametric diagram, is used to describe the constraints among the properties associated with blocks. It allows the specification of continuous components by parametric constraints on class attribute values expressing corresponding differential equations. However, the syntax and the semantics of behavioral descriptions captured in parametric diagrams have not been defined to allow the integration with other simulation and analysis modeling techniques for the proper execution of the models. So, the parametric diagram is used to integrate the system descriptive behavior and structure models expressed in SysML with other simulation and engineering analysis models such as performance, reliability, and mass property models.

The fact that SysML is based on UML 2, will allow system engineers modeling with SysML and software engineers modeling with UML 2 to collaborate on models of the mechatronic system. This will improve communication among the various stakeholders who participate in the mechatronic systems development process and will promote interoperability among modeling tools in different disciplines. All the above mentioned characteristics make SysML ideal for the representation of models used in the MIM paradigm.

2.3 Related Work

Several researchers are already working in the direction of improving the effectiveness of the development process of mechatronic systems. Schafer and Wehrheim [2] survey on current developments in mechatronics and present the architecture of their mechatronic rail system that seems to provide an excellent platform for studding and analyzing future developments and research challenges in mechatronic systems. They identify the need of an integrated framework for the construction of mechatronic systems and they discuss future trends in mechatronics especially from the software engineering

point of view. Habib [10] argues on the urgent need for theories, models, and tools that should facilitate modeling, analysis, synthesis, simulation, and prototyping of mechatronic systems. He emphasizes the argument that the approach based on optimization within each domain separately will not result in the optimum system design, and he proposes a data and model integration approach to address the integration problem. Burmester *et al*. [11] claim that in today's mechatronic systems most of the control and reconfiguration functionality is realized in software. They present mechatronic UML to exploit the Model Driven Architecture approach for the design of hybrid mechatronic real-time systems that have to fulfill safety-critical requirements. "Mechatronic UML" is defined as an extension of UML to built platform independent models for mechatronic systems. Various UML models have been extended to cover the requirements of modeling the structural view as well as the behavioral view of the system. However, the proposed extension is used to model only the software part of the mechatronic system. Authors in [12] briefly refer to a process model of Robert Bosch GmbH for the development of mechatronic systems in Motor Vehicles to support aspects such as reuse, exchangeability, scalability and distributed development. They use the concept of mechatronic component, even though not well defined, as the basic construct of their process. They argue: a) on the need of a clear specification of component interfaces; and b) the great contribution of re-use to increase the quality properties of mechatronic systems and decrease development time. Nordmann [13] is using the concept of mechatronic component and presents an example of using Active Magnetic Bearings to increase performance, reliability, reusability and longer lifetime. Authors in [14] propose for the development of multidisciplinary systems, such as mechatronics, the integration of the various domain-specific tools. They mainly focus on the integration of used data and models and not on a process level integration. Moreover, none of these approaches provide a high level architecture for an integrated, synergistic development process for mechatronic systems and they do not describe a systems level development process based on the mechatronic component and the emerging standard in the domain that is SysML.

3. The 3+1 Architectural View Model

Each of the three lower layers of the MIM architecture provides a specific view of the central models that are captured in the upper layer, i.e. the mechatronic layer. Each view is used to describe the system from the perspective of the corresponding discipline. The software view (s-view), for example, provides the models of the software part of the MTS and allows for software specific tools to be used to elaborate and further refine these models. The IEC61499 function block model is an example of such a domain specific model that can be used to further refine the s-view [15]. Figure 3 depicts the 3+1 SysML view-model that is proposed for the development of mechatronic systems. The MTS model is the heart of this architecture and is depicted in the center of the picture. It is surrounded by 4 views which correspond to the roles that engineers play during the development process of mechatronic systems.

The main view is the MTS-view that corresponds to the mechatronic layer of the MIM Architecture. This is the view that is used by the MTS developer. The other 3 views correspond to the 3 lower layers of the MIM architecture. The m-View, for example, corresponds to the Mechanical layer of the MIM architecture and captures all the mechanics, hydraulics and pneumatics of the MTS model. These models are generated by projecting the MTS models to the mechatronic layer and are fully synchronized with the MTS models. Any modifications imposed by the mechanical engineer to the models of this view directly affect the corresponding central MTS models. Moreover, modifications done by the MTS developer on the central models directly affect the corresponding m-view models. The m-view is mainly used during the primitive MTC development process where a concurrent, synergistic engineering on the three views is adopted at the primitive MTC level as a more effective process. It is also used to have a whole view of the mechanical system model and perform optimization and analysis activities on this.

Figure 4 presents the structure of the primitive MTC and its interfaces to the environment. A primitive MTC may expose to the environment mechanical, electronic and software interfaces through the corresponding ports. Interfaces between its constituent parts are also shown. Sensors and actuators are used to realize the interactions

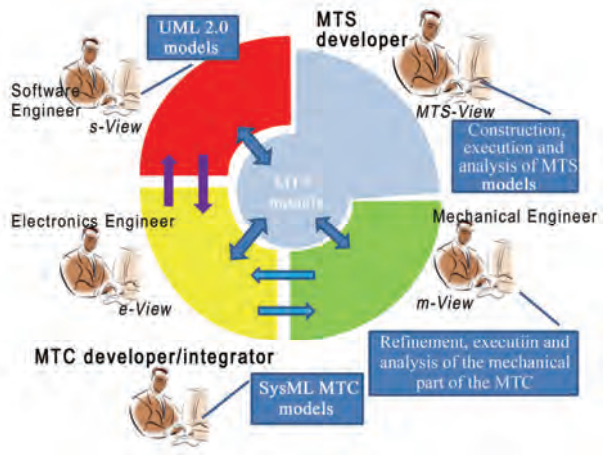


Figure 3. The 3+1 SysML view-model for mechatronic systems development

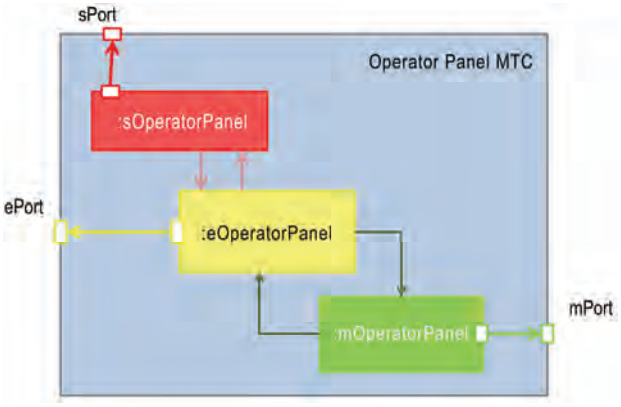


Figure 4. The structure of the primitive mechatronic component

between the mechanical part and the electronic one. This means that sensors and actuators are fully encapsulated by the MTC construct. The mechanical part interacts with the environment only through mechanical ports that are mainly in the form of mounting ports or ports for flow of material or energy. A detailed description of mechanical connections that can be discriminated into fixed and moveable arrangements is given in [16]. The sOperatorPanel, i.e. the software part of the OperatorPanel primitive MTC, exposes its functionality along with the corresponding QoS through a provided interface of the sPort. A hosting functionality for application specific components will be optionally provided by the ePort.

Currently there is no tool to execute the MTS models, not even to analyze their behavior. Discipline specific tools may be exploited for the execution and analysis of MTS models. This is obtained by the proper integration and coordination of specific model execution and analysis tools of the three views. The tool of each view is used to execute the primitive MTC model of its perspective so it has to provide specific interfaces to the tools of the other views, in order to implement the interactions of its own part to the other parts of the primitive MTC. The arrows that cross the boundaries between the three views in Figure 3 represent the interactions of the corresponding models and have to be implemented by specific interfaces of the model refinement, analysis and execution tools of the three disciplines. The AP233 or more formally the ISO 10303-233 standard for systems engineering [17], that provides a data exchange format for the reliable interchange of data between software tools may be exploited to effectively implement these interactions. The execution and analysis of the primitive MTC is obtained through a collaboration of the corresponding tools of the three views. It is clear that the contribution of the three views' specific tools is restricted at the primitive MTC internal level while the execution and analysis of MTC models is done at the MTS level with the coordination and synchronization between MTCs carried by this level. This makes the tool integration a major challenge in the domain of mechatronic systems.

The 3+1 SysML view-model when used with the MTS V-Model that is described in one of the following sections, promote the synergistic integration of the three constituent parts of the mechatronic system and emphasizes the importance of a common model for the system. However, this model can also be used with the traditional development process that is based on the independent development of constituent parts of the mechatronic system and their subsequent integration. Even in this case the existence of a common model for the system greatly improves the effectiveness of the development process.

It should be noted that in each view the corresponding discipline's specific architectures and tools may be exploited, as for example the 4+1 architectural view [18] that may be exploited in the context of the e-view by the software engineer or the MTC developer/integrator.

4. Using SysML to Model the MIM Artifacts

The Systems Modeling Language can be used to represent the artifacts of the mechatronic systems development process that correspond to the system level activities. These include requirements specifications for the MTS and MTC levels, as well as architectural specifications for the MTS and MTC levels. UML 2.0 will be used for the modeling of the software part of the primitive MTC and corresponding tools from the electronics and mechanics domain will be used for the modeling of the other two constituent parts of the MTC. In this section the modeling of the MTS and MTC levels using SysML is considered. It is evident that specific interfaces have to be defined for the integration of the different views and these interfaces have to be realized by the tools used in the various disciplines to create a completely integrated tool chain to support the MTS development process. The SysML to AP233 mapping [19] is towards this direction.

4.1 Modeling of the Mechatronic Component

The MTS stereotype that is shown in Figure 5, which presents part of the SysML4MIM profile, is considered as a composition of MTSComponents and MTSConnectors (not shown in the figure). The abstract stereotype MTSComponet was defined to provide more flexibility in system modeling. It allows the definition of the different disciplines' components in any level of the system's decomposition hierarchy; it also allows the application of the traditional approach where the system is considered as consisting of mechanical, electronic and software components. An MTSComponent that is abstract is specialized to the MTC abstract stereotype and the mCompoment, eComponent and sComponent stereotypes. The mComponent stereotype is used to represent

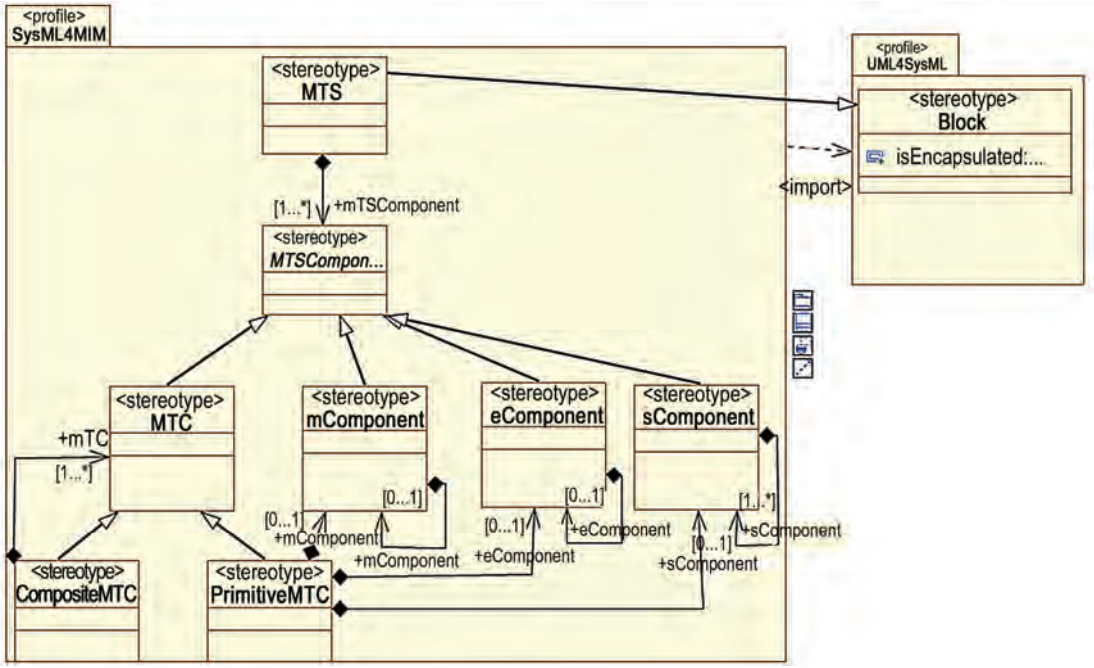


Figure 5. SysML4MIM profile (part); the MTS stereotype

in the model any mechanical component of the mechatronic system. This way of modeling allows the profile to be used also in the traditional development process of mechatronic systems since the MTS may be considered as a composition of m, e and sComponents. The MTC stereotype which is also abstract is specialized to the CompositeMTC and the PrimitiveMTC. This allows a hierarchical decomposition scheme for the MTS up to the level of primitive MTC that is considered as composition of m, e and sComponents. An m, e and sComponent may be further decomposed in corresponding Components allowing a component based synthesis in each one of the three disciplines. The SysML4UML profile that was created using Papyrus, an open source tool for graphical UML 2 modeling (http://www.papyrusum-l.org/), imports the UML4SysML profile that is already supported by Papyrus. This allows the MTS stereotype to extend the Block stereotype of the UML4SysML profile. All the other components also extend the Block stereotype even not shown in figure. The proposed SysML4MIM profile allows the synergistic integration in the development of mechatronic systems to any level of granularity down to the primitive MTC component which is the one that is not decided or it is not possible to be decomposed into lower layer MTC components.

4.2 Modeling of the Mechatronic Port

Allowable inputs and outputs of an MTC are defined using the concept of the port. This allows the design of modular reusable MTCs, with clearly defined interaction points and interfaces with the environment. The construct of Mechatronic port (MTPort) was defined as an extension of the UML port to fulfill this requirement. SysML provides standard ports which support client-server communication and FlowPorts that define flows in and/or out of a block. An MTC may own MTPorts, as shown in Figure 6, which allows the MTC to declare the items it may exchange with its environment and the interaction points through which this exchange is made. Furthermore, MTPorts allow the MTC to declare the provided to the environment services but also the services that the MTC expects from it. An MTPort is defined as an aggregation of mPort, ePort and sPort. Each port is used to represent the interaction point of the corresponding part of the primitive MTC with the environment (see Figure 4). All these ports extend the SysML port; mPort and ePort extend it through the SysML flow port while sPort extends it directly. So, a sPort is characterized by provided and required interfaces. The specification of what can flow in or out of an mPort or ePort is achieved by typing them with a specification of the things that flow in and/or out. It should be noted that an mPort may accept or transmit energy or material but may also accept or transmit information that has been decided to be transferred by mechanical means. Of course the same information may be transferred by electronic signals using an ePort or by software messages using a sPort. The support of several alternatives through configuration, results in increased reuse potential for the MTC. The specification of the services of the sPort is achieved by typing it with the

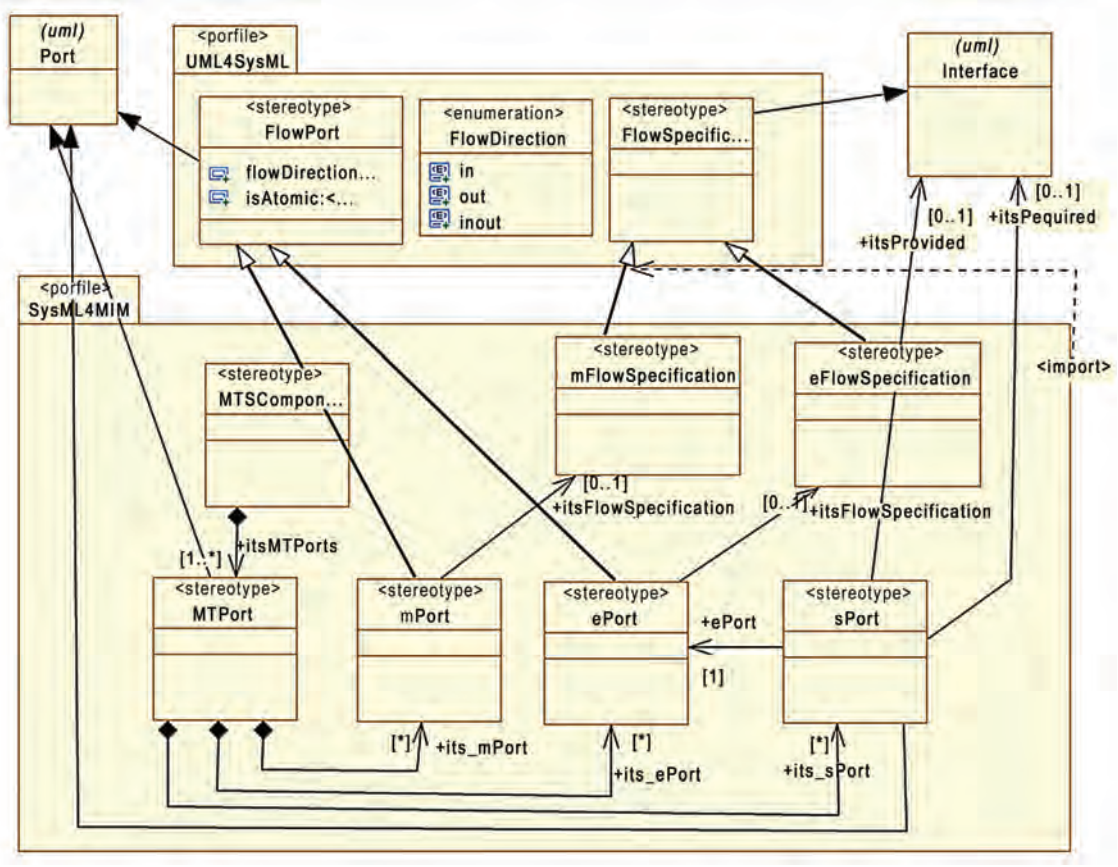


Figure 6. The MTPort stereotype in the SysML4MIM profile

provided and/or required interfaces. Flows of mPorts and ePorts may be atomic or non atomic; an atomic flow is specified with a single type representing the items that flow in or out. A non atomic flow is specified with a flow specification which lists the items that constitute the flow. A sPort accepts software signals, *i.e.* packages of information, which usually need a more complicated specification supported by UML 2.

5. The MTS Development Process

The MIM development process adopts the V-model as basis and updates it to address the needs of the mechatronic systems domain. Figure 7 presents the proposed MTS V-Model. A system modeling process is applied down to the primitive MTC level, as shown in the left-hand part of the V-model. For primitive MTCs that have to be constructed, a concurrent engineering process of all three constituent parts, i.e., mechanics, electronics and software is adopted, as depicted in the bottom of the V-Model. The system integration and verification process is depicted by the right-hand side of the V-model.

MTS-level requirements are captured using the SysML requirements diagram. Essential use cases, which are used to capture the functional requirements at this level, are defined in abstract, simplified, and independent of technology or implementation way. They are written as "an abstract dialog representing user intentions and system responsibilities, and they are typically small and focused on a highly specific user goal, yielding a fine-grained model of user activity" [20]. After the definition of the essential use cases there are two alternatives to proceed in the system's architecture definition phase:

1) Use cases are decomposed in sub-use cases.

2) Responsibilities of the system are identified.

In the first case the decomposition of use cases to sub-use cases allows: a) the reuse of existing components on the basis of their requirement specifications that should have been defined in terms of use cases, and b) modularity and reuse in requirements specification artifacts. In the second case, activity diagrams are defined for each use case in order to identify the activities/responsibilities of the actors and the ones that are required by the system in the context of the specific use case. After this step the list of abstract activities (functions) that have to be performed by the system is available. In other words the responsibilities of the system in

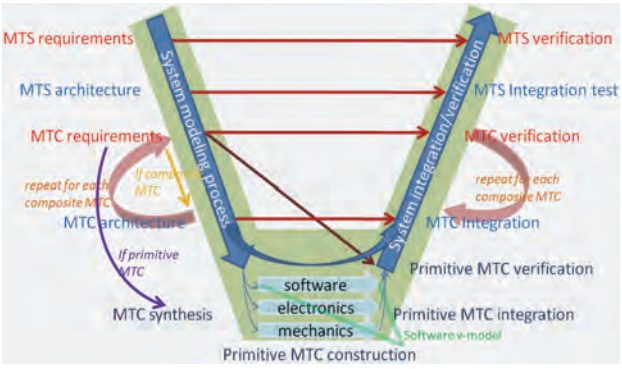


Figure 7. The mechatronic system V-Model (MTS V-Model)

the context of the specific use case are defined. The term responsibility is used to emphasize the fact that only the abstract definition of the activity/function is provided at this time and not its implementation.

As a next step, for both alternatives, a system composition should be proposed to satisfy either the system use-cases or the system responsibilities. Some use cases or system responsibilities may be directly supported by existing mechatronic components. However, it is common that for a required system level use case or responsibility, a collaboration of MTCs has to be defined in order to achieve it. The system level use cases and sub-use cases or the system responsibilities/functions correspondingly are mapped to system's components. An analogous mapping applies also for the non-functional requirements.

The definition of system's structure in terms of MTCs is a design process and results in the selection of system components and the definition of their collaboration. Furthermore each MTC has assigned responsibilities that are handled in the subsequent phases as its required responsibilities. The result in both cases is a system architecture that is comprised of:

a) Class or component diagrams to specify the structure of the system;

b) Sequence or activity diagrams (or even state charts) to specify the components' collaborations to provide the higher layer functionality.

Domain analysis is used to capture the domain key concepts and provide the information required to create the first architectural model of the system. SysML diagrams are used to specify the proposed architecture. Block definition diagrams (bdd) are used to capture the structure of the system and internal block diagrams (ibd) are used to capture the components' interactions, all expressed using the SysML4MIM profile.

During the architecture definition the developer has to assign the system required responsibilities to the system's components. This assignment results to an architectural diagram that represents the system components, their responsibilities and the components interactions. The allocation relationship of SysML provides an effective means to capture this assignment and allow the navigation between the system models by establishing cross-cutting relationships among them. There are two alternatives to proceed in the definition of the architecture:

1) The bottom-up approach (synthesis).

2) The top-down (decomposition).

According to the bottom-up approach, for every system-level responsibility a set of commercial off-the-shelf (COTS) MTCs is selected. We assume that each COTS MTC has its own provided functions that are well defined by the developer of the MTC [7]. These provided functions are part of the MTC package that specifies the real world MTC. QoS characteristics are also included in the MTC package and can be used to examine if the QoS aspects of the proposed collaboration scheme satisfy the required system-level QoS aspects for the specific responsibility. If the QoS characteristics of this specific collaboration meet the QoS requirements of the system-level required responsibility, the design is accepted. Either wise corrective actions should be proposed and analyzed. Corrective actions may include: a) re-engineering of the collaboration scenario, b) the use of components with better QoS characteristics than the ones used in the previous design, or c) a combination of the above. It is assumed that the MTC developer has already performed a QoS analysis for the MTC. All this information comprises the offered QoS characteristics of the MTC [4]. It should also be stated that the MTC developer does not know during the MTC's development time all the systems that this MTC will be used in the future.

According to the top-down approach, for every system-level responsibility that has not been assigned to a single MTC, a set of abstract MTCs is defined and the required MTC responsibilities are specified along with the required collaboration scheme. Required system-level QoS characteristics are decomposed to derive the component-level required QoS characteristics. This process results to the definition of the required QoS characteristics at the level of constituent MTCs. The process of deriving MTC-level QoS characteristics from system-level ones is a complex process and has to be defined. At this time the engineer has well defined required specifications (functional and non functional, including QoS characteristics) for every abstract MTC. Using these required QoS characteristics the engineer is able to select from the market or his components repository the ones that their offered QoS characteristics meet the required ones [8]. If such MTC's do not exist they have to be further analyzed in order to be developed.

Advantages and disadvantages for the above approaches that result in the definition of the architecture of the MTS of its composite MTCs may be identified but it is expected that in the real MTS development process a

combination of both approaches will be used resulting to a hybrid more efficient approach.

The above process, bottom-up or top-down, is again applied to every composite MTC that has to be developed. It is applied iteratively down to the primitive-MTC level; the identification of primitive MTCs signals the end of this iteration. For each composite MTC the system modeling process as defined by the left-hand side part of the V-Model is followed. Analysis is applied and its architecture is defined in terms of constituent components (composite and/or primitive). Sequence diagrams are defined to realize use cases of the MTC and identify the activities that are involved in the specific use case. This is not the case for primitive MTCs that have to bypass the system process and follow a synergistic integration of the three constituent parts, i.e. mechanic, electronic and software (MTC synthesis) as shown in the bottom of the V-model in Figure 7.

For each primitive MTC, verification follows its integration as shown in the right-hand side of the MTS V-Model. Each composite MTC is integrated according to its MTC architecture and then it is verified against its requirements. After the integration and verification of the MTCs of the system, the MTS integration test is performed and the MTS is verified against its requirements.

It should be noted that the system analysis phase is followed by a system architecture design phase as shown in the proposed MTS V-model. This is also the case for the V-Model in software engineering. After this point the proposed V-model is completely differentiated from the traditional software engineering V-model. After the system architectural design, repetitions of analysis followed by architecture design for every composite MTC are applied following the system modeling process up to the primitive MTC level. This is the point where the system development process is terminated and the synergistic integration of constituent parts of the primitive MTC is performed working independently but in a synergistic way in the three disciplines. For every primitive MTCs that has complex software constituent part, a software V-model can be applied for its development, as shown in the bottom of the MTS V-Model.

Figure 8 presents two real world MTCs and the Parallel Kinematic Machine evolium MTS, that were developed based on the basic principles of the MIM architecture by a high-tech Italian company. Each axis of the Parallel Kinematic Machine has its own intelligence, so there is no need of an external entity to control the motion trajectory.

6. Future Developments and Research Challenges

Mechatronic systems development is a very complicated process imposing many challenges. In this section we refer to the ones that are of higher priority considering the 3+1 SysML view-model. The identification of the

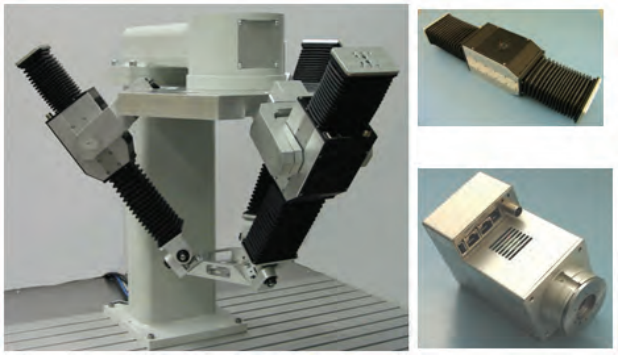


Figure 8. Real-world MTS and MTCs

mechanical discipline information that has to be captured in the system level models is one of the challenges. There are two possible approaches: a) exploit SysML constructs to represent as much of the mechanical discipline information, including component interfaces and behavior; and b) extend SysML constructs with new ones with the objective of creating a complete SysML model of the mechanical component. In the first case specific tools will be used for further refinement of the mechanical models and their subsequent execution and analysis. In the latter, the SysML models have to be automatically transformed to models of the specific tools for execution and model analysis. The integration of SysML with the Modelica language (www.modelica.org/) is towards this direction. This challenge is greatly related to the one that concerns the model execution, analysis and assessment of models on the MTS-view level.

Fully automated generation of the three views from the MTS view, as well as automatic update of the MTS-view with changes in the discipline views are important challenges that have to be addressed to improve the effectiveness of the process. This also imposes the challenge of integration of existing mechanical and electronic domain design tools.

The identification and definition of reusable MTCs is another major challenge in mechatronic systems. The Workpartner [21], a mobile service robot, is planned to be used in the context of a TEKES (Finnish Funding Agency for Technology and Innovation) funded project as a case study for the application of MIM, but also as a case study for the identification of reusable MTCs.

Since many of the MTSs are from the safety critical domain, the integration of the MTS V-model development process and the 3+1 SysML view-model with safety engineering is another major challenge for the MIM paradigm to be effectively exploited in safety- critical mechatronic systems.

7. Concluding Remarks

The traditional approach in the development of mechatronic systems is unable to address the needs of today's

complex mechatronic systems. An integrated framework for the construction of mechatronic systems is missing. The work presented in this paper attempts to contribute to this direction by: a) using SysML to define the artifacts of the MIM paradigm; b) proposing the 3+1 SysML view-model imposed by the MIM architecture; and c) extending the well accepted and widely used in the software domain V-model to address the demands of the mechatronic system development process. However, the challenges for a fully automated MTS development process crosses the boundaries of the three disciplines of mechatronic systems and impose a joint effort and collaboration between computer science, electronics and mechanics. The current status of discipline isolation imposed in many cases by the existing structure of engineering degree programs makes the task even more complicated.

8. Acknowledgments

Part of this work has been funded by TIKOSU, a project belonging to the Digital Product Process -program of the Finnish Funding Agency for Technology and Innovation (TEKES). The author wishes to thank the partners and especially Jarmo Alanen, Kari Koskinen, and Seppo Sierla for fruitful comments on these ideas. Thanks are also due to Piero Larizza, Eric Coatanea and Jussi Suomela for discussions on these concepts.

REFERENCES

[1] G. Rzevski, "On conceptual design of intelligent mechatronic systems," Mechatronics, 2003.

[2] W. Schafer and H. Wehrheim, "The challenges of building advanced mechatronic systems," Future of Software Engineering, International Conference on Software Engineering, IEEE Computer Society, 2007.

[3] Philipp Limbourg, "Dependability modelling under uncertainty: An imprecise probabilistic approach," Springer, 2008.

[4] K. Thramboulidis, "Model integrated mechatronics: Towards a new paradigm in the development of manufacturing systems," IEEE Transactions on Industrial Informatics, Vol. 1, No. 1, February 2005.

[5] OMG, "OMG Systems Modeling Language (OMG SysML™)," V1.0, September 2007.

[6] GD250, "Lifecycle process model 'V-Model'," Available online: http://www.informatik.uni-bremen.de/gd-pa/vmodel/vm1. htm#application

[7] K. Thramboulidis, G. Doukas, and G. Koumoutsos, "A SOA-based embedded systems development environment for industrial automation," EURASIP Journal on Embedded Systems, Article ID 312671, pp. 15, 2008.

[8] K. Thramboulidis, "Challenges in the development of mechatronic systems: The mechatronic component," 13th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA), Hamburg, Germany, September 2008.

[9] OMG, "Unified modeling language: Superstructure," version 2.1.1, formal/2007-02-03.

[10] M. Habib, "Mechatronics," IEEE Industrial Electronics Magazine, Vol. 1, No. 2, Summer 2007.

[11] S. Burmester, H. Giese, and M. Tichy, "Model-driven development of reconfigurable mechatronic systems with mechatronic 'UML' in model driven architecture," Springer Berlin/Heidelberg, Vol. 3599, 2005.

[12] K. Knorr, A. Lapp, P. Torre Flores, J. Schirmer, D. Kraft J. Petersen, M. Bourhaleb, and T. Bertram "A process model for distributed development of networked mechatronic components in motor vehicles," Proceedings of the IEEE Joint International Conference on Requirements Engineering (RE'02), 2002

[13] R. Nordmann, "Use of mechatronic components in rotating machinery," Book: Vibration Problems ICOVP 2005, Springer Netherlands, Vol. 111, January 20, 2007.

[14] J. El-khoury, O. Redell, and M. Torngren, "A tool integration platform for multi-disciplinary development," Proceedings of the 2005 31st EUROMICRO Conference on Software Engineering and Advanced Applications, 2005.

[15] G. Doukas and K. Thramboulidis, "A real-time Linux based framework for model-driven engineering in control and automation," IEEE Transactions on Industrial Electronics (forthcoming).

[16] G. Pahl, W. Beitz, J. Feldhusen, and K. H. Grote, "Engineering design: A systematic approach," Third Edition, Springer-Verlag London, 2007.

[17] R. Eckert, W. Mansel, and G. Specht, "Model transfer among CASE tools in systems engineering," Systems Engineering, Vol. 8, No 1, pp. 41–50, March 2005.

[18] P. Kruchten, "The 4+1 view model of architecture," IEEE Software, Vol. 12, No. 6, pp. 42–50, November 1995.

[19] OMG, "SysML and AP233 mapping activity," OMG SysML portal, http://www.omgwiki.org/OMGSysML/ doku. php?id=sysml-ap233:mapping_between_sysml_and_ap233.

[20] L. Constantine, "Activity modeling: Toward a pragmatic integration of activity theory with usage-centered design," Technical Paper, Available on-line: http://www. foru se. com/ articles/activitymodeling.pdf

[21] "The workpartner mobile service robot," http://automation.tkk.fi/

J. Software Engineering & Applications, 2010, 3: 119-124
doi:10.4236/jsea.2010.32015 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A Codebook Design Method for Robust VQ-Based Face Recognition Algorithm

Qiu Chen[1], Koji Kotani[2], Feifei Lee[1], Tadahiro Ohmi[1]

[1]New Industry Creation Hatchery Center, Tohoku University; [2]Department of Electronics, Graduate School of Engineering, Tohoku University, Japan..
Email: qiu@fff.niche.tohoku.ac.jp

Received September 5th, 2009; revised November 2nd, 2009; accepted November 6th, 2009.

ABSTRACT

In this paper, we present a theoretical codebook design method for VQ-based fast face recognition algorithm to improve recognition accuracy. Based on the systematic analysis and classification of code patterns, firstly we theoretically create a systematically organized codebook. Combined with another codebook created by Kohonen's Self-Organizing Maps (*SOM*) *method, an optimized codebook consisted of* 2×2 *codevectors for facial images is generated. Experimental results show face recognition using such a codebook is more efficient than the codebook consisted of* 4×4 *codevector used in conventional algorithm. The highest average recognition rate of* 98.6% *is obtained for* 40 *persons'* 400 *images of publicly available face database of AT&T Laboratories Cambridge containing variations in lighting, posing, and expressions. A table look-up* (*TLU*) *method is also proposed for the speed up of the recognition processing. By applying this method in the quantization step, the total recognition processing time achieves only* 28 *msec, enabling real-time face recognition.*

Keywords*: *Face Recognition, Vector Quantization* (*VQ*), *Codebook Design, Code Classification, Histogram Method

1. Introduction

After September 11th, security systems utilizing personal biometric features, such as, face, voice, fingerprint, iris pattern, etc. are attracting a lot of attention. Among them, face recognition have become the subject of increased interest [1], which seems to be the most natural and effective method to identify a person since it is the same as the way human does and there is no need to use special equipments. In face recognition, personal facial feature extraction is the key to creating more robust systems.

A lot of algorithms have been proposed for solving face recognition problem. Based on the use of the Karhunen-Loeve transform, PCA [2] is used to represent a face in terms of an optimal coordinate system which contains the most significant eigenfaces and the mean square error is minimal. However, it is highly complicated and computational-power hungry, making it difficult to implement them into real-time face recognition applications. Feature-based approach [3,4] uses the relationship between facial features, such as the locations of eye, mouth and nose. It can implement very fast, but recognition rate usually depends on the location accuracy of facial features, so it can not give a satisfied recognition result. There are many other algorithms have been used for face recognition. Such as Local Feature Analysis (LFA) [5], neural network [6], local autocorrelations and multi-scale integration technique [7], and other techniques [8–14] have been proposed.

Kotani *et al.* [15] have proposed a novel information-processing algorithm called Vector Quantization (VQ) codebook space information processing which differs from the traditional ways of processing algorithm. Based on this algorithm, we have developed a very simple yet highly reliable face recognition method called *VQ histogram method* by using a systematically organized Codebook for 4×4 blocks with 33 codevectors having monotonic intensity variation without DC component.

VQ algorithm [16] is well known in the field of image coding (compression). Input image is first divided into small blocks, which are taken as input vectors in VQ operation. Each input vector is then matched with codevectors in a codebook by calculating distances between them. The codevector having the maximum similarity to the input vector is selected by searching the minimum distance and the index number of the selected codevector is output.

This index number information was paid attention to. It was found that a codevector histogram, which is obtained by counting the matching frequency of individual

codevector, contains very effective facial feature information. By utilizing this technique, a novel face recognition algorithm called VQ histogram method has been developed.

A codebook is very important since it directly affects the quality of VQ processing. In [15], a special codebook was used, which is systematically organized for 4x4 blocks with 33 codevectors having monotonic intensity variation without DC component.

In this paper, a theoretical codebook design method is proposed. At first, a systematically organized codebook is created based on the distribution of code patterns abstracted from facial images [21], and then another codebook with the same size is created using Kohonen's Self-Organizing Maps (SOM) [20]. Combining the two codebooks obtained above, final optimized codebook consisted of 2×2 codevectors for facial images will be generated [22,23]. It can represent the features of the facial images more adequately. Furthermore, a table look-up (TLU) method is also proposed for the speed up of the recognition processing.

This paper is organized as follows. First, VQ histogram method will be introduced in detail in Section 2. Proposed codebook design method combining classification of code patterns and Kohonen's Self-Organizing Maps (SOM) will be described in Section 3. Experimental results compared with the algorithms employing original codebook or SOM codebook separately will be discussed in Section 4. Finally, we make a conclusion in Section 5.

2. Vector Quantization Histogram Method

In this section, we will describe the face recognition algorithm using Vector Quantization (VQ) histogram method [15]. Figure 1 shows face recognition process steps. First, low-pass filtering is carried out using simple 2-D moving average filter. This low-pass filtering is essential for reducing high-frequency noise and extracting most effective low frequency component for recognition. Block segmentation step, in which facial image is divided into small image blocks (for example, 2×2) with overlap, namely, by sliding dividing-partition one pixel by one pixel, is the following. Next, minimum intensity in the individual block is searched, and found minimum intensity is subtracted from each pixel in the block. Only the intensity variation in the block is extracted by this process. This is very effective for minimizing the effect of overall brightness variations. Vector quantization is then applied to intensity-variation blocks (vectors) by using a codebook which prepared in advance. The most similar (matched) codevector to the input block is selected.

After performing VQ for all blocks divided from a facial image, matched frequencies for each codevector are counted and histogram is generated. This histogram becomes the feature vector of human face. In the registration procedure, this histogram is saved in a database as personal identification information. In the recognition procedure, the histogram made from an input facial image is compared with registered individual histograms and the best match is output as a recognition result. Manhattan distance between histograms is utilized as a matching measure.

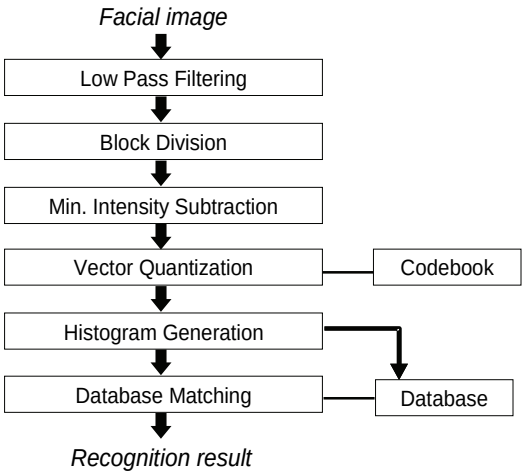


Figure 1. Face recognition process steps

Codebook which consists of typical feature patterns for representing the features of face image is very important. In [15], 32 codevectors of 4x4 codebook are created by changing the direction (8 different directions) and the range of intensity variation (Step values are 2, 6, 10, and 20). By adding one codevector having no intensity variation, complete codebook is organized.

In the next section, we will propose a novel codebook design method for 2×2 codebook, which can represent the features of the facial image more adequately.

3. Codebook Design

Because of the characteristic of face image, the codevectors of 4x4 codebook are all low-frequency patterns. Although the average face recognition rate of 95.6% has been obtained using such a codebook, it is still difficult to say this codebook is the most suitable because the number of categories for the 4x4 block patterns is too large to be classified by only 33 categories, and it has not been proved in theory that these 33 codevectors most adequately represent the face patterns. But in the case of 2×2 code patterns, the condition is considerably different. Nakayama *et al*. [17] have developed a complete classification method for 2×2 codebook design for image compression. By the similar consideration, we classify and analyze the code patterns in the face images, and then theoretically create a new codebook of 2×2 code patterns for face recognition algorithm.

3.1 Previous Work

Nakayama *et al.* [17] proposed a complete classification method for 2×2 codebook design in image compression. Figure 2 shows all categories for the 2×2 image block patterns without considering the location of pixels. In a 2×2 block, pixel intensities are marked by alphabet 'a', 'b', 'c', 'd', and a>b>c>d is prescribed. In [17], it was found that the number of typical patterns for all 2×2 image block is only 11. The number of varieties in pixel arrangement of each 2×2 typical pattern is also shown in Figure 2. That means the total number of image patterns for 2×2 pixel blocks is theoretically only 75.

By the similar consideration, we classified and analyzed the code patterns in the face images [21]. We found that in all filter size, the numbers of code patterns belong to categories 7, 10, and 11 are very few. It means such code patterns are almost not used in face images. Based on this result, we created a new codebook for 2×2 code patterns, and the rules of codebook creation are as follows.

1) Create very small intensity-variation (intensity difference among the block is only 1 or 2) code patterns having monotonic intensity variation, the number of code pattern is 16 by changing the direction of intensity variation.

2) Create code patterns of category No.2, 3, 4, 5, 6, 8 and 9, and intensity differences among the blocks are set to be 3, 6, and 10.

3) Do not make code patterns of category No. 7, 10, and 11.

4) Add one code pattern having no intensity variation.

Thus, complete codebook is systematically organized with 169 code patterns.

By using publicly available face database of AT&T Laboratories Cambridge [19], highest average recognition rate of 97.4% is obtained. Compared to the results of 4x4 codebook which the highest average recognition rate is 95.6%, recognition rate increases by about 2%.

But in [22], it was found that such distribution of codevectors appears non-uniform and concentrated only in some of the regions. According to *Maximum Entropy Principle* (MEP), the maximum entropy distribution will be achieved when the value of a random variable (counts) equals the average. Such a non-uniform distribution can not satisfy the MEP, so the recognition performance can not be expected be best because the average information content will not be maximum.

As a solution, Chen *et al.* [22] optimized the codebook by sorting the frequencies of all individual codevectors abstracted from facial images and excluding the codevectors in the codebook with low frequency. This method improved recognition performance and highest average recognition rate of 98.2% is obtained by using the same face database of AT&T Laboratories Cambridge [19].

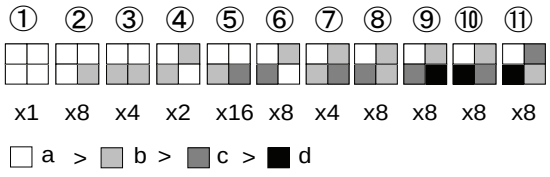


Figure 2. Categories of 2×2 code patterns. Pixel intensities are marked by alphabet 'a', 'b', 'c', 'd', and a>b>c>d is prescribed. The number of typical patterns for all 2×2 image block is only 11, thus total number of image patterns for 2×2 pixel blocks is theoretically only 75

3.2 Proposed Codebook Design Method

The essence of the method in [22] is to abstract the code patterns which are most frequently used in facial images. The consideration is correct, but it ignored the differences between different persons. The frequencies of some code patterns used in facial images may vary greatly for different persons and the values appear small. The code patterns selected to be used in codebook should not only present the most common features of faces, but also discriminate the difference of persons. The latter characteristic is very important in recognition task and can be evaluated by *mean square error* (MSE) of code patterns. Figure 3(a) shows the average histogram of 40 facial images in the database of AT&T Laboratories Cambridge [19] by using the 2×2 codebook generated according to the method proposed in [22]. In this case, the codebook size is set to be 80. The distribution is sorted in order of frequency. But the order of respective MSE values is out of accord as shown in Figure 3(b). The code patterns with high frequencies but low MSE values are useful to generate feature vectors but poor benefit recognition result. So we should consider both factors of frequencies and MSE values.

As a solution, our strategy is as follows.

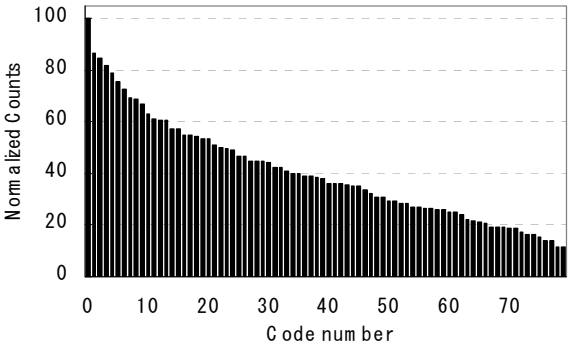
Step 1: Create a systematically organized codebook by applying an improved method based on the code classification including frequencies and MSE values.

Step 2: Create another codebook with the same size using Kohonen's Self-Organizing Maps (SOM) [20].

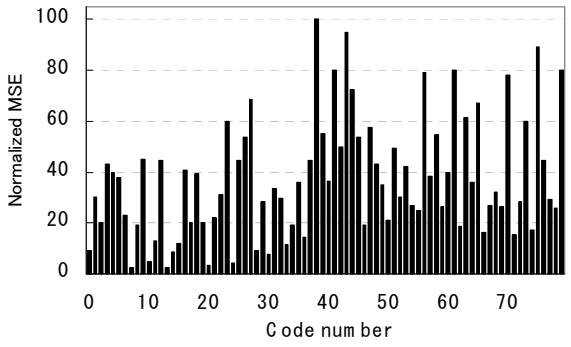
Step 3: Combine the two codebooks obtained above to generate the final optimized codebook consisted of 2×2 codevectors.

3.3 Data Set for Codebook Design

For covering the variations of the photo-taking conditions, two different face databases which are publicly available FERET database [18] and face database of AT&T Laboratories Cambridge [19], are utilized to analyze the code patterns of facial images. 40 facial images from different persons are selected from each database, and the face regions are abstracted with the sizes of 146×200 and 92×112, respectively. The typical examples of facial images are shown in Figure 4.



(a)



(b)

Figure 3. (a) Average histogram of 40 facial images using codebook of size 80. (b) MSE of code patterns. The distribution is sorted in order of frequency in (a). But the order of respective MSE values is out of accord as shown in (b)

(a) (b)

Figure 4. Data set for analysis of code patterns: (a) FERET database, (b) AT&T database

3.4 Codebook Generated by Code Classification

In the rules No. 2 of codebook creation in Subsection 3.1, the intensity variation steps are set to be 3, 6, and 10. From the distribution of the average histogram, we can see the steps for creating code patterns are not suitable obviously. We modify the rules and propose our improved codebook design method.

Figure 5 shows the processing steps of the codebook generation. At first, we change intensity differences among the blocks to be from 1 to 10, and implement the rules No.1–4 in Subsection 3.1 to create an initial large codebook of size 517. Thus the code patterns can almost cover all intensity variation. Utilizing this initial codebook, VQ processing is performed for all intensity variation blocks divided from the facial image in dataset described above, matched frequencies for each codevector are counted and histogram of each facial image is generated. Then average histogram and the normalized frequencies ($\overline{f_i}$) are calculated, where M is the number of facial images. MSE (e_i) (using RMSE, the square root of MSE) of individual codevectors is calculated by Formula 1 and then the scores (s_i) are computed by the weighted average between the normalized frequencies ($\overline{f_i}$) and MSE (e_i) as shown in Formula (2).

$$e_i = \sqrt{\frac{1}{M}\sum_{j=1}^{M}(\overline{f_i} - f_i(j))^2} \quad (1)$$

$$s_i = \frac{k_1\overline{f_i} + k_2 e_i}{\sum_{j=1}^{2} k_j} \quad (2)$$

where k_i (i=1, 2) is a weighting coefficient of respective component. The values of k_1, k_2 are 1, 1 respectively for all images used in our experiments, which determined by actual experiments.

Next, the scores (s_i) of individual codevectors are sorted, and the codevectors with high scores will be extracted. In this way, a systematically organized codebook is generated.

3.5 Codebook Generated by Kohonen's SOM

As a neural unsupervised learning algorithm, Kohonen's Self-Organizing Maps (SOM) [20] is one of the

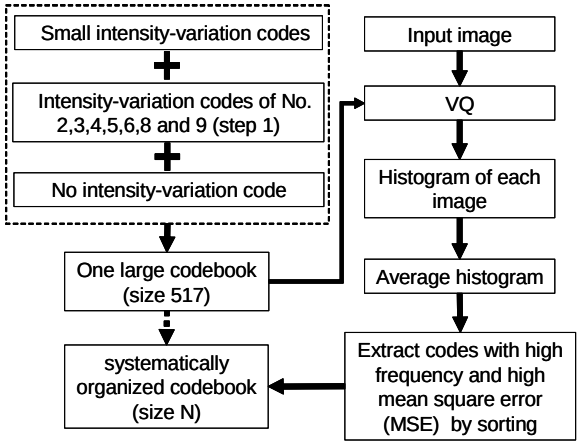


Figure 5. Processing steps of codebook generation based on code classification

standard algorithms used for codebook design in image coding. In this paper, SOM algorithm will also be employed to generate a SOM codebook with the same size as in Subsection 3.4.

Step 1: Transform the facial images in dataset to intensity variation vectors, and combine all data together into one training set.

Step 2: Specify the size of the codebook to N and initialize the codevectors by using continuous intensity variation vectors.

Step 3: Select a new training vector from the training set.

Step 4: Find the best-matching codevector closest to the training vector.

Step 5: Move the best-matching and its neighborhood codevectors towards the training vector.

Step 6: Repeat from Step 3 until the map converges.

3.6 Generation of Optimized Codebook

After the two codebooks described above are generated, they will be combined into one codebook of size 2N. Overlapped codevectors in codebook will be excluded. Next, like the processing in Subsection 3.4, VQ processing is also performed for all intensity variation blocks divided from the facial image in dataset, and average histogram of all images is calculated. The scores of all individual codevectors are calculated and sorted, and the codevectors in this codebook with low scores will be excluded. Thus, the size of codebook will be decreased from 2N to N, and the final optimized codebook consisted of 2×2 codevectors is generated.

4. Experiments and Discussions

4.1 Database

Face database of AT&T Laboratories Cambridge [19] is used for recognition experiments. In the database, 10 facial images for each of 40 persons (totally 400 images) with variations in face angles, face sizes, facial expressions, and lighting conditions are included. Each image has a resolution of 92×112. Five images were selected from each person's 10 images as probe images and remaining five images are registered as album images. Recognition experiment is carried out for 252 ($_{10}C_5$) probe-album combinations by rotation method. The algorithm is programmed by ANSI C and run on PC (Pentium(R)D processor 840 3.2GHz).

4.2 Experimental Results and Discussions

Firstly, we discuss the codebook size N. It is necessary to choose a suitable size of codebook. As the codebook size is too large, number of codevectors increases, the resolution of the histogram may become so sensitive that noise-corrupted codevectors may significantly distort the histogram. On the contrary, if the number of codevectors is too small, the histogram can not sufficiently discriminate between different faces.

Figure 6 shows the comparison of the recognition results using codebooks with different sizes from 30 to 200. The highest average recognition rates obtained in each codebook are shown here. The best performance is obtained at codebook size of 80 which is the same as in [22]. Maximum of the average rate 98.6% is achieved, which is 0.4% higher than that in [22].

We also compare the proposed algorithm with conventional algorithms. Figure 7 shows the recognition results. Recognition success rates are shown as a function of filter size which changes from none filtering to filter size of 23×23. The curve with rhombus marks stand for the average results in 252 ($_{10}C_5$) probe-album combinations using the 2×2 codebook created by proposed design method. The codebook size of 80 is used here. The curve with triangle marks and circular marks refer to the results reported in [21,22] which use original 2×2 codebook. The curve with open square marks refers to the results of 4×4 codebook [15]. In the curve with rhombus marks, recognition rate first increases with increase in filter size, and then, saturated or gradually decreases. The highest average recognition rate of 98.6% is obtained at the filter size of 13×13 while that of 98.2%, 97.4% and 95.6% using codebooks in [15,21,22], respectively. It can be said

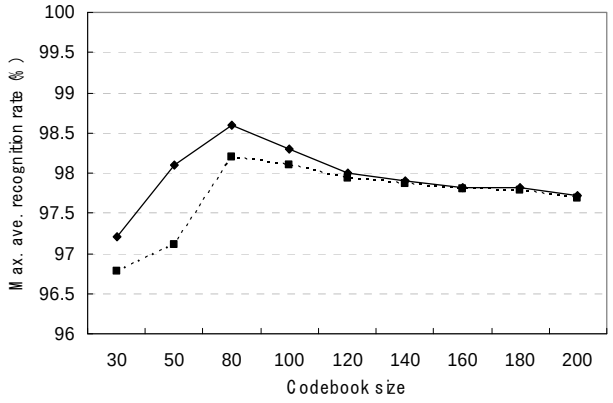


Figure 6. Comparison of recognition results in different codebook size

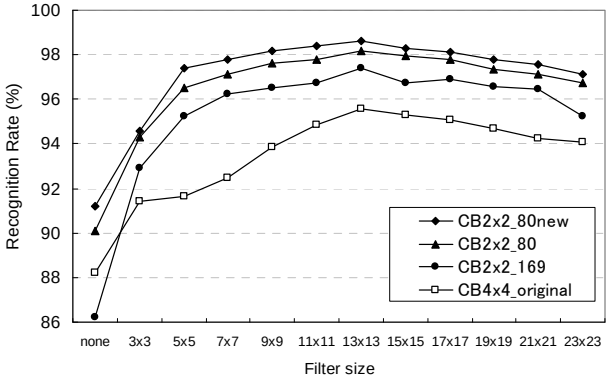


Figure 7. Comparison of the recognition results

that the 2×2 codebook created by proposed method represent the features of the facial images more adequately than conventional codebooks.

4.3 Processing Time

Processing time for single image in the face database of AT&T Laboratories Cambridge [19] is about 57 msec using a codebook of size 80, which is composed of 15 msec for pretreatment including filtering, block division, and minimum intensity subtraction, 30 msec for VQ processing, and 12 msec for database matching.

Furthermore, because a 2×2 codevector can be represented by an array of 4 dimensions, by utilizing the table look-up (TLU) method in the VQ processing step, the VQ processing time can be shorten to be about 1 msec, and the total running time will be 28 msec. It means our fast recognition algorithm achieves real-time face recognition.

5. Conclusions

In this paper, a theoretical codebook design method for robust VQ-based face recognition algorithm is proposed. Combining a systematically organized codebook based on the classification of code patterns and another codebook created by Kohonen's Self-Organizing Maps (SOM), an optimized codebook consisted of 2×2 codevectors for facial images is generated. Utilizing such a codebook of size 80, the highest average recognition rate of 98.6% is obtained for 40 persons' 400 images of the database of AT&T Laboratories Cambridge.

REFERENCES

[1] K. W. Bowyer, "Face recognition technology and the security versus privacy tradeoff," IEEE Technology and Society, pp. 9–20, Spring 2004.

[2] M. Turk and A. Pentland, "Eigenfaces for recognition," Journal of Cognitive Neuroscience, Vol. 3, No. 1, pp. 71–86, March 1991.

[3] L. Wiskott, J. M. Fellous, N. Kruger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 775–779, July 1997.

[4] R. Brunelli and T. Poggio, "Face recognition: Features versus templates," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 15, No. 10, pp. 1042–1052, October 1993.

[5] P. S. Penev and J. J. Atick, "Local feature analysis: A general statistical theory for object representation," Network: Computation in Neural Systems, Vol. 7, No. 3, pp. 477–500, 1996.

[6] R. Chellappa, C. L. Wilson, and S. Sirohey, "Human and machine recognition of faces: A survey," Proceedings IEEE, Vol. 83, No. 5, pp. 705–740, May 1995.

[7] S. Z. Li and A. K. Jain, "Handbook of face recognition," Springer, New York, 2005.

[8] F. Goudail, E. Lange, T. Iwamoto, K. Kyuma, and N. Otsu, "Face recognition system using local autocorrelations and multi-scale integration," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 18, No. 10, pp. 1024–1028, 1996.

[9] K. M. Lam and H. Yan, "An analytic-to-holistic approach for face recognition based on a single frontal view," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 20, No. 7, pp. 673–686, 1998.

[10] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," IEEE Transactions Pattern Analysis and Machine Intelligence, Vol. 19, No. 7, pp. 696–710, 1997.

[11] W. Zhao, "Discriminant component analysis for face recognition," Proceedings ICPR'00, Track 2, pp. 822–825, 2000.

[12] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," IEEE Transactions on Neural Networks, Vol. 13, No. 6, pp. 1450–1464, 2002.

[13] S. G. Karungaru, M. Fukumi, and N. Akamatsu, "Face recognition in color images using neural networks and genetic algorithms," International Journal of Computational Intelligence and Applications, Vol. 5, No. 1, pp. 55–67, 2005.

[14] S. Aly, N. Tsuruta, and R. Taniguchi, "Face recognition under varying illumination using Mahalanobis self-organizing map," Artificial Life and Robotics, Vol. 13, No. 1, pp. 298–301, 2008.

[15] K. Kotani, Q. Chen, and T. Ohmi, "Face recognition using vector quantization histogram method," IEEE 2002 International Conference on Image Processing, II–105–108, 2002.

[16] A. Gersho and R. M. Gray, "Vector quantization and signal compression," Kluwer Academic, 1992.

[17] T. Nakayama, M. Konda, K. Takeuchi, K. Kotani, and T. Ohmi, "Still image compression with adaptive resolution vector quantization technique," International Journal of Intelligent Automation and Soft Computing, Vol. 10, No. 2, pp. 155–166, 2004.

[18] P. J. Phillips, H. Wechsler, J. Huang, and P. Rauss, "The FERET database and evaluation procedure for face recognition algorithms," Image and Vision Computing Journal, Vol. 16, No. 5, pp. 295–306, 1998.

[19] AT&T Laboratories Cambridge, "The database of faces," at: http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html

[20] T. Kohonen, "Self-Organizing Maps," Springer, USA, 1995.

[21] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "Face recognition using codebook designed by code classification," 2006 IEEE International Conference on Signal and Image Processing, Hubli, India, pp. 397–401, December 2006.

[22] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "A VQ-based fast face recognition algorithm using optimized codebook," Proceedings of the 2008 International Conference on Wavelet Analysis and Pattern Recognition, Hong Kong, pp. 298–303, August 2008.

[23] Q. Chen, K. Kotani, F. Lee, and T. Ohmi, "VQ-based face recognition algorithm using code pattern classification and Self-Organizing Maps," Proceedings of 9th International Conference on Signal Processing (ICSP'08), Beijing, pp. 2059–2064, October 2008.

J. Software Engineering & Applications, 2010, 3: 125-133
doi:10.4236/jsea.2010.32016 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Building Requirements Semantics for Networked Software Interoperability

Bin Wen, Keqing He, Jian Wang

State Key Lab of Software Engineering, Wuhan University, Wuhan, China.
Email: binwenwebb@gmail.com

Received November 9th, 2009; revised December 1st, 2009; accepted December 20th, 2009.

ABSTRACT

Naturally, like the web, integrated software systems in Internet will have to be distributed and heterogeneous. To improve the interoperability of services for SAAS, it is crucial to build requirements semantics that will cross the entire lifecycle of services especially on requirements stage. In this paper, a requirements semantics interoperability extending approach called Connecting Ontologies (*CO*) *that will act as semantics information carrier designing to facilitate the requirements identification and services composition is proposed. Semantic measurement of Chinese scenario is explored. By adopting the approach, a series of tools support for transport domain are developed and applied based on CO and DPO* (*Domain Problem Ontology*) *to enforce requirements engineering of networked software efficiently.*

Keywords**:** *Networked Software, Requirements Semantics, Requirements Engineering, Connecting Ontologies*

1. Introduction

Ideally, users can access services based on their requirements without regard to where the services are hosted or how they are delivered. Various computing paradigms have promised to deliver IT as services including grid computing, P2P computing, and more recently Cloud computing. The latter term denotes the infrastructure as "Cloud" from which businesses and users are able to access application from anywhere in the world on demand. Thus, the computing world is rapidly transforming towards developing software for millions of consume as a service, rather than to run on their individual computers [1].

The development of networked software has emerged varied forms and definitions. One is pervasive computing, such as grid computing, e-science, and transparent computing, which focus on resource sharing. Another category is cloud computing based on SAAS (software as a service) and related studies include SOA, Web Service, Semantic Web Service etc. SAAS and virtualization of hardware and software are two main features for Cloud computing. Networked software that this paper refers to [2] belongs to the second sort that is complex information system based on Internet towards service computing. Distribution, autonomy, opening and heterogeneity are its basic features and stakeholders to be faced having various sorts and interests. Typically, supporting diversified, personalized and dependable services to improve user QoE (Quality of Experience) is the highest goal.

Requirements engineering (RE) is crucial to the success of software engineering, especially for networked software, and considering issues mainly include dynamic elicitation and analysis, evolution modeling, requirements management and model verification of user requirements and so on. Requirements modeling methods mostly are classified as structural requirements modeling and object-oriented requirements modeling according to paradigm, and both of them can deal with functional and nonfunctional requirements analysis. Now the typical software RE approaches are goal-oriented, ontology-oriented, scenario-based, problem framework, pre-requirements analysis based on domain modeling, document driving and aspect-oriented method [3].

The most widely significant approaches for networked software RE are goal-oriented and pre-requirements analysis based on domain ontology approach. Goal-oriented approach concentrates on analysis and modeling of early requirements so as to help developer understand the motivation and expectation for various roles, and involves the identification and analysis of functional and nonfunctional requirements goal. At present software RE is switching from object-oriented to goal-oriented [4,5], whereas goal-oriented approach has produced commercial products for tool supporting, for instance Cediti goal analyzer: Objectiver. Accordingly goal-oriented requirements analysis has become the hot spot of the

studying of RE.

Virtually, pre-requirements analysis based on domain modeling [6,7] is the process of requirements analysis based on domain-level ontology knowledge. The issue of ODE method based ontology [8] only acquires domain conceptual knowledge especially, but it ignores the modeling for task and functional knowledge.

All the above-mentioned requirements modeling methods consider only for object-orient development. The applicability and feasibility of those approaches for service-oriented computing must be reconsidered. Regarding the features for service computing, role, goal, process and service, the four fundamental elements can be used to modeling for the users' truly intentions of networked software. A meta-modeling framework containing the four fundamental elements, namely RGPS [9], is presented for conducting synergy and ordered structure requirements specification from disordered requirements information. Furthermore, choosing ontology meta-modeling [10] and encapsulating domain reusable core services asset, O-RGPS (Ontology-RGPS) meta-model proposal [2] is also put forward (see Figure 1).

Based on O-RGPS requirements meta-model framework, user requirements can be described from different angle, level and granularity in order to form domain requirements asset and store as OWL for reuse.

Interaction and collaboration of networked software is a restricted semantic interoperable issue on essence. Then, how to constrain and extend the semantic interoperability in the process of self-organization and action emergence for the distributing services resource? How to categorize the structure of interoperability? How to satisfy stakeholders' requirements?

Regarding the above issues, this paper proposes an

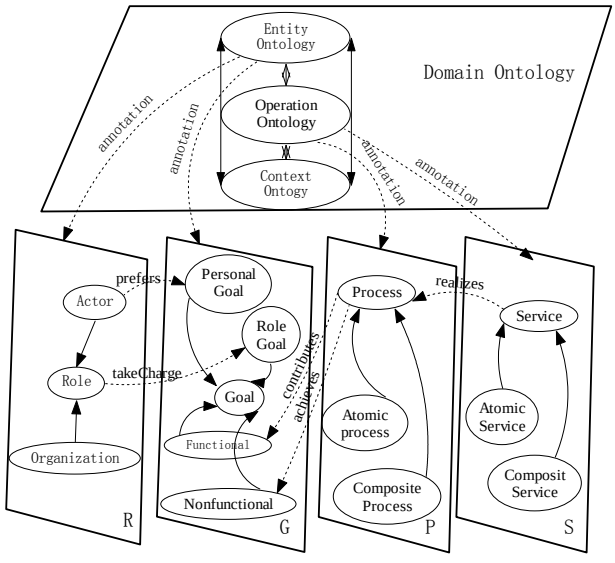


Figure 1. Domain asset customizing based-on O-RGPS

requirement semantic interoperable extending approach for networked software based on connecting ontologies (CO) and furnishes the unified and dynamic semantic information carrier for service aggregating and evolution modeling.

The rest of the paper is organized as follows: Section 2 explores software RE method based on domain ontology and requirements asset; furthermore, provides formal definition and aggregating method of connecting ontologies, and presents the related algorithm and integrating environment design for interoperable extending of networked software requirements semantics; Section 3 summarizes the related cutting-edge work in the research community; at the last, we conclude the paper and survey the future work.

2. Connecting Ontologies for Networked Software

Networked software system includes the overall architecture and goal software system that can embody dynamic property of the architecture. Goal software system is composed of services, whereas service resources distribute in network and are loosely coupled, dynamic binding and permit various levels of semantic interoperability.

Since service resources are dynamically distributed, for the sake of acquiring requirements knowledge from multi-domain service resources, disseminated ontology registry repositories in network require ontology encapsulation which is unified annotation of service with respect to requirements semantic. Ontology registry repositories will accord with ISO meta-model framework MFI (ISO/IEC SC32 19763) [11] that we participate. Requirements are gained by requirements acquiring & analysis (RAA) approach, and Requirements Sign Ontology (RSO, Definition 11, similar to process specification or workflow of application) is generated. Based on RSO, published ontologies of requirements semantic for available services are dynamic found and matched in network. Matched ontologies and RSO form ontologies group that is loosely coupled connected and dynamic generated, named Connecting Ontologies (CO). Stated in Figure 2, is requirements modeling approach for networked software based on CO. In ontology level, requirements semantic are dynamic acquired with semantic extending and matching. Furthermore, initial requirements model is generated by reusing multi-domain requirements asset. CO is the process of dynamic generating and continuous evolving, as stakeholders' requirements are uninterruptedly changed and loosely coupled for multi-domain requirements asset.

2.1 Domain Ontology Based on Description Logic

In the line of computer, ontology is explicit representation and description of conceptualization objects.

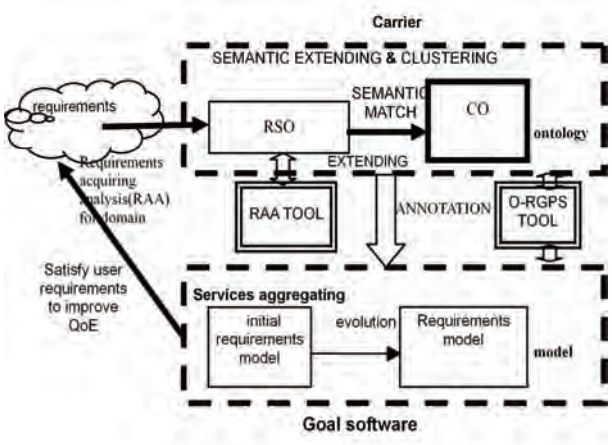


Figure 2. Requirements modeling for networked software based-on connecting ontologies

Ontology can also be used for software RE as requirements representation and carrier. At the same time, since reusability of broad-spectrum ontology is relatively hard, the principal application direction of ontology for software requirements should be domain-oriented and problem-oriented.

Firstly, this section gives the definition of domain ontology based on description logic and other related definitions. Next section will apply these definitions. Then requirements elicitation based on domain ontology and requirements asset is designed and implemented.

Definition 1 (Domain Ontology based on description logic). Domain Ontology is expressed as DO=<D, C, T, A, LH>, where D represents domain; C represents a set of concepts; T represents TBox; A represents ABox; and LH represents lattice hierarchy of concepts.

Definition 2 (Relation Triple among concepts). For domain ontology DO=<D, C, T, A, LH> with $c_p, c_q \in C$ and relation $r(c_p, c_q) \in T$, if c_p and c_q satisfy (1) $c_p \sqsubseteq c_q$ (2) $c_p \sqsubseteq \exists r.c_q$ (3) $c_p \sqsubseteq \forall r.c_q$, and $\sqsubseteq$ is concepts inclusion relation, then $\chi =< c_p, r, c_q >$ represents relation triple between c_p and c_q.

Definition 3 (Semantic Association). For two relations $\chi_1 =< c_{p1}, r_1, c_{q1} >$ and $\chi_2 =< c_{p2}, r_2, c_{q2} >$, $\chi_1 \odot \chi_2$ denotes semantic association between χ_1 and χ_2 where $\chi_1.c_{q1} = \chi_2.c_{p2}$.

Definition 4 (semantic association path). For a set of relations $X = \{\chi_i \mid \chi_i =< c_{pi}, r_i, c_{qi} >, i = 1, 2, \ldots, n\}$ and relation triples $\chi_s =< c_{ps}, r_s, c_{qs} >$, $\chi_d =< c_{pd}, r_d, c_{qd} >$, $\chi_s, \chi_d \in X$, if $\exists m \le n-2$, for $\forall \chi_{i+j} \in X, j = 1, 2, \ldots, m$, then DO have a semantic association path in X from χ_s to χ_d where semantic associations $\chi_s \odot \chi_{i+1}, \chi_{i+1} \odot \chi_{i+2}$ and $\chi_{i+m} \odot \chi_d$ exist, namely semantic association path between concept c_{qs} and c_{qd}.

Definition 5 (concept semantic depth, Depth). Apart form the class for itself, the meaning of ontology concept is also described by the associated classes, namely concept semantic depth. To calculate semantic depth, let the Depth of ontology root concept is zero, if the Depth of concept c, Depth(c), is I, then the Depth of its father concept (if existed) is I-1 and the Depth of its child concept (if existed) is I+1.

2.2 Connecting Ontologies

Connecting ontologies based on semantic matching of multi-domain requirements asset only utilize local or part of ontologies registry repositories for services. Modularization is an important technique of ontology reuse for services. Different researchers have different definitions or designations including segment, module, view or sub-ontology etc. The paper adopts sub-ontology [12] notion. Some definitions and algorithms are presented as follows.

Definition 6 (sub-ontology). For domain ontology DO=<D,C,T,A,LH>, a sub-ontology sub-Onto consists of 5 elements <C_{sub}, T_{sub}, A_{sub}, LH_{sub}, I>, where C_{sub} represents the set of sub-Onto concepts which denotes the context of sub-ontology; $|C_{sub}| < |C|$; there exist semantic association or semantic association path in C_{sub}; $T_{sub} \subseteq T$, $A_{sub} \subseteq A$ represent sub-Onto's local knowledge base for T_{sub}, A_{sub}; LH_{sub} represents lattice hierarchy of concepts; and I represents index pointer towards DO. If sub-ontology=DO or sub-ontology have nondeterministic domain, then I is nil.

Definition 7 (sub-ontology space in same source). For DO=<D,C,T,A,LH>, sub-ontology space in same source Space represents {<sub-Onto$_b$, B, DO| $\forall$ sub-Onto$_b$.I=DO, B $\in$ Index>}

2.2.1 Algorithm: Sub-Ontology Extracting Algorithm

For DO=<D,C,T,A,LH>, <CON,n,DO> is the input of sub-ontology extracting, where CON={con_1, con_2, …, con_k} represents a set of concepts which will be matched; DO represents father ontology; n represents the depth of travel. Based on [12], we can get sub-ontology extracting algorithm. The outcome of the algorithm is a sub-ontology sub-Onto.

Sub-ontology extracting algorithm can be seen from Algorithm 1.

Attentively, semantic similarity matching can be described in details: for any two concepts C1 and C2, assuming string S1 and S2 is the name of C1 and C2 respectively. Firstly, lexical analysis that preposition, conjunction, pronoun and interjection are cancelled is carried out for two strings, whereas continuous and meaning words are reserved. Strings S1 and S2 will be transferred

Algorithm 1 sub-ontology extracting algorithm.
INPUT: <CON, n, DO>
OUTPUT: sub-Onto
1: i<-1
2: Repeat
3: get i-th concept coni from CON
4: remove coni from CON
5: if exist a concept c satisfy concept coni semantic similarity match constraint in DO
6: then
7: do breadth-first traversal from c through relation in DO
8: if another concept c′ in DO is reached in traversal
9: then
10: add the triples along the traversed path from c to c′ into Tsub
11: add c to a set Csub
12: add c′ to a set Csub
13: remove c′ from CON
14: end if
15: stop when traversal up to a depth of n;
16: end if
17: i<- i+1
18: Until CON is empty or i>=k
19: m<- |Csub|
20: If m<k then return nil
21: end if
22: get the set of propertys Tsub from DO for all the concepts in CON
23: get the set of individuals Asub from DO for all the concepts in CON
24: get the extracting sub_Onto=<Csub,Tsub,Asub,LHsub,DO>
25: return sub_Onto

to <$S1_{w1}$,…,$S1_{wn}$> and <$S2_{w1}$,…,$S2_{wm}$>. For any words $S1_{wi} \in$ <$S1_{w1}$,…,$S1_{wn}$> and $S2_{wj} \in$ <$S2_{w1}$,…,$S2_{wm}$>, we can calculate two words' similarity similarityScore($S1_{wi}$, $S2_{wj}$)=wst.lookup($S1_{wi}$,$S2_{wj}$). This similarity is acquired by looking up similarity table which is generated by experts in matching computing by using words association tool (such as WordNet) in advance. If n<=m, then for $S1_{wi}$, we can find $S2_{wj}$ in accordance with maximum similarity, namely matchscore($S1_{wi}$, $S2_{wj}$)=similarityScore($S1_{wi}$, $S2_{wj}$). Finally similarity between two concepts is matchscore(C1,C2)=Sum(matchscore($S1_{wi}$, $S2_{wj}$))/n.

2.2.2 Algorithm: Sub-Ontology Merging Algorithm

For a set of sub-ontology, onto-set consists of {Sub-$Onto_1$, Sub-$Onto_2$, …, Sub-$Onto_n$}, $n \geq 2$, and the outcome of the algorithm generates a sub-ontology Onto= Merge(onto-set).

Sub-ontology merging algorithm can be seen in Algorithm 2 in details.

Definition 8 (maximum self-contained sub-ontology on concepts). For a set of concepts C which will be matched and a sub-ontology extracting algorithm, the last sub-ontology represents maximum self-contained sub-ontology on concepts C, where the set of concepts in the extracted sub-ontology unable to increase along with addition of travel depths to cease the extracting process.

Definition 9 (domain requirements ontology). For

Algorithm 2 sub-ontology merging algorithm.
INPUT:onto-set={Sub-Onto1, Sub-Onto2,….., Sub-Onton},n >=2
OUTPUT: Merge(onto-set)
1: add all concepts in Sub-Onto1.Csub from onto_set to a new set C
2: add all items in Sub-Onto1.Tsub from onto_set to a new set T
3: add all items in Sub-Onto1.Asub from onto_set to a new set A
4: i<- 2;
5: repeat
6: get i-th Sub_Onto Sub-Ontoi from onto_set
7: for each concept CK in Sub-Ontoi.Csub
8: if Ck not in C then
9: add Ck to C
10: end if
11: end for
12: for each item Mj in Sub-Ontoi.Tsub
13: if Mj not in T then
14: add Mj to T
15: end if
16: end for
17: for each item Aq in Sub-Ontoi.Asub
18: if Aq not in A then
19: add Aq to A
20: end if
21: end for
22: i<- i+1
23: until i>=n
24: get the corresponding Onto = <C,T,A,LH,nil>

convenient requirements acquisition and matching, domain requirements ontology is a special DO which only have two concepts with semantic depth Depth=1 in the sets of concepts: Operation denotes requirements verb concept and Entity denotes requirements noun concepts. Maximum self-contained sub-ontology of the set of operation is called operation ontology and maximum self-contained sub-ontology of the set of entity is called entity ontology for domain requirements ontology.

Definition 10 (Domain Problem Ontology, DPO). Domain Problem Ontology (DPO) represents as Merge ($\bigcup_{asset_i \in RGPS}$(Extracting (P, Dep,$asset_i$),$index_i$)), where P represents a set of problem's concepts; Dep refers to travel depth; RGPS represents domain-customized asset based RGPS; $index_i$ represents source ontology index with respect to matched problem concepts of RGPS asset.

Note that the Problem is a specific application context, for example travel is a Problem for traffic domain.

Definition 11 (Requirements sign ontology, RSO). Requirements sign ontology RSO consists of 3 elements <DSorl, Concept, Control>, where DSorl represents input in domain requirements service language; C represents the set of extracting concepts from DSorl; Concept $\supseteq$ DPO.C; Control represents control structure among matched service ontologies mainly including sequence, choice, split-union, any order, cycle.

CO are a sub-ontologies set with different sources in which involve dynamic finding and matching ontologies of published services, and RSO serves as mediator and

conducts the process of generating CO for service-oriented requirements.

Definition 12 (connecting ontologies, CO).Connecting ontologies (CO) consists of <RSO, DPO, Mapping-Onto-Set>, where RSO represents requirements sign ontology; DPO represents problem-oriented domain problem ontology; Mapping-Onto-Set represents matched sub- ontology set of different source.

Based on sub-ontology extracting algorithm and the direction of RSO, requirements semantic of CO firstly execute the matching for DPO. The rest of unabsorbed parts by DPO for CO run ontologies finding and matching from multi-domain services in network to satisfy requirements semantic for stakeholders. General speaking, the matched ontologies always denote some sub-ontologies of ontologies with respect to multi-domain services, and they are semantically matching with RSO, namely O_i (i=1~n). Then, as seen in Figure 3, connecting sub-ontology O_0 of DPO and sub-ontologies O_i of ontologies for multi-domain services according to RSO that acts as the center will dynamically generate CO. Accordingly, dynamically generated CO not only contain O_0 which is domain-oriented and tightly couple with DPO, but also do it include some services ontologies O_i for different domain i and loosely coupling with RSO. A few of unmatched services based CO will be solved by customizing manufacture.

2.3 Domain Problem Ontology

According to Definition 10, Domain Problem Ontology (DPO) is really a composite sub-ontology in terms of problem by extracting from Domain Ontology and RGPS requirements assets that express as OWL format. DPO is very important in the creating process of CO and acts as problem vision for CO. Creating CO firstly need adopting and matching with DPO, so the quality of DPO is crucial for the success of appropriate and preferred match regarding the contract ontology (i.e. CO) of all circles for software web clustering.

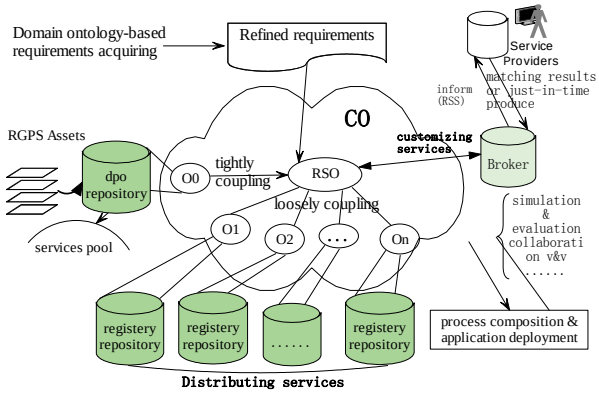


Figure 3. Connecting ontologies

We believe that: 1) semantic distance is only necessary and fundamental measure method for semantic interoperability capability; 2) for semantic interoperability measurement, semantic distance is not sufficient condition; 3) not only do semantic interoperability capability relate to similarity but also tightly associate with the contracted standard (i.e. CO) for both sides and really CO is sufficient condition for interoperability.

Generating DPO can adopt two fashions: semi-automated method directed by domain experts and fully automated method. We have realized the first fashion in our domain modeling tool designing to acquiring RGPS assets and automated fashion is now designing and optimizing. For automated fashion, we considered problem as follows: 1) the relation between DPO extracting depth (traverse depth) and CO matching degree with RSO; 2) the relation between DPO extracting depth (traverse depth) and extracting time cost.

For the above issues, we work out an experiment for evaluating these relations.

2.3.1 Experiment Design

Regarding low-scale Transport ontology (concepts number below 200) and OWL formatted R, G and P, experiment will evaluate the capability between DPO extracting depth associated with CO matching degree and time spending. Firstly, using Algorithm 1, 4 ontologies including Transport ontology, R, G and P [9], will be executed in accordance with the word “travel” and its synonym and outcome will be merged to generate DPO by Algorithm 2. RSO can be obtained by requirements acquiring tool [13] that we have implemented. Matching degree is manually achieved by domain experts between RSO and DPO.

2.3.2 Result Evaluation and Discussion

In the simulate experiment, the initial value of DPO extracting depth is 1. Through changeable extracting depth, we can get different matching degree and time cost for different depth value in order to analysis the influence of depth for entire CO generating process. Figure 4 is the result for different depth value.

According to the result, higher depth value will have higher matching degree with RSO. When DPO extracting depth is higher, the scale of DPO sub-ontology is also biggish correspondingly. Considering the principle of space locality, the reuse probability of DPO will evidently increase to enhance the matching degree with RSO. But higher depth value will lead to more time spending for creating DPO. At the same time, matching degree do not obviously enhance when the depth value increase from 6 to 8. It shows that only increasing depth value is not always efficient for improving matching degree. Since adopting sound depth value is very important for DPO to optimize the matching performance. The time

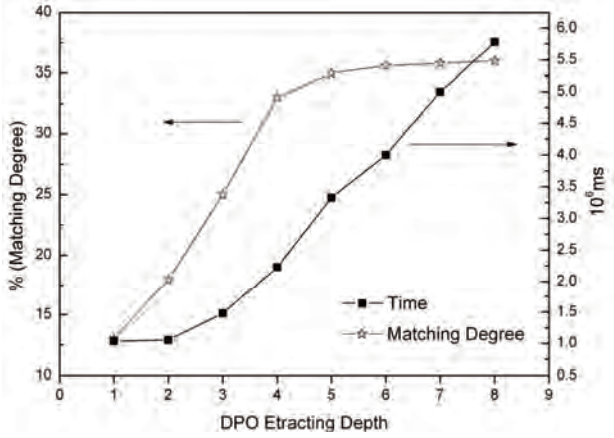


Figure 4. DPO extracting simulation result

cost of the experiment is higher than large-scale single ontology extracting in [12] because the experiment adds the spending of merging process.

The drawback of this experiment is low-scale original ontology, so future work will execute on large-scale ontologies to obtain valuable result for real-world.

2.4 Interoperability Extending Integrating Environment for Requirements Semantic Based CO

Regarding travel problem in urban traffic domain, simulation tests for acquiring requirements semantic based on CO [14] have shown that the semantic interoperability extending approach provides semantic information carrier for networked software and furnishes semantic goal for on-demand service aggregating. But now both RSO perfection and CO dynamic generating mainly rely on manually participating and customizing by requirements analyzers frequently, and quantitative measurement is absent for denoting semantic distance and interoperability level. Farther studies are listed as follows: 1) interoperability extending integrating environment for requirements semantic; 2) measurement system for requirements semantic interoperability.

2.4.1 Requirements Semantics Distance for Chinese Context

Now, software requirements semantics mainly adopts ontology encapsulation style, and requirements matching will reduce to similarity comparing among entities. Basic elements of entity include concept, relation and instance. Main measurement feature of concept are: concept name (no semantics, only consider linguistic and literal similarity, such as some distance formula [15]), concept semantics similarity, concept structure. Main measurement feature in relation involve property name, domain and range. Instance is auxiliary measurement for concept.

Semantics distance refers to a measurement of semantics similarity or association between two semantic entities. Semantic entities involving this paper are key words of documents. In general, semantics distance is a real number in $[0,\infty)$. Semantics distance has tight association with word similarity. Between two words, the bigger semantics distance is, the lower semantics similarity is and vice versa. They can be built a simple correspondence that need satisfy some conditions as follows: 1) similarity is 1 when semantics distance is 0 between two words; 2) similarity is 0 when semantics distance is infinity between two words; 3) between two words, the bigger semantics distance is, the lower semantics similarity is (monotony descend).

For two words w_1 and w_2, similarity expressed as $Sim(w_1,w_2)$, semantics distance is $Dis(w_1,w_2)$, then one can define a simple transfer relation that satisfy the above conditions:

$$Sim(w_1,w_2)=\frac{\alpha}{Dis(w_1,w_2)+\alpha} \quad (1)$$

α is a adjustable parameter that embody the words' distance value when similarity is 0.5. In the most cases, directly computing the words' similarity is difficult, so distance measurement can be calculated in advance and then transfer the similarity for words.

In general, thesaurus is the basis of the semantics distance measurement throughout computing MSCA (the Most Specific Common Abstraction) to acquire. To calculate semantics distance, one must use a comprehensive and exact structural semantic resource repository. Hownet (http://www.keenage.com) that involves more complete semantics knowledge content and is referred in some Chinese information processing is suitable for this studying.

Hownet includes two main definitions: concept and sememe. Concept is a description for vocabulary's semantics and every word can be expressed several concepts. Concept applies a knowledge representation language that uses sememe as vocabulary to describe.

Differentiated from the other thesaurus (e.g. Wordnet), Hownet don't reduce concept to a tree-like hierarchical architecture and that try to depict every concept using a series of sememes. Hownet adopts 1500 sememes which are divided into some categories as follows:

1) Event; 2) entity; 3) attribute; 4) aValue; 5) quantity; 6) qValue; 7) SecondaryFeature; 8) syntax; 9) EventRole; 10) EventFeatures.

For these sememes, they can be reduced to 3 groups: group 1 is called basic sememe to describe semantics feature for single concept containing sememes from category 1 to category 7; syntactic sememe only include category 8 to describe syntactic feature for words; group 3 contain category 9 and 10 called relation sememe to denote relation between concepts (similar to lattice relation from lattice syntax).

Semantics distance $d_1(p_1,p_2)$ between two sememes p_1 and p_2 is the path length from p_1 to p_2 in the sememe hierarchy structure.

For concept S_1 and S_2 which they have only one sememe in Hownet, semantics distance $d_1(S_1,S_2)$ is called the first basic sememe; except from the first basic sememe expression, for concept S_1 and S_2 which their semantics in Hownet is a set of basic sememes, $d_2(S_1,S_2)$ is defined as this part's semantic distance.

Corresponding to relation sememe description, its value is a feature structure. Considering every feature for the feature structure, its attribute is a relation sememe and its value is a basic sememe or a concrete word. This part of semantics distance for two concept S_1 and S_2 denote as $d_3(S_1,S_2)$.

For every feature of the above feature structure, if its value is a set in which the element of the set is a basic sememe or a concrete word, $d_4(S_1,S_2)$ can be designed to describe the part of relation signal sememe's semantics distance for concept S_1 and S_2.

Naturally, for the first basic sememe $d_1(S_1,S_2)$, $S_1(S_2)$ have a element-sememe $p_1(p_2)$ in Hownet, then $d_1(S_1,S_2) = d_1(p_1,p_2)$.

For the other basic sememes, if S_1 includes m sememes, S_2 includes n sememes, then

$$D_2(S_1,S_2)=\begin{Bmatrix} avg[d(p_{1i}, p_{2j})], m>0, n>0 \\ |m-n|, else \end{Bmatrix} \quad (2)$$

where p_{1i} is the sememe of S_1, p_{2j} is the sememe of S_2.

The following is a java program for calculating relation sememe:

```
private double disMap(
        Map<String, List<String> map1,
       Map<String, List<String> map2)
    {
      if (map1.isEmpty() || map2.isEmpty())
      {return
      Math.abs(map1.keySet().size() - map2.keySet().size());
      }
      double min = DEFAULT_DISTANCE;
      for (String key : map1.keySet( )) {
       if (map2.containsKey(key)) {
       List<String> list1 = map1.get(key);
       List<String> list2 = map2.get(key);
       double sim = disPriList(list1, list2);
       if (sim < min) {
         min = sim;
      }}}
      return min;
    }
```

Similarly, we can also get the java program for calculating relation signal sememe's semantics distance.

Considering the above-mentioned factors, for two concepts S_1 and S_2, semantics distance is defined as [15]:

$$d(S_1,S_2)=\sum_{i=1}^{4}\beta_i \bullet d_i(S_1,S_2) \quad (3)$$

where β_i $(1 \le i \le 4)$ is adjustable parameter and $\beta_1+\beta_2+\beta_3+\beta_4=1$, $\beta_1 \ge \beta_2 \ge \beta_3 \ge \beta_4$; if d_i=0, then β_i will assign other item proportionally. The act for global similarity from d_1 to d_4 is descending order. Since the first basic sememe expression reflects the main feature for concept, its weigh value should be defined comparatively bigger and larger than 0.5 usually.

Based on semantics distance between Chinese concepts, we can calculate semantics distance between two sentences w_1 and w_2 for Chinese SORL [16], where w_1 contains m concepts (S_{11}, …, S_{1m}), w_2 has n concepts (S_{21}, …, S_{2n}).

If w_1 is context-unaware and S_{1i} is unknown, then $Dis(w_1, w_2)$=min $Dis(S_{1i}, S_{2j})$, $1 \le i \le m$, $1 \le j \le n$.

If w1 is context-aware and S_{1i} is definite, then $Dis(w_1, w_2)$=min Dis(S_{1i}, S_{2j}), $1 \le j \le n$.

Similarity measurement between two ontologies will be calculated based on the above parts according to weight value synthetically. The relation between ontology similarity measurement and connecting ontologies can be induced as follows: firstly the extracting operation for ontologies is processed to adopt limited candidate ontologies; then calculating ontology similarity among ontologies will be run in order to choose the most similar ontologies for matching.

On the basis of studying in this section, we have designed Chinese semantics distance measurer and matcher for software requirements semantics matching measurement on connecting ontologies to build a measurement ground for connecting ontologies generating.

2.4.2 Integrating Environment

This section presents the design of interoperability extending integrating environment for requirements semantic based CO in Figure 5. Applying sub-ontology extracting algorithm, DPO can be generated from requirements asset that has been produced by domain modeling tool in the phase of requirements elicitation. DPO and domain requirements asset together become reusable asset for requirements acquiring and modeling tool.

Within the requirements acquiring and modeling tool, semantic matcher, which can execute matching operation with semantic distance measurement tool to achieve the matching for role, goal, and process of requirements asset, will be added. Main functions of semantic distance measurer include: measure semantic distance between two concepts; measure semantic distance between two ontologies; measure semantic distance between two services. Existing basis is: 1) thesaurus: WordNet (English), HowNet (Chinese); 2) similarity calculating based on two thesaurus.

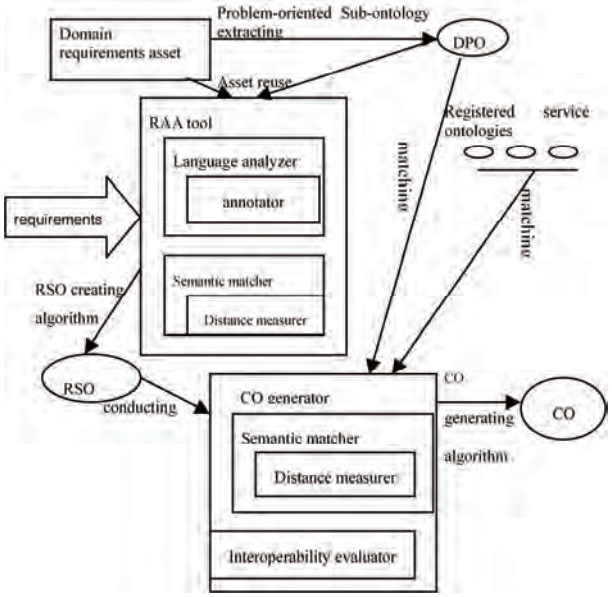


Figure 5. Requirements semantics interoperability extending integrating environment based-on CO

Adopted approach is: calculating two concepts similarity from words similarity; calculating ontologies similarity based on concepts similarity; calculating services similarity based on concepts similarity.

To generate CO, the function of CO generator is driven and conducted by the control structure of RSO, and it will use semantic matcher and interoperability level evaluator. It can automatically complete the task for looking up reusable resources with CO generating algorithm purposed in the above part to the more extent.

After received CO, interoperability level evaluator, which will evaluate semantic interoperability level, able to decide the preference grade for candidate services and forecast the QoE of users.

We have designed and implemented a series of tools for supporting service identifying and composition based on CO and DPO. Relative prototype and validation of the proposed approach have also partly achieved. Experiment has demonstrated that the proposed approach is useful for service finding and integrating. The snapshot of primary tools and Prototype system for context of traffic travel problem domain can see from Figure 6.

Figure 6. Prototype context and tools of traffic travel problem domain

3. Related Work

Application of ontology in RE starts from domain engineering. As reusable core resources in product line, domain requirements [17] mainly solve requirements modeling issue for component-oriented software system.

Dr. Jerome Euzenat from INRIA Grenoble Rhone-Alpes in France has studied semantic interoperability issues based ontology mapping [18,19] and acts as principal in NeOn project of EU FP6 plan. In June 2008, Informatics of EU startup semantic interoperability central plan for Europe and set up first session in Brussels aiming at realizing semantic data interoperability for E-government in Europe. Open source SILIME project of MIT-Semantic Interoperability of Metadata and Information in unLike Environments attempts to semantic interoperability for data resources (such as data library).

The studying of connecting ontologies is new direction in the world. Initial investigation studies original domain-level ontology for heterogeneity and explores how to create new ontology for covering original ontology with collaboration and consistence, and also containing ontology grouping technology (for example ontology mapping, ontology aligning, ontology merging etc.). In 2007, the paper by Shuaib Karim [20] presented a CO application framework that need not cover original ontology and focus on studying transfer principle and intermediate concept among original ontologies. Cregan Anne [21] proposes to build semantic interoperability by CO and gives some CO examples of gene ontology in 2008. However, in Cregan Anne's paper, connecting manners of CO, incentive of connecting, method and critical content of building semantic interoperability are absent. We also notice that Linked Open Data [22] initiative has become the existing foundation for federal Web of Data.

Now, together with CO and RE, the investigation of

requirements semantic interoperability extending for networked software with respect to service-oriented computing just begins to proceed, and a great deal of theoretical and technological issues will require to solve.

4. Conclusions

This paper explores ontology-based RE, for interoperability extending of requirements semantics; we present CO approach to improve requirements modeling under the condition of distributed services aggregation with loosely coupling and different domain. Some formal definition and generating algorithm of CO are given. With the novel approach, a integrating environment and measurement system based on CO is designed and implemented.

Further work can be classified as follows: studying partial meaning of semantic interoperability for networked software requirements; build CO based on Linked Open Data infrastructure; empirical testing for integrating environment with multi-domain, such as financial risk assessment, environment protection and so on.

5. Acknowledgments

This research has been partly supported by the National Basic Research Program of China (Grant No. 2007CB-310801) and the National Natural Science Foundation of China under Grant No.60970017 and 60903034.

REFERENCES

[1] Rajkumar Buyyaa, Chee Shin Yeoa, Srikumar Venugopala, James Broberg, and Ivona Brandic, "Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility," Future Generation Computer Systems, Vol. 25, No. 6, pp. 599–616, June 2009.

[2] K. Q. He, R. Peng, W. Liu, *et al.* "Networked Software," Science Press, Beijing, 2008.

[3] Z. Jin, L. Liu, and Y. Jin, "Software Requirements Engineering: Principles and Method," Science Press, Beijing, 2008.

[4] J. Mylopoulos, L. Chung, and E. Yu, "From object-oriented to goal-oriented requirements analysis," Communications of ACM, Vol. 42, No. 1, pp. 31–37, January 1999.

[5] A. V. Lamsweerde and E. Letier, "From object orientation to goal orientation: A paradigm shift for requirements engineering," Radical Innovations of Software and System Engineering in the Future, pp. 325–340, 2004.

[6] R. Q. Lu, Z. Jin, and G. Chen, "Ontology-oriented requirements analysis," Journal of Software, Vol. 11, No. 8, pp. 1009–1017, August 2000.

[7] Z. Jin, "Ontology-based requirements elicitation," Chinese Journal of Computers, Vol. 23, No. 5, pp. 486–492, May 2000.

[8] R. A. Falbo, G. Guizzardi, and K. C. Duarte, "An ontological approach to domain engineering," In Proceedings of the International Conference on Software Engineering and Knowledge Engineering (SEKE02), Ischia, Italy, pp. 351–358, 2002.

[9] J. Wang, K. He, P. Gong, *et al.* "RGPS: A unified requirements meta-modeling frame for networked software," In Proceedings of Third International Workshop on Advances and Applications of Problem Frames (IWAAPF'08) at 30th International Conference on Software Engineering (ICSE'08), Leipzig, Germany, pp. 29–35, May 2008.

[10] K. Q. He, F. He, and B. Li, "Research on service oriented ontology meta modeling theory and methodology," Chinese Journal of Computers, Vol. 28, No. 4, pp. 524–533, April 2005.

[11] K. Q. He, Y. F. He, and C. Wang, "International standard: Information technology-metamodel framework for interoperability (mfi)-3: metamodel for ontology registration," (ISO/IEC19763–3), online at: http://www.iso.org/iso/iso_catlogue/catalogue_tc/catalogue_detail.htm?csnumber=38637. ISO, 2007.

[12] Y. Mao, Z. Wu, and H. Chen, "Sub-ontology based resource management for web-based e-learning," doi: http://doi.ieeecomputersocirty.org/10.1109/TKDE.2008.127, 2008

[13] B. Hu, K. Q. He, H. F. Chen, and J. Wang, "Requirements driven web service composition based on RGPS domain assets: Approach and realization," Journal of Chinese Computer System, Vol. 30, No. 5, pp. 859–862, May 2009.

[14] K. Q. He, "Semantic interoperability refining and clustering theory and its application in on demand service aggregation," Science in China, F: Information Science (unpublished).

[15] L. Lin, "Text clustering research based on semantic distance," Master's thesis, Xiamen University, April 2007.

[16] W. Liu, "Research on services-oriented software requirements elicitation and analysis," PhD thesis, Wuhan University, June 2008.

[17] M. Mikyeong and Y. Keunhyuk, "An approach to developing domain requirements as a core asset based on commonality and variability analysis in a product line," IEEE Software Engineering (unpublished), Vol. 31, No. 7, pp. 551–569, July 2005.

[18] J. Euzenat, "An api for ontology alignment," In Proceedings of 3rd International Semantic Web Conference (ISWC), Hiroshima, Japan, Lecture Notes in Computer Science, Vol. 3298, pp. 698–712, 2004.

[19] J. Euzenat and P. Shvaiko, "Ontology matching springer," Heidelberg, Germany, 2007.

[20] Shuaib Karim, Khalid Latif1, and A. Min Tjoa1, "Providing universal accessibility using connecting ontologies: A holistic approach," Lecture Notes in Computer Science 4556, pringer-Verlag, Berlin Heidelberg, Vol. 3, pp. 637–646, S 2007.

[21] Cregan Anne, "W3c semantic web ontology languages: Owl and rdf tutorial," Technical Report, ISO/IEC JTC1 SC32 11th Open Forum on Metadata Registries, Sydney, Australia, May 2008. Tutorial.ppt.

[22] C. Bizer, T. Heath, and T. Berners-Lee, "Linked data: The story so far," International Journal on Semantic Web and Information Systems, Vol. 5, No. 3, pp. 1–22, 2009.

J. Software Engineering & Applications, 2010, 3: 134-140
doi:10.4236/jsea.2010.32017 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Research on Knowledge Transfer Influencing Factors in Software Process Improvement

Jiangping Wan[1,2], Qingjing Liu[1], Dejie Li[1], Hongbo Xu[3,4]

[1]School of Business Administration, South China University of Technology, Guangzhou, China; [2]Institute of Emerging Industrialization Development, South China University of Technology, Guangzhou, China; [3]School of Computer Science and Engineering, South China University of Technology, Guangzhou, China; [4]Guangzhou O-Engineer Information Technology Ltd, Guangzhou, China.
Email: scutwjp@126.com, ruthy07@163.com, ab23456@163.com

Received October 28th, 2009; revised November 17th, 2009; accepted November 25th, 2009.

ABSTRACT

Knowledge transfer model of software process improvement (*SPI*) *and the conceptual framework of influencing factors are established. The model includes five elements which are knowledge of transfer, sources of knowledge, recipients of knowledge, relationship of transfer parties, and the environment of transfer. The conceptual framework includes ten key factors which are ambiguity, systematism, transfer willingness, capacity of impartation, capacity of absorption, incentive mechanism, culture, technical support, trust and knowledge distance. The research hypotheses is put forward. Empirical study concludes that the trust relationship among SPI staffs has the greatest influence on knowledge transfer, and organizational incentive mechanism can produce positive effect to knowledge transfer of SPI. Finally, some suggestions are put forward to improve the knowledge transfer of SPI: establishing a rational incentive mechanism, executing some necessary training to transfer parties and using software benchmarking.*

***Keywords*:** *Software Process Improvement, Knowledge Transfer, Influence Factors, Pattern*

The software process is the set of tools, method, and practices we use to produce a software product. The objectives of software process improvement (SPI) are to process produce products according to plan while simultaneously improving the organization's capability to produce better products [1]. The six basic principles of SPI by Watts S. Humphrey are as follows: 1) Major changes to the software process must start at the top; 2) Ultimately, everyone must be involved; 3) Effective change requires a goal and knowledge of the current process; 4) Change is continuous; 5) Software process changes will not be retained without conscious effort and periodic reinforcement; 6) Software process improvement requires investment [1]. Alfonso Fuggetta argues that the scope of software improvement methods and models should be widened in order to consider all the different factors affecting software development activities. We should reuse the experiences gained in other business domains and in organizational behavior research. Statistics is not the only source of knowledge. We should also appreciate the value of qualitative observations [2]. Wan Jiangping argues that managers should think deeply into their think processes. The following issues in software organization can be resolved with SPI: 1) The processes and their principles for how to inherit and acquire others' knowledge; 2) The processes and their principles for conversion knowledge into their capability [3]. Literature 6 describes a repository of 400 process improvement experiments and presents patterns that help organizations plan their improvement initiatives [4].

*This research was supported by Key Project of Guangdong Province Education Office (06JDXM63002), Soft Science project of Guangdong Province (2007B070900026), NSF of China (70471091), and QualiPSo (IST-FP6-IP-034763).

1. Introduction

1.1 Organization Knowledge in Software Process Improvement

Organizational knowledge creation is the process of making available and amplifying knowledge created by individuals as well as crystallizing and connecting it to an organization's knowledge system [5]. Software organization is a highly knowledge-intensive enterprise, knowledge transfer is critical for software enterprise. It is obvious that software process is also an organizational knowledge intensive learning process and needed to be supported with knowledge management [6].

Sandra A. Slaughter and Laurie J. Kirsch conceptualize knowledge transfer portfolios in terms of their composition (the types of mechanisms used) and their intensity (the frequency with which the mechanisms are utilized). They hypothesize the influence of organizational design decisions on the composition and intensity of knowledge transfer portfolios for SPI. They then posit how the composition and intensity of knowledge transfer portfolios affect performance improvement. Their findings indicate that a more intense portfolio of knowledge transfer mechanisms is used when the source and recipient are proximate, when they are in a hierarchical relationship, or when they work in different units [7]. Literature 8 includes: 1) A knowledge management framework for SPI; 2) An innovative knowledge modeling and control approaches; 3) Mining and retrieval approaches on the software process assets; 4) A knowledge management for SPI.

1.2 Knowledge Transfer

Bloodgood considers knowledge transfer as knowledge transfer and transmit among various organizations and individual [9]. Argote considers enterprise knowledge transfer as a process that one organization's experiences impact on other's organizational action. It is that knowledge change or change knowledge recipients' behavior [10]. Davenport considers knowledge transfer as unified process which consists of both knowledge transfer process and knowledge absorbs process [11]. The effective knowledge transfer is that transfer knowledge is reserved [12]. Ingram considers knowledge transfer as process sharing knowledge in organization through various channels in order to make use of extant knowledge effectively [13]. Dong-Gil Ko *et al.* consider knowledge transfer as transmitting process in which knowledge transfer from owners to recipients for their learning and application [14].

In our understanding, knowledge transfer includes three aspects which are the process spreading from owners to recipients, activities occurring under contextualization and special goal. But the ultimate goal is to make the knowledge of the owners be the recipients' and narrow the knowledge gap between owners and recipients so as to promote the co-development of individuals and organizations. We define knowledge transfer as the process making knowledge transferring from the source of knowledge to recipients in contextualization.

1.3 Knowledge Transfer Model

The knowledge transfer model mainly includes process model and factors model. The process model is a model dividing knowledge transfer into different stages. The representative process models are Nonaka knowledge spiral model [15], Szulanski four stages model [12], and Gilbert&Cordey-Hayes five steps model. While factors model bases on factors in the process the knowledge transfer [16]. The representative ones of it are the four factors model invented by Jeffrey L. Cummings and Bing-Sheng Teng which includes sources of the knowledge, recipients of the knowledge, knowledge and context and transfer framework invented by Vito Albino *et al.* including transfer subject, context, content, and transfer media. Jeffrey L. Cummings and Bing-Sheng Teng's factors model is applied in this study [17].

1.4 Knowledge Transfer Influencing Factors

Knowledge transfer influencing factors are in the following [18]: 1) Characteristics of knowledge transferred include causal ambiguity and unprovability. 2) Characteristics of source of knowledge include knowledge providers shortage of motivation to transfer knowledge and unbelieving. The knowledge owners will not sharing knowledge with others because they are afraid losing knowledge possession, superiority complex, right and status and so on, lack time to sharing knowledge with others and couldn't proper reward and return on knowledge sharing. When the experts are not discovered and believed, their suggestions may be rejected and more challenged. 3) Characteristics of recipient of knowledge include knowledge recipients' both absorbing capability and keeping capability. It is very important for the knowledge recipient's capability to absorb others' knowledge and integrate into individual knowledge on the condition he will accept the knowledge developed by other. 4) Characteristics of context include barren organizational context and arduous relationship. Both will impart on knowledge transfer.

2. Research Model and Hypotheses

The knowledge transfer model of SPI, which includes five factors involving knowledge transferred, source of knowledge, recipient of knowledge, relationship of two parties and context. Besides is proposed, we consider knowledge transfer of SPI as a process which includes transmission, absorption and feedback (Figure 1). Knowledge transferred between source of knowledge and recipient of knowledge in the SPI must experience three stages. The stage of transmission is to transmit knowledge from source of knowledge to recipients of knowledge. The stage of absorption is about processing, sorting, and absorption process with their own mental model when recipients receive new knowledge. In stage of feedback, recipients of knowledge constantly communicate and feedback with source of knowledge in the process of absorption to master the knowledge transferred by the owners of knowledge. Thus, based on knowledge transfer model of SPI, we propose conceptual framework of ten key influencing factors of knowledge transfer in SPI. The influencing factors include ambiguity, systematism, transfer willingness, capacity of impartation, capacity of absorption, incentive mechanism, culture, technical

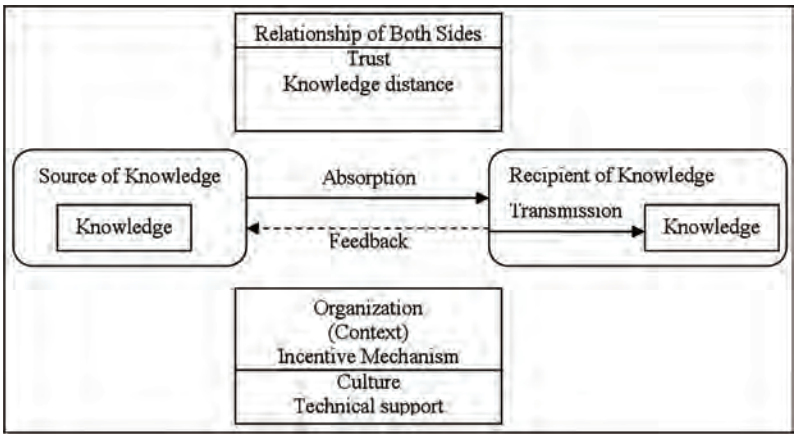


Figure 1. Knowledge transfer model of SPI

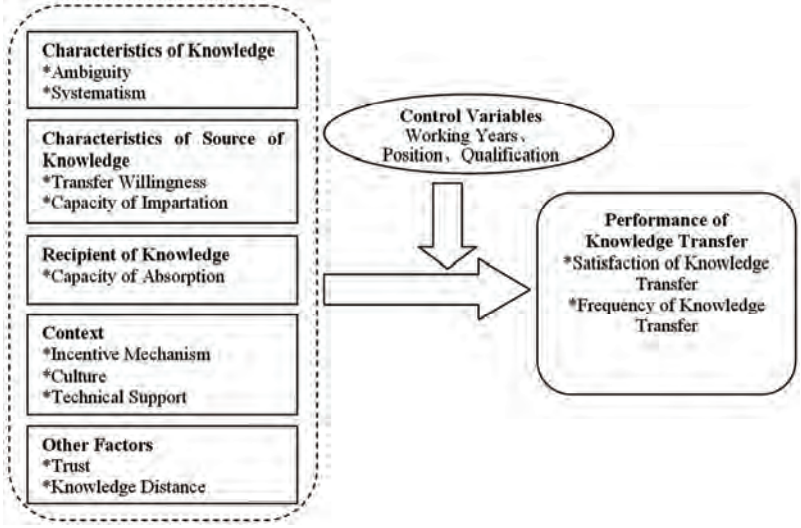


Figure 2. Conceptual framework of influencing factors

support, trust and knowledge distance (Figure 2). Then we propose our research hypotheses based on knowledge transfer model and conceptual framework of knowledge transfer influencing factors (Table 1).

3. Research Design

3.1 Scale Design

We studied the variable indicators on a five-point Likert-type scale. The questionnaire of pre-investigation was divided into four parts. The first part was questionnaire direction, including questionnaire background and introduction of basic information. The second part was the basic information of respondents, including gender, age and qualifications of respondents; the third part was the main body of the questionnaire, mainly about the issues of influence factors, including a total of 46 items. The last part with regard to performance of knowledge transfer involved a total of 4 items. After the completion of the pre-prepared questionnaire, we pre-prepared questionnaire in a small number of target groups by e-mail. We issued 10 questionnaires and six were received. In the end, we received a total of 114 questionnaires, of which eight were invalid, 106 were valid.

3.2 Data Collection

All the samples of our research are SPI staffs from Guangdong Software Organization who mainly concentrate in Guangzhou, Shenzhen and Zhuhai. From the job level of respondents, senior manager accounts for 8.5%, the percent of project manager is 8.5%, while the general staff contributes the large percent of 66.0%. From the qualifications of respondents, undergraduate accounts for about 74.5%, graduate 25.5%. From the age of respondents, age ranged from 24 to 30 accounts for the largest proportion of 67.9%, age bellow 24 and above 30 are 14.2% and 17.9% respectively. From the staff size of process improvement of investigated company, staff size bellow 10 comes up to 63.2%, staff size between 11 to 30 accounts for 17.9%, size above 31 reaches 18.9%. It can clearly be seen that, the samples have representative to some extent, and are suitable for the next phase of data analysis.

Table 1. Research hypotheses

Items	Hypotheses
Ambiguity	H1: Ambiguity of knowledge transferred has correlation with performance of knowledge transfer
	H1a: Ambiguity of knowledge transferred has negative correlation with satisfaction of knowledge transfer
	H1b: Ambiguity of knowledge transferred has negative correlation with frequency of knowledge transfer
Systematicness	H2: Systematism of knowledge transferred has correlation with performance of knowledge transfer
	H2a: Systematism of knowledge transferred has negative correlation with satisfaction of knowledge transfer
	H2b: Systematism of knowledge transferred has negative correlation with frequency of knowledge transfer
Transfer Willingness	H3: Transfer Willingness of source of knowledge has correlation with performance of knowledge transfer
	H3a: Transfer Willingness of source of knowledge has positive correlation with satisfaction of knowledge transfer
	H3b: Transfer Willingness of source of knowledge has positive correlation with frequency of knowledge transfer
Capacity of Impartation	H4: Capacity of Impartation of source of knowledge has correlation with performance of knowledge transfer
	H4a: Capacity of Impartation of source of knowledge has positive correlation with satisfaction of knowledge transfer
	H4b: Capacity of Impartation of source of knowledge has positive correlation with frequency of knowledge transfer
Capacity of Absorption	H5: Capacity of Absorption of recipient of knowledge has correlation with performance of knowledge transfer
	H5a: Capacity of Absorption of recipient of knowledge has positive correlation with satisfaction of knowledge transfer
	H5b: Capacity of Absorption of recipient of knowledge has positive correlation with frequency of knowledge transfer
Incentive Mechanism	H6: Incentive Mechanism of knowledge has correlation with performance of knowledge transfer
	H6a: Incentive Mechanism has positive correlation with satisfaction of knowledge transfer
	H6b: Incentive Mechanism has positive correlation with frequency of knowledge transfer
Culture	H7: Organizational culture has correlation with performance of knowledge transfer
	H7a: Organizational culture has positive correlation with satisfaction of knowledge transfer
	H7b: Organizational culture has positive correlation with frequency of knowledge transfer
Technical Support	H8: Technical Support of knowledge transfer has correlation with performance of knowledge transfer
	H8a: Technical Support of knowledge transfer has positive correlation with satisfaction of knowledge transfer
	H8a: Technical Support of knowledge transfer has positive correlation with frequency of knowledge transfer
Trust	H9: Trust relationship between transmitter and recipient of knowledge has correlation with performance of knowledge transfer
	H9a: Trust relationship between transmitter and recipient of knowledge has positive correlation with satisfaction of knowledge transfer
	H9b: Trust relationship between transmitter and recipient of knowledge has positive correlation with frequency of knowledge transfer
Knowledge Distance	H10: Knowledge Distance between transmitter and recipient of knowledge has curve correlation with performance of knowledge transfer
	H10a: Knowledge Distance between transmitter and recipient of knowledge has curve correlation with staff's satisfaction of knowledge transfer When knowledge distance is short, the correlation appears negative; as it becomes moderate, the correlation appears positive and when knowledge distance is considerable long, and the correlation becomes negative again.
	H10b: Knowledge Distance between transmitter and recipient of knowledge has curve correlation with staff's frequency of knowledge transfer. When knowledge distance is short, the correlation appears negative; as it becomes moderate, the correlation appears positive and when knowledge distance is considerable long, the correlation becomes negative again.
Control Variables	H11: The individual's working years, position and qualification have influence on performance of knowledge transfer

3.3 Result Analysis and Explanation

3.3.1 Statistical Result Analysis

In order to ensure the scientific nature of the proposition certification, it is necessary to test the reliability and validity of the measure model. First, all variables' Cronbach's coefficient are significantly higher than the minimum threshold 0.70, factor analysis and confirmatory factor analysis are all met reference standards, so we can judge it has internal validity. Second, for all indicators, standardized loading factors are also higher than that of the recommended minimum critical level 0.50, the statistical value of Battelle is a much smaller than 0.01. All of these indicate that all scales have highly convergent validity. The above shows that the research ha good external validity. Integrated the test of reliability and validity, the scales are reliable and effective which can be used to verify model assumptions.

Correlation analysis was executed between knowledge transfer performance and its influence factors by using SPSS16.0 statistical software, and the result was presented in Table 2. The result concludes that culture has no significant correlation to performance of knowledge transfer. Capacity of impartation and incentive mechanism has significantly positive correlation to performance of knowledge transfer at the significant level of 0.05. While ambiguity has significantly negative correlation to performance of knowledge transfer at the significant level of 0.01, the rest factors have significantly positive correlation to performance of knowledge transfer at the significant level of 0.01. To avoid multiple co-linear problems, we defined values of knowledge

transfer influence factors as independent variables, and executed regression analysis by step regression method when considering the causal relationship among knowledge transfer performance and its influence factors. The results are in Table 3.

From the overall regression effect with F=18.967, P= 0.000, regression equation has achieved a very significant level which indicates better regression effect. Meanwhile, variance expansion factor VIF had small value and multiple co-linear problems were not obvious. The adjusted determination coefficient is 0.339 indicating that three indicators of knowledge transfer influence can explain 33.9% of the total variance. The coefficient values of constant and variables are all less than 0.05 suggesting that they have significant meaning.

Table 2. Correlation analysis between influencing factors and performance of knowledge transfer

Performance / Influencing Factors	Performance of Knowledge Transfer	
	Correlation	Sig.
L1 Ambiguity	-.143*	.000
L2 Systematism	-.109**	.000
L3 Transfer Willingness	.375**	.000
L4 Capacity of Impartation	.285*	.0016
L5 Capacity of Absorption	.392**	.000
L6 Incentive Mechanism	.387**	.009
L7 Culture	213	.159
L8 Trust	.558**	.000

Table 3. Overall effect parameter by step regression

Model	R	R^2	Adjusted R^2	Standard Error	F	Sig.
1	.512	.262	.255	.625	37.004	.000
2	.560	.313	.300	.606	23.495	.000
3	.598	.358	.339	.589	18.967	.000

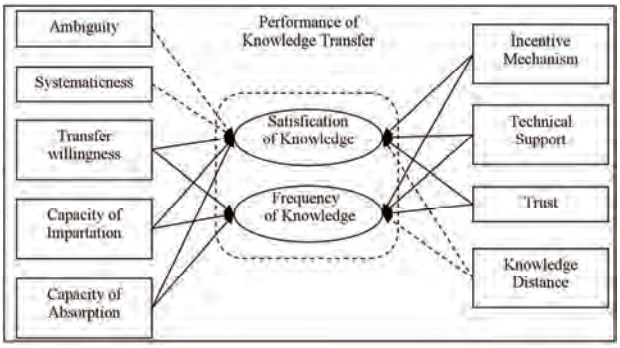


Figure 3. Influencing relationship of performance of knowledge transfer in software process

In order to clear the influence factors and their direction of the performance of knowledge transfer in SPI, we employed Figure 3 to express test results of Table 4 where the line indicated positive relationship and the dotted line negative relationship. From Table 4 and Figure 3, we can conclude that the majority results of empirical research are consistent with our hypotheses and they are listed as follows: In the characteristics of knowledge, the ambiguity impacting on the performance of knowledge transfer mainly manifested on the satisfaction of transfer. The more ambiguous the source of knowledge is, the more time and energy will be spending when we express the knowledge out from source of knowledge. The systematization of knowledge impacting on the performance of knowledge transfer mainly indicates that it has a negative effect on the performance of knowledge transfer. The higher the systematism of knowledge is, the more difficult for transmitter of knowledge to express real meaning of the knowledge.

3.3.2 Empirical Result Analysis

According to the above statistical results, the test results of our hypotheses were concluded in Table 4.

In the characteristics of source knowledge, the transfer willingness of knowledge has significantly positive correlation to the satisfaction and frequency of knowledge transfer. In the characteristics of recipients of knowledge, the recipients' capacity of absorption has significantly positive influence on the effect of knowledge transfer.

In the environmental factors of knowledge transfer in SPI, the incentive mechanism has significantly positive influence on the satisfaction and frequency of knowledge transfer. The employees of the organization will retain their knowledge because they are worried about losing their authority when imparting their knowledge to others, if the organization does not take certain incentives.

In the relationship between source of knowledge and recipient of knowledge, trust relationship has the largest impact on the performance of knowledge transfer of all influence factors. This shows that trust relationship between source of knowledge and recipient of knowledge is the most basic factor of knowledge transfer in SPI. The curve relationship between knowledge distance and the performance of knowledge transfer is not obvious.

4. Management Enlightment and Suggestion

In order to improve the staffs' performance of SPI in practice, the following three aspects are necessary.

1) Establishing a reasonable incentive mechanism. Software organizations should build a reasonable incentive mechanism to enhance the transfer willingness of source of knowledge and consciousness of recipients of knowledge, it can also promote trust relationship between the source of knowledge and recipient of knowledge. To enhance the transfer willingness of source of knowledge, it is necessary to give corresponding material

Table 4. Test results of hypotheses (a=10, b=11)

Hypothesis	Results	Hypothesis	Results
H1	Support	H6a	Support
H1a	Support	H6b	Support
H1b	Not	H7	Not Support
H2	Support	H7a	Not Support
H2a	Support	H7b	Not Support
H2b	Not	H8	Support
H3	Support	H8a	Support
H3a	Support	H8b	Support
H3b	Support	H9	Support
H4	Support	H9a	Support
H4a	Support	H9b	Support
H4b	Not	HA	Not Support
H5	Support	HAa	Not Support
H5a	Support	HAb	Not Support
H5b	Support	HB	Not Support
H6	Support		

or mental compensation and think highly of achievement of source of knowledge. Similarly, recipient of knowledge also need some encouragement to accept and use new knowledge. It can make the two sides of knowledge transfer participate actively by sharing their interests and therefore promote organizational SPI.

2) Carrying out necessary training for the both sides of knowledge transfer. After solving the transfer willingness of the two sides, the transfer capacity of source of knowledge, capacity of absorption of recipient of knowledge and distance between source of knowledge and recipient of knowledge have greater influence on knowledge transfer. Thus, software organizations need to give corresponding training for both of the two parties. They can employ external experts to train their staffs so that the staffs' capacity of imparting knowledge can gain improvement.

3) Using software benchmarking [19]. In pursuit of a capability model rating, software benchmarking (in our understanding, the benchmarking is standard best knowledge patterns, such as CMMI and SWEBOK [20], etc.) would help its process improvement and assessment effort. This benchmarking questionnaire can be grouped into five categories: a) Philosophy of implementation—how each company achieved CMMI compliance in terms of schedule, teams, and planning. b) Management commitment—the strength of institutional support for the process improvement effort. c) Cultural change and institutionalization—issues that arose regarding acceptance of the new process philosophy. d) Definition of organization—because the CMMI assessment is for specific organizations, these questions assessed the scope of their effort (for example, a section, company, or corporation). e) Objective evidence—the CMMI assessment process requires objective evidence that the new process is being followed, so these questions probed how each company collected evidence.

5. Conclusions

In this study, the knowledge transfer model of SPI and the conceptual framework of 10 key influence factors are established. Then research hypotheses are put forward. Empirical study concludes that the trust relationship among SPI staffs has the greatest influence on knowledge transfer, and organizational incentive mechanism can produce positive effect to knowledge transfer of SPI. We believed that the research is helpful for SPI practitioners to improve their performance of knowledge transfer.

6. Acknowledgements

Thanks for helpful discussion with Mr. Hou Yawen, Mr. Zhou qiyang, Mr. Li Jiangzhang, Mr. Nihao, Mr. Zhou Zhijun, and the hard work of my student Zeng Yonghua and Zheng Chuwei.

REFERENCES

[1] W. S. Humphrey, "Managing the software process," Reading, Addison-Wesley, MA, 1989.

[2] Alfonso Fuggetta, "Software process: A roadmap," Proceedings of the Conference on The Future of Software Engineering, Limerick, Ireland, pp. 25–34, June 04–11, 2000.

[3] J. P. Wan and J. M. Yang, "Knowledge management in SPI," Application Research of Computer, Vol. 19, No. 5, pp. 1–3, 2002.

[4] M. Blanco, P. Gutiérrez, and G. Satriani, "SPI patterns: Learning from experience," IEEE Software, Vol. 18, No. 3, pp. 28–35, 2001.

[5] I. Nonaka and G.g von Krogh, "Perspective-tacit knowledge and knowledge conversion: Controversy and advancement in organizational knowledge creation theory," Organization Science, Vol. 20, No. 3, pp. 635–652, 2009.

[6] J. P. Wan, "Research on software product support structure," Journal of Software Engineering and Applications, Vol. 2, No. 3, pp. 174–194, 2009.

[7] S. A. Slaughter and L. J. Kirsch, "The effectiveness of knowledge transfer portfolios in SPI: A field study," Information Systems Research, September 2006.

[8] X. G. Zhang, "Research on knowledge management technology in SPI," Chinese Academy of Sciences Doctoral Thesis, 2004.

[9] J. M. Bloodgood and W. D. Salisbury, "Understanding the influence of organizational change strategies on information technology and knowledge management strategies," Decision Support Systems, Vol. 31, No. 1, pp. 55–69, 2001.

[10] L. Argote, "Organizational learning: Creating, retaining and transfering knowledge," Kluwer Academic Publishers, pp. 143–189, 1999.

[11] T. H. Davenport and L. Prusak, "Working knowledge:

How organization manage what they know," Harvard Business School Press, Boston, 1998.

[12] G. Szulanzki, "Exploring internal stickiness: Impediments to the transfer of best practice with the firm," Strategic Management Journal, Vol. 17, pp. 27–43, 1996.

[13] L. Argote and P. Ingram, "Knowledge transfer: A basis for competitive advantage in firms," Organizational Behavior and Human Decision Processes, Vol. 82, No. 1, pp. 150–169, 2000.

[14] D.-G. Ko, L. J. Kirsch, and W. R. King, "Antecedents of knowledge transfer from consultants to clients in enterprise system implementations," Management Information System Quarterly (Special Issue), Vol. 29, No. 1, pp. 59–85, 2005.

[15] I. Nonaka and H. Takeuchi, "The knowledge creating company," Oxford University Press, New York, 1995.

[16] M. Gilbert and M. Cordey-Hayes, "Understanding the process of knowledge transfer to achieve successful technological innovation," Technovation, Vol. 16, pp. 301–312, 1996.

[17] J. L. Cummings and B. S. Teng, "Transferring R & D knowledge: The key factors affecting knowledge transfer success," Journal of Engineering and Technology Management, Vol. 20, pp. 39–68, 2003.

[18] M. Polanyi, "The study of man," Routledge & Kegan, London, Vol. 12, 1957.

[19] G. C. Thomas and H. R. Smith, "Using structured benchmarking to fast-track CMM process improvement," IEEE Software, Vol. 18, No. 5, pp. 48–52, 2001.

[20] P. Bourque, R. Dupuis, A. Abran, J.W. Moore, and L. Tripp, "The guide to the software engineering body of knowledge," IEEE Software, Vol. 16, No. 6, pp. 35–44, 1999.

J. Software Engineering & Applications, 2010, 3: 141-149
doi:10.4236/jsea.2010.32018 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A Novel Spatial Clustering Algorithm Based on Delaunay Triangulation

Xiankun Yang[1,2], Weihong Cui[1]

[1]Institute of Remote Sensing Application, Chinese Academy of Sciences, Beijing, China; [2]Graduate University of Chinese Academy of Sciences, Beijing, China.
Email: xiankungis@163.com.

Received August 17th, 2009; revised September 16th, 2009; accepted November 13th, 2009.

ABSTRACT

Exploratory data analysis is increasingly more necessary as larger spatial data is managed in electro-magnetic media. Spatial clustering is one of the very important spatial data mining techniques which is the discovery of interesting relationships and characteristics that may exist implicitly in spatial databases. So far, a lot of spatial clustering algorithms have been proposed in many applications such as pattern recognition, data analysis, and image processing and so forth. However most of the well-known clustering algorithms have some drawbacks which will be presented later when applied in large spatial databases. To overcome these limitations, in this paper we propose a robust spatial clustering algorithm named NSCABDT (*Novel Spatial Clustering Algorithm Based on Delaunay Triangulation*). *Delaunay diagram is used for determining neighborhoods based on the neighborhood notion, spatial association rules and collocations being defined. NSCABDT demonstrates several important advantages over the previous works. Firstly, it even discovers arbitrary shape of cluster distribution. Secondly, in order to execute NSCABDT, we do not need to know any priori nature of distribution. Third, like DBSCAN, Experiments show that NSCABDT does not require so much CPU processing time. Finally it handles efficiently outliers.*

***Keywords*:** *Spatial Data Mining, Delaunay Triangulation, Spatial Clustering*

1. Introduction

Data mining is a process to extract implicit, nontrivial, previously unknown and potentially useful information (such as knowledge rules, constraints, regularities) from data in databases [1,2]. The explosive growth in data and databases used in business managements, government administration, and scientific data analysis has created a need for tools that can automatically transform the processed data into useful information and knowledge [3]. Spatial data mining as a subfield of data mining refers to the extraction from spatial databases of implicit knowledge, spatial relations or significant features or patterns that are not explicitly stored in spatial databases [4]. It is concerned with the discovery of spatial relationships and intrinsic relationships between spatial and non-spatial data. With the large amount of spatial data obtained from satellite images and geographic information systems (GIS), it is an inevitable task for humans to explore spatial data in detail. Spatial datasets and patterns are abundant in many application domains related to the Environmental Protection Agency, the National Institute of standards and Technology, and the Department of Transportation. Challenges in spatial data mining arise from the following issues [3,5]. Firstly, classical data mining is designed to process numbers and categories. In contrast, spatial data is more complex and includes extended objects such as points, lines and polygons. Secondly, classical data mining works with explicit inputs, whereas, spatial predicates and attributes are often implicit. Finally, classical data mining treats each input independently of other inputs, while spatial patterns often exhibit continuity and high autocorrelation among nearby features.

Clustering is the process of grouping a set of objects into classes or clusters so that objects within a cluster have similarity in comparison to one another, but are dissimilar to objects in other clusters. So far, many clustering algorithms have been proposed. They differ in their capabilities, applicability and computational requirements. Based on a general definition, they can be categorized into five broad categories, i.e., hierarchical, partitional, density-based, grid-based and model-based [4]. 1) Partitional clustering methods [6], for example,

CLARANS. It classifies data into some groups, which together satisfy the following requirements: firstly, each group must contain at least one object; secondly, each object must belong to exactly on group. It is noticed that the second requirement can be relaxed in some fuzzy partitioning techniques. 2) Hierarchical clustering methods [7,8], such as DIANA [9] and BIRCH [8]. A hierarchical method creates a hierarchical decomposition of a given set of data objects. Hierarchical methods can be classified as agglomerative (bottom-up) or divisive (top-down), based on how the hierarchical decomposition is formed. 3) Density-based clustering methods. Their general idea is to continue growing a given cluster as long as the density (the number of objects or data points) in the "neighborhood" exceeds a threshold. Such a method is able to filter out noises (outliers) and discover clusters of arbitrary shape. Examples of density-based clustering methods include DBSCAN [10], OPTICS [11], GDBSCAN [12] and DBRS [13]. 4) Grid-based clustering methods, such as STING [14] and WaveCluster [15]. Grid-based methods quantize the object space into a finite number of cells that form a grid structure. All of the clustering operations are performed on the grid structure (i.e., on the quantized space). The main advantage of this approach is its fast processing time, which is typically independent of the number of data objects and dependent only on the number of cells in each dimension in the quantized space. 5) Model-based clustering methods. For example, COBWEB. It is clearly that no particular clustering method has been shown to be superior to all its competitors in all aspects. Typically, the problem is that clusters identified with one method cannot be detected by other methods [16]. This is because that many clustering methods need user-specified arguments or prior knowledge to produce their best results. Such information needs are supplied as density threshold values, merge/split conditions, number of parts, prior probabilities, assumptions about the distribution of continuous attributes within classes, and/or kernel size for intensity testing, for example, grid size for raster-like clustering [17] and radius for vector-like clustering [18]. This parameter-tuning is expensive and inefficient for huge data sets because it demands several trial and error steps.

Clustering based on Delaunay triangulation is not a new and has been described in some papers [16, 19, 20, 21]. Kang *et al* [14] proposed a clustering algorithm that utilizes a Delaunay triangulation; however, there is a need in the algorithm to provide a global argument as a threshold to discriminate perimeter values or edges lengths. As a result, the algorithm is not able to detect local variations. The first non-parametric clustering algorithm based on the Delaunay diagram, called AMOEBA, has been proposed in Estivill-Castro and Lee [16]. It overcomes some of the problems of the static approaches that required a distance threshold to be specified, but still fails to find relatively sparse clusters in certain situations. An upgraded version of AMOEBA, called AUTOCLUST, has been proposed by the same authors in Estivill-Castro and Lee [21].

But these methods also have some drawbacks. For example, if two clusters are mixed or connected by bridges, this methods described above cannot detect all the two clusters as shown in Figure 1. In this paper we propose a robust spatial clustering algorithm named NSCABDT (Novel Spatial Clustering Algorithm Based on Delaunay Triangulation). NSCABDT uses the Delaunay triangulation as analysis source, because Delaunay triangulation is a structure that is linear in the size of the data set and implicitly contains vast amounts of proximity information. That is to say, we can use the graph information of Delaunay triangulation and metric information to obtain remarkable robust clustering. In this study, we first construct a graph, and record the information of the graph as presented in Section 3. In the graph, vertices represent data points and edges connect pairs of points to model spatial proximity or interactions and all clustering operations are performed on the graph information.

The remainder of the paper is organized as follows: In Section 2, we will give an introduction to data preprocessing for NSCABDT. And Section 3 presents the NSCABDT algorithm. Section 4 reports the experimental evaluation. Finally, Section 5 concludes the paper.

2. Definitions and Notions

2.1 Spatial Clustering methods

Geographic data often show properties of spatial dependency and spatial heterogeneity [22]. Spatial dependency is a tendency of observations located close to one another in the geographical space to show a higher degree of similarity or dissimilarity (depending on the phenomenon under study). Closeness can be defined very generally—through distance, direction and/or topology. Spatial heterogeneity or inconsistency of the process with respect to its location is often visible, while many geographic

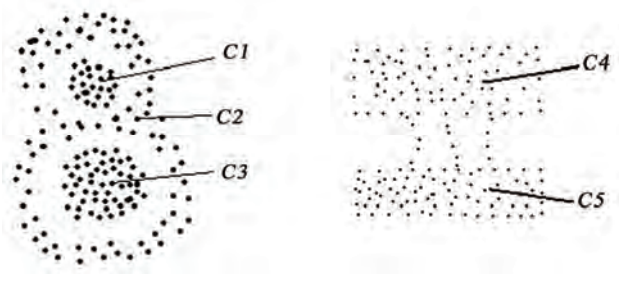


Figure 1. Three very dense clusters (*C*1, *C*2, *C*3), but most of clustering methods cannot detect the cluster *C*2. Because of the bridges between cluster *C*4 and cluster *C*5, the two clusters often are incorrectly thought to be one cluster

processes have a local character. Spatial dependency and heterogeneity can reflect the nature of the geographic process. Central to spatial data mining is clustering, which seeks to identify subsets of the data having similar characteristics. Two-Dimensional clustering is the non-trivial process of grouping geographically closer points into the cluster. Therefore, a model of spatial proximity for a discrete point-data set $P = \{p_1, \cdots, p_n\}$ must provide robust answers to which are the neighbors of a point p_i and how far the neighbors relative to the context of the entire data set P. A cluster is a group of objects, which are homogeneous among themselves. Clustering has been identified as one of the fundamental problems in the area of knowledge discovery and data mining, and it is of particular importance for spatial data sets. A distinct characteristic of spatial clustering for data mining applications is the huge size of the data files involved [23]. As Tobler's famous proposition [24] states: "Everything is related to everything else, but near things are more related than distant things." Thus proximity is pretty critical to spatial analysis and in spatial settings; clustering criteria almost invariably makes use of some notions of proximity, usually based on the Euclidean metric, as it captures the essence of spatial autocorrelation and spatial association [14].

$$d(X_j, Z_j) = \sqrt{(x_{i1} - x_{j1})^2 + (x_{i2} - x_{j2})^2 + \cdots + (x_{ip} - x_{jp})^2} \quad (1)$$

$d(X_j, Z_j)$is the Euclidean metric. And $(x_{i1}, x_{i2}, \cdots, x_{ip})$ and $(x_{j1}, x_{j2}, \cdots, x_{jp})$ are two $p(p \geq 2)$ dimensions data objects.

We assume that $S = \{p_0, p_1, p_2, \cdots p_{n-1}\}$ is a set of n data items in the $m-$ dimensional real space $\Re^m$. A cluster is a collection of S that is similar to one another within the same cluster and is dissimilar to the objects in other clusters. A cluster of data objects can be treated collectively as one group. We assume that C_k is a cluster of S and $C = \{C_1, C_2, \cdots, C_k\}$ is the collection of $C_i (1 \leq i \leq k)$, then:

$$S = \bigcup_{j=i}^{p} C_j \quad (2)$$

$$C_j \neq \emptyset \quad (j = 1,2,...,k) \quad (3)$$

$$C_i \bigcap C_j = \emptyset \quad (i, j = 1,2,..., j; i \neq j) \quad (4)$$

If Z_j is the clustering center of $C_j (1 \leq j \leq k)$, then:

$$J = \sum_{j=1}^{k} \sum_{s_j \in C_j} d(p_j, Z_j) \quad (5)$$

where J should be minimal.

2.2 Delaunay Triangulations

In mathematics, and computational geometry, a Delaunay triangulation for a set S of points in the plane is a triangulation $D(S)$ such that no point in S is inside the circumcircle of any triangle in $D(S)$. Delaunay triangulations maximize the minimum angle of all the angles of the triangles in the triangulation; they tend to avoid skinny triangles. The triangulation was invented by Boris Delaunay [25]. Delaunay triangulations have been widely used in a variety of applications in geographical information systems (GIS). Using Delaunay triangulations, it is simpler to tackle the problems associated with spatial topology automated contouring, two-and-half dimensional (2.5-D) visualization, surface characterization and reconstructions, and site visibility analyses on terrain surfaces.

Given the set of data points $S = \{p_0, p_1, \cdots, p_{n-1}\}$ in the plane, the Voronoi region of $p_i \in S$ is the locus of points (not necessarily data points) which have p_i as a nearest neighbor; that is, $\{x \in \Re^2 \mid \forall j \neq i, d(x, p_i) \leq d(x, p_j)\}$. Taken together, the n Voronoi regions of S form the Voronoi diagram of $\boldsymbol{S}$ (also called the Dirichlet tessellation or the proximity map). The regions are (possibly unbounded) convex polygons, and their interiors are disjoint [23]. Based on Delaunay's definition [25], the circumcircle of a triangle formed by three points from the original point set is empty if it does not contain vertices other than the three that define it (other points are permitted only on the very perimeter, not inside).

The Delaunay triangulation $D(S)$ of S is a planar graph embedding defined as follows: the nodes of $D(S)$ consist of the data points of S, and two nodes p_i, p_j are joined by an edge if the boundaries of the corresponding Voronoi regions share a line segment.

Delaunay triangulations capture in a very compact form the proximity relationships among the points of S. They have many useful properties, the most relevant to our application being the following:

1) If p_j is the nearest neighbor of p_j from among the data points of S, then $< p_i, p_j >$ is an edge in $D(S)$. That is to say, the 1-nearest-neighbor digraph is a subgraph of the Delaunay triangulation.

2) The number of edges in $D(S)$ is at most 3n -6.

3) The average number of neighbors of a site s_i in $D(S)$ is less than six.

Figure 2. A data set (n=15) and its Delaunay triangulation

4) The Delaunay triangulation is the most well proportioned over all triangulations of *S*, in that the size of the minimum angle over all its triangles is the maximum possible.

5) If p_i and p_j form a triangle in $D(S)$, then the interior of this triangle contains no other point of S.

6) The triangulation $D(S)$ can be robustly computed in $O(n\log n)$ time.

7) The minimum spanning tree is a subgraph of the Delaunay triangulation, and, in fact, a single-linkage clustering (or dendrogram) can be found in $O(n\log n)$ time from $D(S)$.

Figure 2 shows a set of 15 data points and its corresponding Delaunay triangulation. More information regarding Delaunay triangulations and Voronoi diagrams can be found in other literature. From Figure 2, we can conclude that, in a proximity graph like Delaunay triangulation (Delaunay diagram); the points are connected by edges, if and only if they seem to be close by some proximity measure [26]. By applying to this rule, if two points are connected by a small enough Delaunay edge, the two points belong to the same cluster.

3. Initialization Using the Delaunay Triangulations

3.1 Data Preprocessing

Given a set of data points $S=\{p_0, p_1, \cdots, p_{n-1}\}$ in the plane (as shown in Figure 2), $n=15$. The triangulations were computed by Bowyer-Watson algorithm in $O(n\log n)$ time. In the creation process of Delaunay triangulation, we recorded node, edge and surface information of Delaunay triangulation for clustering later. This nodes, edges and surfaces information was stored in Oracle database. Oracle database includes numerous data structures to improve the speed of SQL queries. Taking advantage of the low cost of disk storage, Oracle includes many new indexing algorithms that dramatically increase the speed with which Oracle queries are serviced. And, Oracle database includes so many statistical functions which include descriptive statistics, hypothesis testing, and correlations analysis, for distribution fit and so forth. The statistical functions in the database can be used in a variety of ways, for example, we can call Oracle's DBMS_STAT_FUNCS functions to obtain basic cont, mean, max, min and standard deviation information of Delaunay triangulation edges. For Figure 2, we got three tables as follows:

In the Table 1, the first column is the index of the points in S, the second column is X coordinate and the third column is Y coordinate respectively. The degree denotes the number of Delaunay edges which incident to a point. The "ClassType" column represents the category number after clustering process, and after clustering process if it is -1, we think the point is an outlier or noise.

In the Table 2, the second column is the index of the edge's starting point, and the third column is the index of the edge's end point. The length of edges is represented by the fourth column. In our algorithm, every edge is needed to be computed only once.

The chart illustrates the table structure and relationships of the three tables. The Delaunay triangulation node table contains all the spatial objects (points); the

Table 1. Delaunay triangulation nodes table

Index	X	Y	Degree	ClassType
1	3853964.924	-803305.9261	6	-1
2	3853985.696	-803331.6837	4	-1
3	3853998.714	-803330.2989	6	-1
4	3853994.005	-803335.8381	5	-1
5	3853992.066	-803329.7449	4	-1
6	3854013.393	-803354.3946	6	-1
7	3853989.297	-803371.0124	6	-1
8	3853990.128	-803377.6595	4	-1
9	3853996.221	-803375.7208	5	-1
10	3854049.121	-803327.2523	5	-1
11	3854045.52	-803337.4999	7	-1
12	3854058.538	-803336.669	4	-1
13	3854051.337	-803334.4533	3	-1
14	3854039.981	-803371.8433	4	-1
15	3854047.459	-803375.7208	4	-1

Table 2. Delaunay triangulation edge table

Index	Start	End	Length
1	1	8	76.03228669
2	8	7	6.69884313
3	7	1	69.50014083
4	1	7	69.50014083
5	7	2	39.49321264
6	2	1	33.08972562
7	1	2	33.08972562
8	2	5	6.65851675
9	5	1	36.11126413
10	1	5	36.11126413
……	……	……	……

Table 3. Delaunay triangulation surface table

In-dex	Node1	Node2	Node3	Edge1	Edge2	Edge3
1	1	8	7	1	2	3
2	1	7	2	4	5	6
……	……	……	……	……	……	……

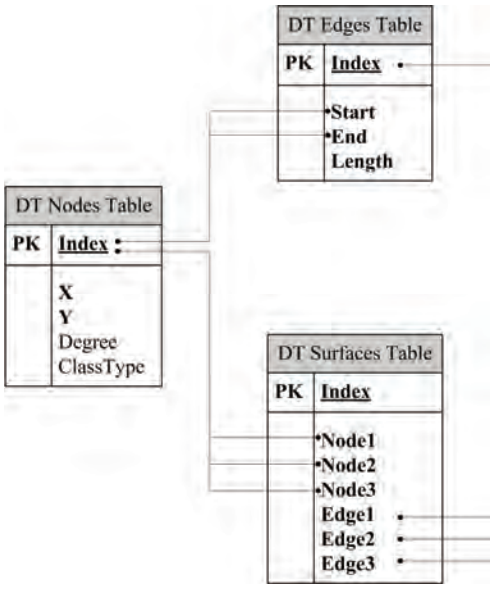


Figure 3. Table structure and relations in the database

Delaunay triangulation edge table includes all the Delaunay edges and the relationships with the Delaunay triangulation node table. And the Delaunay triangulation surface table records all the Delaunay triangulation surfaces and the relationships with Delaunay triangulation nodes table as well as Delaunay edge table.

3.2 Some Definitions and Notions in NSCABDT

Given a set of data points $S=\{p_0,p_1,\cdots,p_{n-1}\}$ in the plane (as shown in Figure 2), after data preprocessing, we got the nodes, edges and surfaces information of the Delaunay triangulation. Given a set of edges $E=\{e_0,e_1,\cdots,e_{n-1}\}$, for each edge $e_k(0\le k\le n-1)$ in E is a record of Delaunay triangulation edge table. For each edge $e_k<p_i,p_j>$, $(0\le k\le n-1)$ p_i is its starting point, and p_j is its end point. Both p_i and p_j belong to S. And $N(p_i)$ denotes a set of edges which incident to p_i.

Definition 1 (Local_Mean): We denote by mean (p_i) the mean length of edges in $N(p_i)$.

$$Local\_Mean(p_i)=\sum_{j=1}^{d(p_i)}Len(e_j)\Big/d(p_i) \quad (6)$$

where $d(p_i)$ denotes the degree of p_i in graph theory; and $Len(e_j)$ denotes the length of the Delaunay edge e_j.

Definition 2 (Global_Mean): We denote by mean S the mean length of edges in E.

$$Global\_Mean(S)=\sum_{j=1}^{sum(E)}Len(e_j)\Big/sum(E) \quad (7)$$

where $sum(E)$ is the number of edges in E.

Definition 3 (Global_Sta_Dev): We denote by global standard deviation of the lengths of all edges. That is,

$$Global\_Sta\_Dev(S)=\sqrt{\sum_{i=0}^{n}(Global\_Mean(S)-Len(e_i))^2\Big/n} \quad (8)$$

Definition 4 (Relative_Mean): We let $Relative\_Mean(p_i)$ denote the ratio of $Local\_Mean(p_i)$ and $Global\_Mean(S)$. That is,

$$Relative\_Mean(p_i)=Local\_Mean(p_i)/Global\_Mean(S) \quad (9)$$

Definition 5 (Positive Edge): If the length of a Delaunay edge is less than the given criterion function $F(p_i)$, the edge is a positive edge. Positive edges and points incident to them form a new proximity graph and the newly created graph is subgraph of the Delaunay graph (Delaunay Triangulation).

Definition 6 (Positive path): A path in current proximity graph where every edge in the path is a positive edge; and all the points connected by actives paths belong to a cluster.

Finally, this edge analysis is captured in a criterion function $F(p_i)$. The cut-off value for edges incident in p_i is defined as follows:

$$\begin{aligned}F(p_i)&=Global\_Mean(S)+Global\_Sta\_Dev(S)\\&\times Relative\_Mean(p_i)^{-1}\\&=Global\_Mean(S)+Global\_Sta\_Dev(S)\\&\times\frac{Global\_Mean(S)}{Local\_Mean(p_i)}\end{aligned} \quad (10)$$

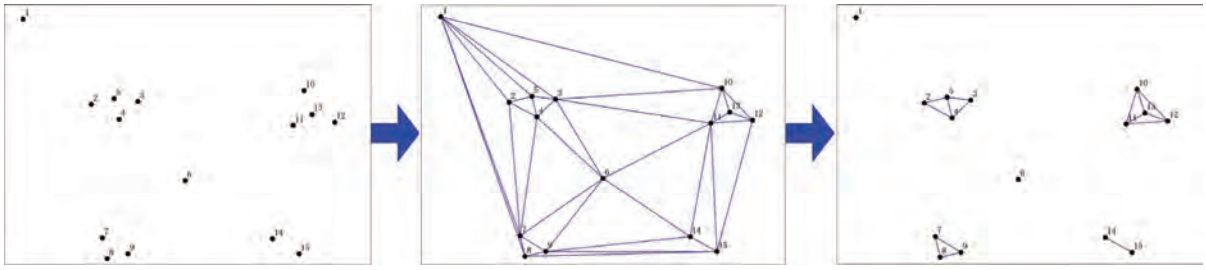


Figure 4. NSCABDT procedure

Definition 7 (Effective Region): For a point p in S, we call

$$p_\delta = \{x \,||\, p - x| < \delta, p \in S, \forall x \in R^2\} \quad (11)$$

The effective region of point p with respect to radius δ. As the union of the effective region of each point, we define the effective region of a random point set.
For a point set S, we call

$$\overline{S} = \bigcup_{p \in S} p_\delta \quad (12)$$

The effective region of point set S with respect to radius δ.

Definition 8 (γ – *boundary*): We define the boundary set as

$$V_\gamma = \overline{V}_\gamma \cap V, \quad \overline{V}_\gamma = \overline{V} \setminus \{\overline{V} \Theta \gamma Dm(\delta)\} \quad (13)$$

where $\gamma > 1$ is a constant and $Dm = \{x \,||\, x| \leq \delta\}$ is the set of all points in the circle with the radius δ. We call V_γ the γ – *boundary* of the point set S.

Definition 9 (γ – *curve*): The principal boundary of a random point set is the principal manifold of the point in the γ – *boundary* of a point set.

We also call this principal manifold extracted from point set S the γ – *curve* of S.

If the edges which incident to p_i are greater than or equal to $F(p_i)$, the edges are eliminated and the edges that are less than the criterion function survive. For each definition above, it is not necessary to iteratively calculate the results by programming; because we can get the results by using oracle statistical functions. For example, for Definition 1, a SQL statement can be created as follows: *SELECT avg(length) FROM edgetable WHERE start* = p_i, where "edgetable" is the table which restores the edges information of Delaunay triangulation.

We now present the algorithm of NSCABDT:

Initialize the points of a data points set S as being assigned to no cluster; Initialize an empty data set C;

1) Create Delaunay triangulation and record the information of Delaunay triangulation in Oracle database.
2) For each node p_i in Delaunay triangulation, extract edges $N(p_i)$ incident to node p_i via SQL queries and calculate $Local\_Mean(p_i)$ as well as $F(p_i)$.
3) For each edge e in $N(p_i)$, if $Len(e) \geq F(p_i)$, the edge will be deleted.
4) After 3, if $d(p_i) = 0$, the node p_i will be deleted, otherwise, the node p_i is added to C.
5) Using the same method, iteratively calculate all the nodes which connect with p_i.
6) Extract the boundary of the cluster C and eliminate the bridges.
7) If all the points have been not processed, end the process. Otherwise, initialize a new empty data set C, go to next un-processed node.

Phase 1 of NSCABDT is the construction of Delaunay triangulation. Then, recursively, all points in a connected component are reported as a cluster. Thus every edge is tested for the criterion function only once. After eliminating no-interesting edges and noises, only positive edges are remaining. According to the positive path, we can iteratively find all the points connected by positive paths and add the points to a cluster.

Obviously, it can be seen from the Figure 4 that it consists of two phases; the first phase is building Delaunay Triangulations from spatial objects. And on the second phase, we eliminate all edges in the way which we introduced above. And then, we got that point 1, point 6, point 14 and point 15 are outliers.

In order to eliminate the bridges between two different clusters, a detection of cluster boundary is executed; the algorithm is according to [5]. The boundary of a point set is extracted by the principal curve analysis. The principal curve analysis is a generalization of principal axis analysis, which is a standard method for data analysis in pattern recognition. For a cluster, if we can get two different boundaries, we think there are two smaller clusters in the point set, and bridges exist between the two smaller clusters. If an edge is not in the boundaries, the edge should be deleted.

The algorithm for eliminating the bridges between two

different clusters is as follows:

1) For the collection of all edges *E*, get the median length via SQL queries. And set it as δ.
2) Get the effective region of random point set V.
3) Get $\gamma-boundary$ of random point set V.
4) Get $\gamma-curve$ of random point set V.

Although, the construction of Delaunay triangulation using all points in *S* is a time-consuming process for a large number of points even if we use an optimal algorithm, we can use the information stored in the database instead of the construction of Delaunay triangulation again and we also can get the median length via SQL queries. Obviously, it is more efficient.

4. Experimental Results

We evaluate NSCABDT according the three major requirements for clustering algorithms on large spatial databases as stated above. We compare NSCABDT with the clustering algorithm DBSCAN in terms of effectivity and efficiency. The evaluation is based on an implementation of NSCABDT in .NET 2005. All the experiments were run on Windows Server 2003.

4.1 Discovery of Clusters with Arbitrary Shape

Clusters in spatial databases may be of arbitrary shape, e.g. spherical, drawn-out, linear, elongated etc. Furthermore, the databases may contain noise [27]. We used visualization to evaluate the quality of the clusterings obtained by the NSCABDT. In order to create readable visualizations without using color, in these experiments we used small databases. Due to space limitation, we only present the results from one typical database which was generated as follows:

1) Draw three polygons of different shape (one of them with a hole) for three clusters.
2) Generate 500, 200 and 200 uniformly distributed points in each polygon respectively.
3) Insert 100 noise points into the database, which is depicted in Figure 5.

For NSCABDT, we set 10% noise for the sample database. NSCABDT discovers all clusters and detects the noise points from the sample database. The clustering result of NSCABDT on this database is shown in Figure 7. Different clusters are depicted using different symbols and noise is represented by crosses. This result shows that NSCABDT assigns nearly all points to the correct clusters.

4.1 Efficiency

It has been proved that DBSCAN has better performancethan partitioning and hierarchical algorithms for spatial data mining, so we only compare our algorithm with DBSCAN [28]. In the following, we compare NSCABDT with DBSCAN with respect to efficiency on synthetic databases. The run time and correct rate for NSCABDT, DBSCAN on these test databases are listed in Table 4.

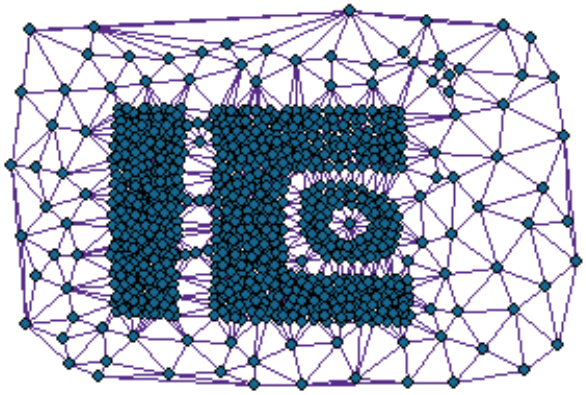


Figure 5. A data set and its Delaunay triangulation (n=1000)

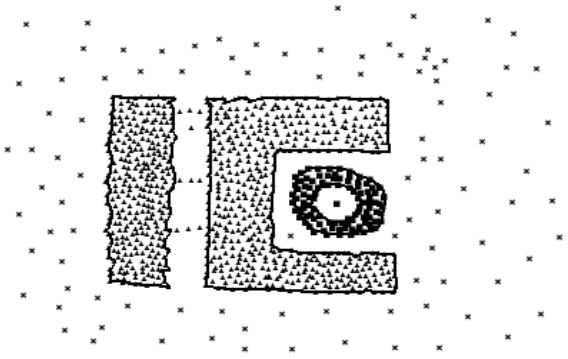


Figure 6. Extract the boundary of the cluster and eliminate the bridges

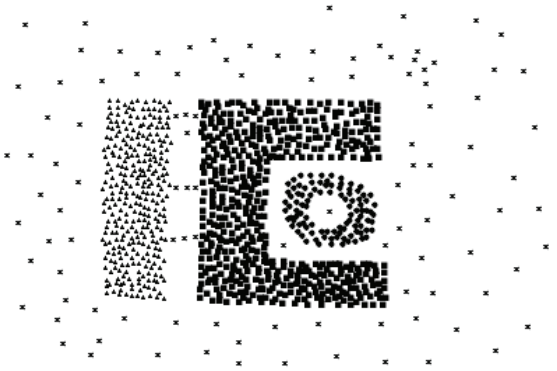


Figure 7. Clustering by NSCABDT. Finally we got 3 clusters

We generated some large synthetic test databases with 5000, 6000, 7000, 8000, 9000 and 10000 points to test the efficiency and scalability of DBSCAN and NSCABDT. We can conclude that NSCABDT is significantly slower than DBSCAN (see Figure 8), but the correct rate of NSCABDT is higher than DBSCAN (see Figure 9).

Because our approach does not require any assumptions or declarations concerning the distribution of the

Table 4. Run time in seconds

Number of Points	5000		6000		7000		8000		9000		10000	
	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time	Correct rate	Run time
DBSCAN	97.8%	36.7	97.2%	52.8	97.4%	72.6	97.9%	93.7	97.5%	121.5	97.8%	154.6
NSCABDT	99.7%	48.2	99.8%	71.4	99.8%	93.8	99.7%	117.9	99.7%	151.3	99.7%	189.4

data, the parameters of DBSCAN is difficult to be fixed. If the parameters are not inappropriate, the correct rate will be very low. DBSCAN must continually ask for assistance from the user. The reliance of DBSCAN on user input can be eliminated using our approach.

5. Conclusions

The application of clustering algorithms to large spatial databases raises the following requirements [27]: 1) minimal number of input parameters, 2) discovery of clusters with arbitrary shape and 3) efficiency on large databases. The well-known clustering algorithms offer no solution to the combination of these requirements.

In this paper, we introduce the new clustering algorithm NSCABDT. Our notion of a cluster is based on the distance of the points of a cluster to their neighbors. The neighboring region formed in our algorithm reflects the neighbor's distribution. Experimental results demonstrated that our clustering algorithm can provide significant improvement of accuracy of the cluster detecting, especially for objects with arbitrary and linear distribution.

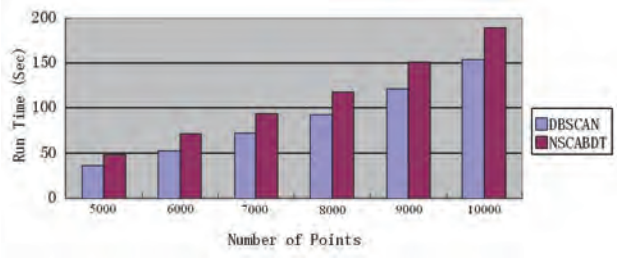


Figure 8. Efficiency: SCABDT VS DBSCAN

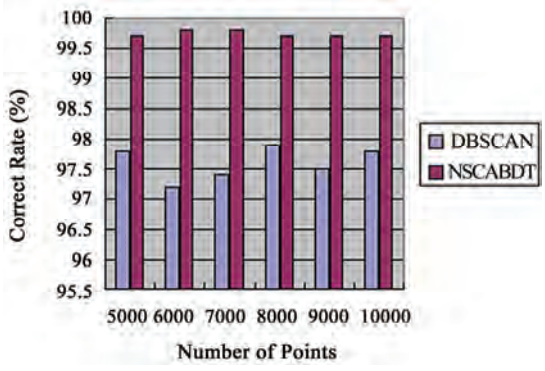


Figure 9. Correct rate: SCABDT VS DBSCAN

REFERENCES

[1] G. Piatetsky-Shapiro and W. J. Frawley. "Knowledge discovery in databases," AAAI/MIN Press, 1999.

[2] U. Fayyad, G. Piatetsky-Shapiro, and P. Smyth. "The KDD process for extracting useful knowledge from volumes of data," Communications of ACM, Vol. 39, 1996.

[3] S. Shekhar, C. T. Lu, P. Zhang, and R. Liu, "Data mining for selective visualization of large spatial datasets," Processing of 14th IEEE international conference on tools with artificial intelligence (ICTAI'02), 2002.

[4] J. Han and M. Kamber, "Data mining: Concepts and Techniques," Academic Press, 2001.

[5] I. Atsushi and T. Ken, "Graph-based clustering of random point set," Structural, Syntactic and Statistical Pattern Recognition, Springer Berlin, pp. 948–956, 2004.

[6] R. T. Ng and J. Han, "CLARANS: A method for clustering objects for spatial data mining," IEEE Transactions on Knowledge and Data Engineering, Vol. 14, No. 5, pp. 1003–1016, 2002.

[7] S. Guha, R. Rastogi, and K. Shim, "CURE: An efficient clustering algorithm for large databases," Proceedings of the 1998 ACM SIGMOD international conference on Management of data, ACM Press, pp. 73–84, 1998.

[8] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: An efficient data clustering method for very large databases," Proceedings of the 1996 ACM SIGMOD international conference on Management of data, ACM Press, pp. 103–114, 1996.

[9] L. Kaufman and P. J. Rousseeuw, "Finding Groups in Data: An introduction to cluster analysis," John Wiley & Sons, 1990.

[10] M. Ester, H. P. Kriegel, J. Sander, and X. Xu, "Density-based algorithm for discovering clusters in large spatial databases with noise," Proceedings of the 1996 Knowledge Discovery and Data Mining (KDD'96) international conference, AAAI Press, pp. 226–231, 1996.

[11] M. Ankerst, M. M. Breunig, H. P. Kriegel, *et al*., "OPTICS: Ordering points to identify the clustering structure," Proceedings of the International Conference on Management of Data (SIGMOD), ACM Press, pp. 49–60, 1999.

[12] J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Density-Based Clustering in Spatial Databases: The algorithm GDBSCAN and its applications," Data Mining and Knowledge Discovery, Vol. 2**,** No**.** 2, pp.169–194, 1998.

[13] X. Wang and H. J. Hamilton, "DBRS: A density-based spatial clustering method with random sampling," Proceedings of the 7th PAKDD, Springer, pp. 563–575, 2003.

[14] R. P. Haining, "Spatial data analysis in the social and environmental sciences," Cambridge University Press, 1990.

[15] J. Han and M. Kamber, "Data Mining: Concepts and techniques," Second Edition, Morgan Kaufmann, 2006.

[16] V. Estivill-Castro and I. Lee, "AMOEBA: Hierarchical clustering based on spatial proximity using delaunay diagram," Proceedings of the 9th international symposium on spatial data handling, pp. 7a. 26–7a. 41, 2000.

[17] E. Schikuta and M. Erhart, "The BANG-clustering system: Grid-based data analysis," Proceedings of the 2nd international symposium IDA-97, Advances in intelligent data analysis, Springer-Verlag, pp. 513–524, 1997.

[18] S. Openshaw, "A mark 1 geographical analysis machine for the automated analysis of point data sets," International Journal of GIS, Vol. 1, No. 4, pp. 335–358, 1987.

[19] In-Soo Kang, Tae-wan Kim, and Ki-Joune Li, "A spatial data mining method by delaunay triangulation," Proceeding of 5th ACM Workshop on Geographic Information Systems, Las Vegas, Nevada, pp. 35–39, 1997.

[20] C. Eldershaw and M. Hegland, "Cluster analysis using triangulation," Computational Techniques and Applications (CTAC97), World Scientific, Singapore, pp. 201–208, 1997.

[21] V. Estivill-Castro and I. Lee, "AUTOCLUST: Automatic clustering via boundary extraction for mining massive point-data sets," Proceedings of the 5th international conference on geocomputation, 2000.

[22] H. J. Miller, "Geographic data mining and knowledge discovery," Handbook of geographic information science. Malden, MA: Blackwell, pp. 149–159, 2009.

[23] V. Estivill-Castro and M. E. Houle., "Robust Distance-based clustering with applications to spatial data mining," Algorithmica, Vol. 30, No. 2, pp. 216–242, 2001.

[24] W. R. Tobler, "A computer movie simulating urban growth in the Detroit region," Economic Geography, Vol. 46, No. 2, pp. 234–240, 1970.

[25] B. Delaunay, "Sur la sphère vide, Izvestia Akademii Nauk SSSR, Otdelenie Matematicheskikh i Estestvennykh Nauk," pp. 793–800, 1934.

[26] G. Liotta, "Low degree algorithm for computing and checking gabriel graphs", Report No. CS–96–28, Department of Computer Science in Brown University, Providence, 1996.

[27] X. Xu, M. Ester, H. Kriegel, and J. Sander, "A distribution-based clustering algorithm for mining in large spatial databases," Proceedings of the 14th International Conference on Data Engineering (ICDE'98), pp. 324–331, 1998.

[28] D. Y. Ma and A. D. Zhang, "An adaptive density-based clustering algorithm for spatial database with noise," ICDM, Proceedings Fourth IEEE International Conference, pp. 467–470, 2004.

J. Software Engineering & Applications, 2010, 3: 150-159
doi:10.4236/jsea.2010.32019 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A Radar Visualization System Upgrade

Hector N. Acosta, Marcelo A. Tosini, María C. Tommasi, Lucas Leiva

INTIA Research Institute, INCA Group, Buenos Aires, Argentine
Email: {nacosta, mtosini, ctommasi, lleiva}@exa.unicen.edu.ar

Received September 9th, 2009; revised September 12th, 2009; accepted December 1st, 2009.

ABSTRACT

This work develops a system to visualize the information for radar systems interfaces. It is a flexible, portable software system that allows to be used for radars that have different technologies and that is able to be adapted to the specific needs of each application domain in an efficient way. Replacing the visualization and processing units on existing radar platforms by this new system, a practical and inexpensive improvement is achieved.

Keywords: *Radar, Data Visualization, Signal Processing, Signal Pattern Recognition*

1. Introduction

Conceptually, a radar system consists of five components: a transmitter, a receiver, an amplifier, an analyzer and a visualizer. This work describes the development of a software system to replace the last two components, in order to bring up to date the present services of different radars, air traffic control radars, tactical radars, navigation radars for military or civil use, both naval and aeronautic.

The analyzer of a radar system has to obtain the desired information from the received signals, and to determine if the reflections got through the antenna corresponds to targets of interest for the system. The analyzing components of modern systems carry out a great number of functions that allow to synthetize the desired information in an efficient and simple way, as noise minimize, prediction of target course, objective identification.

The visualizer component has to show the information processed by the analyzer on a screen or a display. This can be done in different ways depending on the needs of the system. For example, search or surveillance radars that cover 360º use to present information as a Plan Position Indicator (PPI) (Figure 1), which show targets in a polar form centered in the radar position, whereas tactical Air-to-Air radars, use presentations as "B-Scope" (Figure. 2), which put the different targets on the display according to distance information (vertical line) and azimuth (horizontal line) [1].

Radars are very expensive electronic equipment and as every technologic tool must be depreciated very soon because it should be replaced by new, more complex models. Companies are reluctant to change equipment, but operators intend to be up to date with new technology, by getting constant training. This situation puts a challenge between equipment being used and operators training. As a partial solution, this article introduces and analyzes the use of a generic parametrizable system that permits to update the radar in a rather practical and economical way. This approach proposes using the existing technologic platform and adding new functionality to provide services that approximate desired solutions avoiding expensive and sometimes inviable alternative of whole replace of the equipment.

In the following sections the design and development work of a radar signals visualization system is described. Then, the implementation of the system is given in detail as a case study adapted to a specific technological platform.

2. Architectural Design

A radar system is composed of five components: a transmitter, a receiver, an amplifier, an analyzer and a visualizer. As the system to be developed must replace and enhance the analyzing and visualizing components, it's important to describe some characteristics of the other components, in order to establish the operative conditions of the system and, thus, its requirements. The architectural design of the system is based on a detailed analysis of the functional and non-functional characteristics required.

2.1 Functional Characteristics

The system must be able to process signals that come from different devices, as well as from other software and/or hardware components. Signals processed by the system may be classified into the following types:

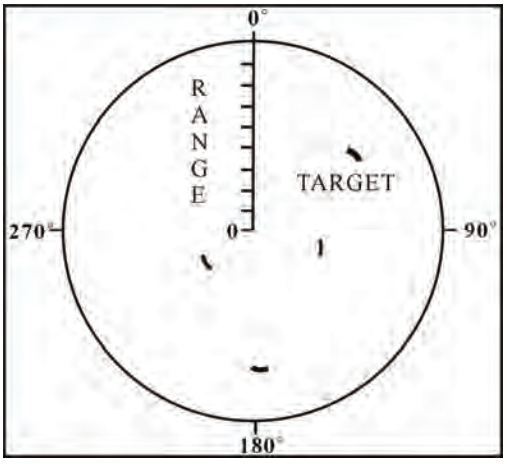


Figure 1. Presentation "Plan Position Indicator" (PPI) 1 CONICET

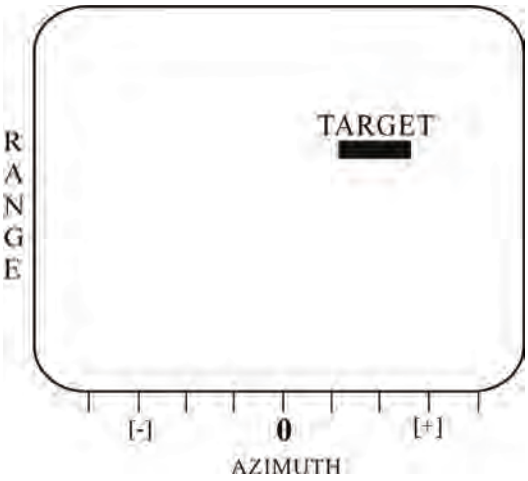


Figure 2. "B-Scope"

- Analogic and/or digital signals coming from the radar (antenna position, echo receiving, synchronization, operation modes of the radar).
- Analogic and/or digital signals coming from hardware devices external to the radar (GPS, timers, course and attitude indicators).
- Digital signals coming from independent software components that can communicate with the system for specific purposes (targets detection, geographic information system, navigation system, anti collision system).
- Signals coming from manual controls of the support device of the system (mainly keyboard and joystick).

Signals from the radar device are variable and have different characteristics. First of all, it is necessary to know the reflections values obtained from the antenna of the radar. Besides, those data must be processed along with the antenna position (which depends on the radar type and the operation modes provided) and synchronization information between times of sending and receiving, in order to achieve an adequate representation of the zone explored by the radar. Besides there are signals originating from specific functionalities provided by the radar, for example, targets tracking functions, used in air traffic control radar systems or tactical radars for military uses.

Often, modern radar systems communicate with other devices for specific purposes. In meteorological or air traffic control radars is very usual, for example, add to the information detected by the radar geographic information or satellital images of the explored area. In radars installed on ships or aircrafts it is of the utmost importance to have georeferencial navigation information, usually provided by some device or external system (GPS, for example). This is why, it is very important when designing and developing a system with these characteristics to have in mind that input signals can be distinct in nature and source.

Finally, it must be taken into account that modern radars have different modes of operation, in each of which the functioning of the device can be changed, altering, for example, the way the antenna moves, and thus, the exploration zone, or simply the information on the display. These changes in functioning parameters must be modified by the operator, through system controls.

As the result of signals processing, the system must show information about the objects detected by the radar on a display. This information depends on the application domain, nevertheless it can be generalized, so that the system presents a set of "objects" and "attributes", updating the changes of the objects and selecting those attributes that can be ignored or pointed out for presentation.

With regard to functionalities of interface, the system must provide a multifunction visualizing unit (MFD – Multi Function Display) to show graphical presentations flexible and easy to modify and adapt to the user's needs. It is essential for this purpose to be able to define different kinds of presentations and configuring their characteristics. So, the system must have ability to manage the iconography and graphical and textual information to visualize in different contexts in flexible way.

2.2 Non-Functional Characteristics

Basically, this system must carry out the acquisition, interpretation and presentation of information in real time. There are conditions that critically affect the performance of the system, which must be considered in the architectural design.

The system must fulfill hard performance requirements, reacting to user orders and updating all detected changes of the information to show on MFD with celerity. The display is an accurate representation of the zone being explored.

The system must also be updated easily in order to

adapt to changes, both regarding to the application domain and users' needs. It also must adapt in a simple way to process signals coming from new hardware devices or software components, to modify the algorithms for signal processing, to change or add presentation of information formats, among other important functionalities.

Another basic aspect is the technological independence. The software embedded in the system must be based on open source code (ANSI C) and have all the sources of every level of coexisting software (operating system, drivers, fonts, interfaces).

Finally, it is relevant to consider that the system must be implemented on a platform providing a high level of performance and synchronization.

2.3 Software Architecture

The characteristics desired in software architecture must ensure an easy evolution of the system. That evolution must consider changes in time, the impact of these changes in the different components and/or devices interacting with the system and the independence of functionalities of processing and visualization of the underlying technological platform [2].

Considering these objectives and the functional and non-functional characteristics above described, a software architecture based on three components was designed: The first is the acquisition component, which collects information that allows to control different devices and to gather analogical and digital signals. The second is the analyzing component, developed by different interpretation algorithms that allow identifying relevant information to select and sort the elements that must be visualized. The third is the visualizing component that allows information under different configurations to be charted. Thus, two components of the present radar (analyzer and visualizer) are replaced by a modern module composed by three components: acquisitor, analyzer and visualizer.

2.3.1 Acquisition Component

It implements the reading of the different signals that the system must process. It is mainly composed of control drivers for different boards and hardware devices of input signals.

The signals that must be processed come from different sources (radar antenna, timers, GPS, navigation system, manual controls), therefore, the system must synchronize the reception of signals.

2.3.2 Analyzing Component

It processes and analyzes a set of signals with different characteristics; some of which could need storage of historic information (tracking algorithms and path predictions, study of frequency for target recognizing). It receives a variable set of data as input that represent the different changes of input signals, and produces a set of data representing changes in the information to display on the MFD.

The functions that the system must implement depend directly on the application domain. There are many signal processing techniques for radar systems development on different domains.

Techniques based on Fourier Analysis have important application in many radar systems, a number of computational calculus methods have been developed as the Cooley-Tukey technique used for spectrum analysis and digital signal processing [1]. Generally, techniques based on Fast Fourier Transform (FFT) are used in pulse-doppler radars, radars for tracking mobile targets, vibration analysis on laser radar systems [3].

2.3.3 Pattern Recognition Component

Algorithms for pattern recognition based on artificial intelligence techniques are also used, neural nets and fuzzy logic [4–6]. On last generation radars, pattern recognition is used to classify known echos in different categories.

For tactical radars it is important to consider target tracking: only one target, multiple targets and "false alarms". These approaches use probabilistic and estimation techniques, Kalman filters, fuzzy logic, neural networks [7].

In this case, a pattern recognition system based on RBF networks is implemented

2.3.4 Visualizing Component

It implements all the functionalities of MFD, without processing or transformation of the received information, it only provides functions to configure graphic presentations and shows information for the operator on the display.

There are many kinds of presentations according to different application domain. For example, the iconography used to describe an objective on the display is different for military purposes radars or for aerial navigation radar, civil or military.

3. Signals Acquisition

The acquisition component was designed to provide functions of signal acquisition, data conversion and synchronization of processes to read information.

It is composed of the control drivers of the hardware devices involved and software for reading data. It interacts with the different components obtaining relevant information and carries out all needed transformation of data required by the system. Therefore, analogic signals are digitalized using analog/digital converter boards, both if they are connected to the PC or to the FPGA. This component also implements all conversion of data necessary for the processing component to receive an standardized input set of information, according to a previously defined format.

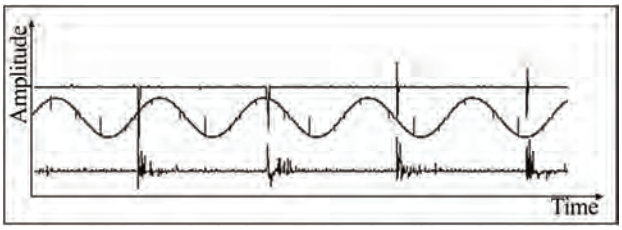


Figure 3. Signal L1 (upper), Signal H1 (middle) and Signal P1 (lower)

The following is an example of the characterization of some signals studied on the radar commercialized as Cyrano (Figure 3):

General Synchro signal: (L1). Voltage range: +/- 40 Volts. Chain of pulses with 500 ms of separation between pulses and width of pulse of 2 ms. Each pulse indicates the beginning of a phase of issue of the radar antenna. The system produces 2000 pulses a second.

Video indicator signal: (P1). Voltage range: +/- 5 Volts. This signal is the output of the stage of intermediate frequency that forms the signal coming from the radar antenna. As the radar produces 2000 pulses a second, a signal of 1500 cycles for each pulse of the antenna is obtained. Frequency: 3 Mhz. Noise level: When not transmitting it is 2 Volts. When transmitting it is +/- 200 mV.

Antenna Y scanning signal: (H1). Voltage range: +/- 37 Volts. This signal indicates the horizontal shift of the radar antenna. Values measured in radar of 37 V and 530 Hz, with 180º for +60º and 0º for –60º antenna position.

In Figure 3, a section of exploration of 4 cycles is observed. On signal L1, the pulses of synchronization determine the beginning of the cycle. As for the signal H1, it can be seen that for each cycle of L1, there is approximately a complete cycle of H1, which indicates the horizontal position of the radar antenna at the beginning and at the end of each cycle. Finally, P1 describes the video information received by the radar in each cycle showing some nearby contacts and many contacts of little intensity (noise) along the rest of the section.

4. Signals Processing

4.1 General Description

The Analyzing Component was designed to analyze the different input signals in real time, calculating the values of the "elements" or "objectives" of the presentations and sending this information to the Visualizing Component

To compute the data of some objectives it is necessary to analyze the information coming from several signals. The information of some signals must be used in the calculation of several objectives, so any change in the input signals implies recalculate the data of all the objectives affected by that change. This mechanism is implemented through a two dimensional matrix containing the information of dependence among objectives to be charted and input signals. This "Matrix of Dependencies" contains in each cell a pointer to a function; if there is a dependence between signal and objective a function is referenced, which calculates the information corresponding to the objective and sends it to the Visualizing Component, otherwise, the reference is nil.

The input data received from the Acquisition Component and the output data sent to the Visualizing Component are transmitted by structures of communication called «SignalPipe», implemented by sockets of TCP protocol. These structures were implemented to provide a standard interface for the communication among the components.

Before being processed, the signals are filtered in order to reduce the white noise coming from conversion. To do so, a Kalman filter is used [8]. This filter reduces the chance of making mistakes while recognizing patterns.

The time update equations are defined as:

$$\hat{x}_k^- = \hat{x}_{k-1}$$

$$P_k^- = P_{k-1} + Q$$

and the measurement update equations are:

$$K_k = \frac{P_k^-}{P_k^- + R}$$

$$x_k = x_k^- + K_k(z_k + x_k^-)$$

$$P_k = (1 - K_k)P_k^-$$

Thus, the signals are filtered reducing significantly their noise. (Figure 4)

4.2 Component Implementation

The Analyzing Component in charge of the processing of signals (Figure 5) is composed of:

A structure of information storage refered to the specified set of input signals.

A set of calculus algorithms of all the attributes and objectives that plot the defined presentations.

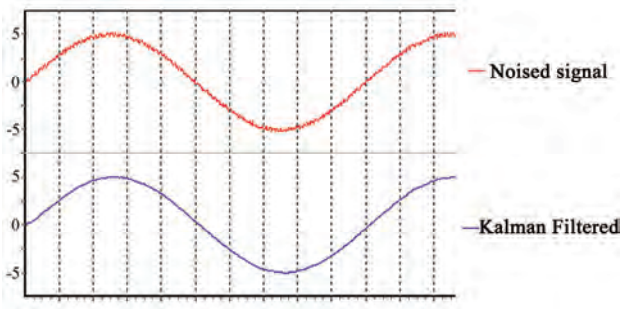


Figure 4. White noise reduction by Kalman filter

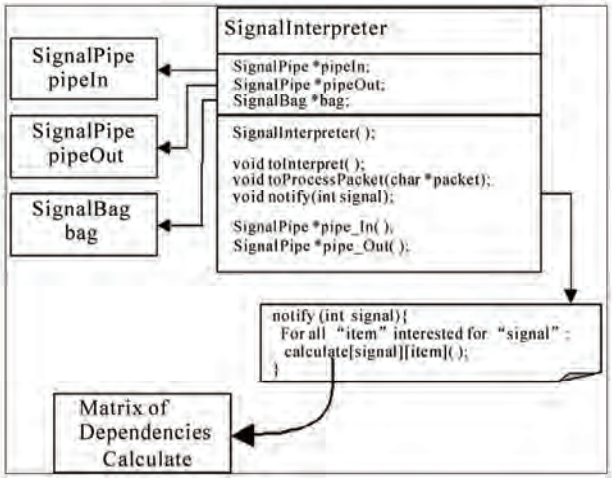


Figure 5. Implementation diagram of the Analyzing Component

A structure of dependences between input signals and output objectives.

References to communication structures with the Acquisition Component and the Visualizing Component.

The structure of signal storage implemented in the class «SignalBag» keeps all the necessary information for different signals during runtime. Every time a change of a signal is received the structure is updated and the affected objectives are notified. Internally, this structure has a buffer of storage for each signal. These buffers are of variable length (according to the characteristics of each signal) and can be updated and easily examined both to incorporate the new data and to use the stored data to calculate the visualization information.

The notification process invocates the calculus routines of the affected objectives and the sending of the updated data to the Visualizing Component. To carry out this task, the component has a structure to establish which objectives depend on which signals. This structure "Matrix of Dependencies" and contains references to the processing algorithms for different objectives. These algorithms are run by the «calculate» method.

The references to the structures of communication with the Acquisition and Visualizing Components are called «pipeIn» and «pipeOut».

The class «SignalInterpreter» is the main class of the component and implements the mechanism of signal processing. It is realized by the method «toInterpret» (Figure 6). The method «toInterpret» implements the main function of the component, which receives from the input pipe «pipeIn» the changes in the signals. For each packet received the method «toProcessPacket» is invocated, it updates the values of the signals in the "bag" structure and invocates the functions of calculus of the objectives affected by the change, which notify their results to the output pipe «pipeOut».

4.3 Data Sending and Receiving

The sending and receiving of information with the Acquisition and Visualizing Components is carried out according to the conventions defined by the structure of communication «SignalPipe», where the messages are textual and have a format of 3-ary ELEMENT-ATTRIBUTE-VALUE. When the messages to be sent correspond to 2-ary the third field is completed with the value zero ("0").

The messages received by the component use the format of two elements, where the field ELEMENT corresponds to an identifier of the input signal and the field VALUE corresponds to the numerical value. Messages sent to the Visualizing Component use the format of three elements where the field ELEMENT corresponds to an identifier of the objective in the presentation, the field ATTRIBUTE corresponds to a characteristic of the objective and the field VALUE corresponds to the numerical value.

5. Pattern Recognition

Pattern automatic identification is achieved through the use of classification techniques. Neural networks are one of the approaches to be used.

In the context of neural networks, classification involves to get a function responsible for grouping the data into categories, based on some characteristics. This function is materialized by a neuro-classifier which is trained using different classes of input data together with their categories. A classifier neural network associates an input vector with a represented category producing as output a signal whose value indicates the membership of the

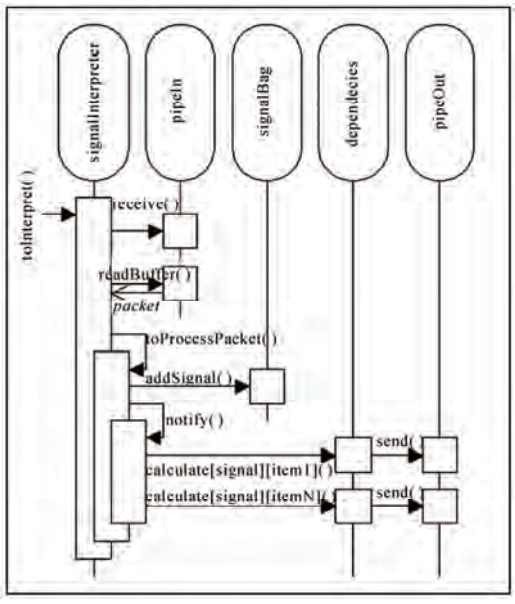


Figure 6. Interaction diagram of method «toInterpret»

input to the category. The network can also indicate the degree of acceptance of the input to that category; therefore, the output values are not restricted to binary signals.

The RBF networks are generally used as neuro-classifiers and they are well adapted to target identification in radar images [9]. This kind of networks has an input layer composed of branches nodes and a hidden layer for which each node has a special activation function. This function is placed into the central vector of the cluster within the feature space; thus the function has a significant response for those vectors near its center. Each result from the hidden layer has a weight value associated. The output layer is responsible for summing up the products got from the results of the hidden layer, together with their weight values. (Figure 7).

The system presented in this paper has a trained neural network able to classify targets. This net is implemented on a Xilinx's Virtex-4 (xc4vsx25-11-ff668) FPGA [10]. The main advantage of implementation using this platform is the natural parallelism among the networks that is allowed.

The neural network gets the values of <<SignalBag>> taking them as if they were an image. This "image", in a monochromatic format with 12 bits per pixel resolution, is composed by 1500 rows (input data at 3 Mhz) and 2000 columns corresponding to the whole scan of the antenna (±60º).

A profiling over the video signal (P1) is performed for both axes, to determine the presence of objects. In Figure 8 the result got for a radar image where a B-52 is present is shown.

In the case presented there are two regions to be analyzed: R1 and R2. The first of them corresponds to the location of a target, whereas the second one corresponds to the earth echo.

Region of interest is defined as the sub region that can contain a pattern. For this implementation, the size of the region is fixed (150×150 pixels). The center of each region detected during the profiling will also be the center of the region to analyze.

In this case, both regions (Figure 9) are evaluated since the implementation of the classifier allows effective evaluation at a reasonable rate.

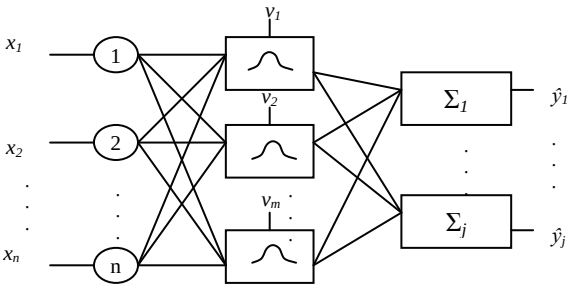


Figure 7. RBF Neural Network

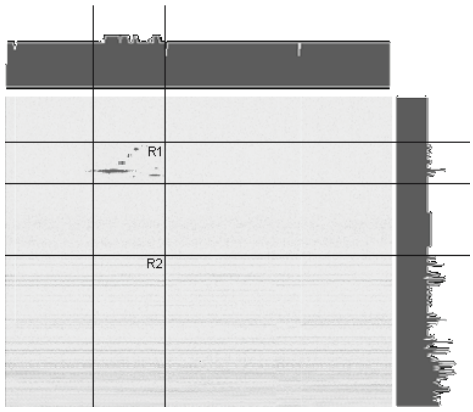


Figure 8. Section profile of a radar image

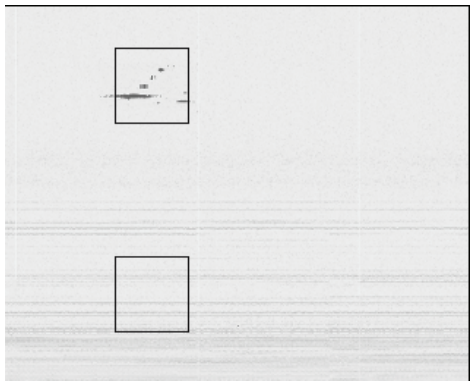


Figure 9. Regions of Interest

Each one of these regions is projected in order to take the most relevant features of the region. Currently the composite profile is being used as characteristic vector since it allows distinction of shapes. However, different techniques for extracting characteristics are being analyzed; these techniques allow more efficient differentiation of patterns (Specially Wavelets). These feature vectors are evaluated by the neural network to classify them.

The hardware neural network architecture does not have on-chip learning; therefore its knowledge is preset. The number of neurons that conforms it depends on the amount and quality of the patterns that allows classification. In this case, the net was trained using 30 patterns corresponding to M-3 class and 30 patterns corresponding to B-52, having as a result a neural network of 25 neurons, with an operational frequency of up to 1000 patterns analyzed per second.

6. Information Visualization

6.1 General Description

The MFD has, at least, a "view" or "presentation". A view is a set of graphical or textual information presented to the user in a predetermined format.

The Visualizing Component was designed to support

several parallel views with different formats and conventions that can be modified or replaced in a flexible manner. Two elements were defined for this purpose: «GraphicalItem» «Presentation».

A «GraphicalItem» is any graphical element or textual information that must be presented in the MFD; for example, a contact, the scale rule, information of the scanning of the radar.

A «Presentation» is basically a set of graphicable items. A presentation can be a sinusoidal graph to describe the values of certain signal of the radar, a diagram of the zone explored by the radar, or else a list of numerical data for the operator. Those examples can be different presentations for a same situation. Each kind of presentation is more or less useful according to needs. The present design makes it possible to develop each kind of presentation in an easy and efficient way, because a new presentation only takes three steps: Write an specification of the view in a file, write the code for the graphicable item and set a presentation that includes the specifications and items previously defined.

6.2 Component Implementation

The Visualizing Component was developed to handle efficiently graphical and textual information given by «GraphicalItem» and organized as different kinds of screens called «Presentation» whose format is configurable. The visual parameters that represent attributes (color, size, position) of the «GraphicalItem» must be updated in real time.

These functionalities are implemented as follows:

The class «RadarView» is in charge of all functions of the presentation.

The class «GraphicalItem» is in charge of all functions of the items to be displayed.

The formats of the presentations are specified in configuration files to be able to change the presentation with no need to modify or recompile the system.

The changes in the information to show on the display are received through the structure of communication «SignalPipe» with the component of processing «Analyzer».

The class «RadarView» (Figure 10) sets the visualizing format of the view obtaining the information from a previously defined configuration file. The aim of the configuration file is to define the formats of the representation system and the iconography to be used.

In runtime the configuration of the presentation is stored in an inner 2-dimensional structure called «Configuration» which also stores information of the modifications of the presentation when there are changes in the system (change in the mode of operation of the radar, or detection of a specific objective). It is also possible to alter the format of representation in runtime by only updating the matrix from another configuration file.

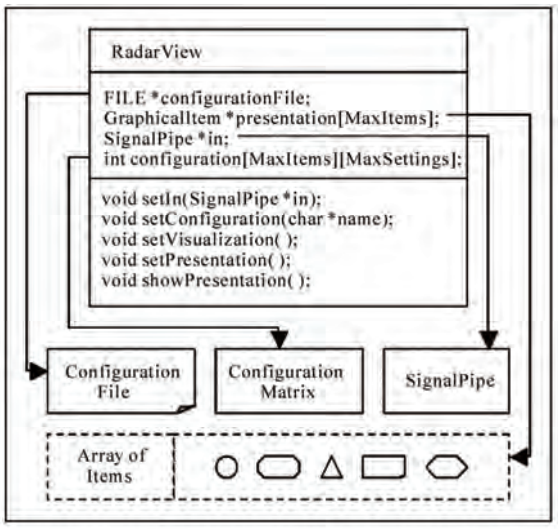


Figure 10. Implementation diagram of the class «RadarView»

The graphical items are stored in an array called «Presentation» (Figure 6) defined from the configuration matrix, and as the data of the updating of the different graphical items are modified to provide an inner representation of the view to be presented in the MFD (changes in the location of a contact or in the mode of operation of the radar).

6.3 View Implementation

A presentation is composed of a set of objective elements called “Graphical Items”. Each item is represented as an instance of the abstract class «GraphicalItem». As many of the elements to be presented on the MFD are textual information, an abstract class which defines the specific behavior of a textual graphicable item called «GText».

The class «GraphicalItem» defines the behavior and properties characterizing an element of the presentation. To model a presentation it is necessary to design a class for each type of item that will compose it and these must be sub classes of «GraphicalItem». Thus, each sub class implements the behavior defined by the super class and a unified mechanism of treatment of the different graphicable elements is obtained.

Each graphicable item is defined by a set of attributes common to all of them (name, position in the axis X, axis Y, colour) whose values are initialized from the format of representation set in the configuration matrix and they are updated in real time from the information of changes coming from the interpretation component.

6.4 View Configuration

The aim of configuration mechanism through files is to offer flexibility to handle the presentation formats. This mechanism allows the system to receive all the necessary

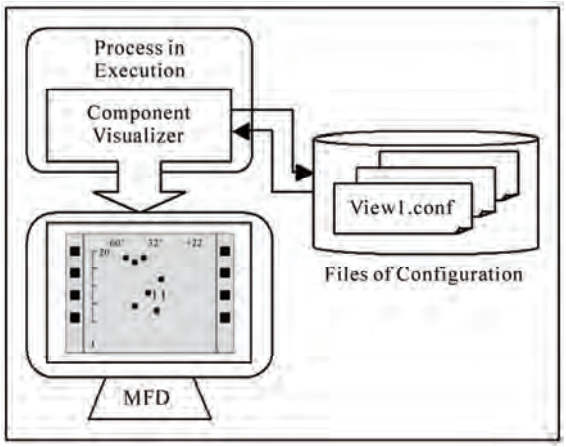


Figure 11. Diagram of interaction between files of configuration and the system

information to display a presentation on the screen from a file independent from the system (Figure 11). Thus, it is possible to modify the system through the configuration files with no need to recompile the code.

For each representation it is necessary to code an associated configuration file that includes a clear definition of the graphicable item including a predetermined presentation and the values corresponding to its inner attributes.

These items of the presentations enable to define views according to specific needs of each application domain. For example, an item to be defined in each presentation is its background which can be a color, a map, or a satellital photograph both for a air traffic radar and a meteorological radar. The system also provides the ability to present on screen textual information in different typographies and sizes, elemental graphs and images in different formats to identify a certain objective by means of a graphic representation.

7. Fine Tuning For Real Radars

As a case study a system previously described was implemented. The software system has been completely programmed in ANSI/C, and because of the characteristics for handling processes and real time services, it has been implemented on Linux operating system. The circuits and drivers have been completely specified by using VHDL on Xilinx platform.

7.1 Signals Acquisition

The technological platform used for the development was mainly based on a PC orienting the design to the definite implementation on a PC/104. As some recognition algorithms can not be implemented in software because of the performance requirements, coprocessors to implement these functions on FPGA were developed.

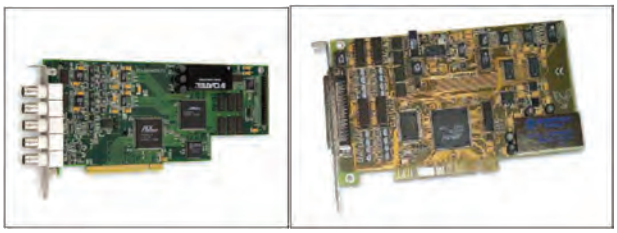
Additional boards with drivers for two devices of analogical and digital signal acquisition were added to the PC. The former is a PCI board made by Measurement Computing, model PCI-DAS-4020/12 (Figure 12(a)). The latter is a PCI board made by Eagle Technology; model PCI-703S (Figure 12(b)). The PCI-DAS-4020/12 has 4 channels for analogical input, 2 channels for analogical output and 24 channels for digital I/O, with a resolution of 12 bit samples at a frequency of up to 80 Mhz. The PCI-703S has 8 channels for analogical input, 2 channels for analogical output and 8 channels for digital I/O, with a resolution of 14 bits samples at a frequency of up to 400 Khz.

The complex signal acquisition is made through FPGAs which realize a preprocessing to filter the analogical pulses and to recognize useful signal patterns. The FPGAs used are: Virtex-4 XC4VLX25 (Figure 13(a)) of Xilinx [11] of 24192 equivalent cells, 168 KB of RAM, 448 I/O pins and 48 DSPs which enables the operation as an analogical/digital conversor of signals of up to 105 Mhz with 14 bits of resolution per sampling [12]. The other FPGA is the Virtex-2 PRO XC2VP30 of Xilinx of 30816 equivalent cells, 428 KB of RAM and 644 I/O pins (Figure 13(b)).

7.2 Signal Processing

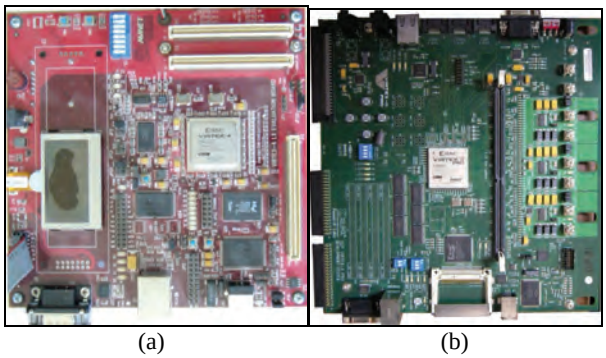
The algorithms of signal interpretation are realized on the two working platforms, both on the PC and by electronic circuits of specific application (FPGA) operating as preprocessors.

All the operations previous to the visualization, noise filtering, identification of possible objectives and calculus



(a) (b)

Figure 12. PCI-DAS-4020/12 (a) and PCI-703S (b)



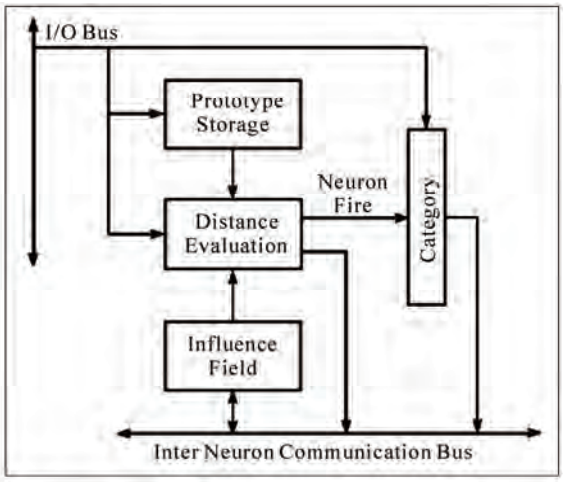
(a) (b)

Figure 13. Left Virtex-4 XC4VLX25 (a) and Right Virtex-2 PRO XC2VP30 (b)

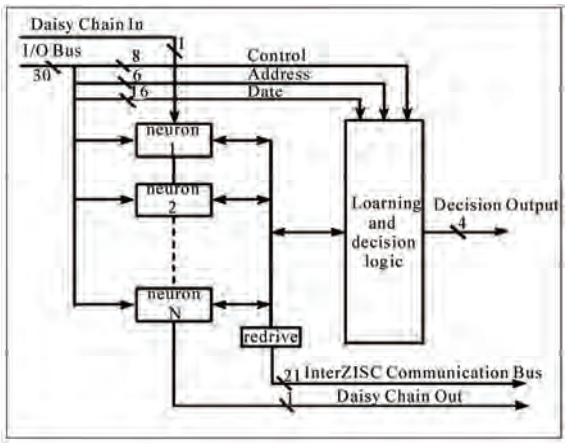
of distances from the identified objectives are realized on the PC.

Owing to the high frequency of sampling of some signals and the high cost of processing in real time, it is necessary the inclusion of dedicated hardware for the processing of signals such as video, synchronisms, telemetry marks (tracking) and others. These signals provide the system with detailed information of targets, times and distances with precision of the order of microseconds. Such detailed analysis can not be made through software algorithms on the PC owing to the real time requirements.

The analysis and interpretation by hardware is materialized on devices of programmable logic or FPGAs that enable the synthesis of dedicated hardware circuits programmed in VHDL. The designed component behaves as a dedicated coprocessor of signals.



(a) Neuron structure



(b) Interconnection among neurons

Figure 14. Neuron structure and interconnection among neurons

This subsystem provides functions such as target recognition on a radar echo, search for determined targets in zones of radar echo (in a determined range of distance) and position of the telemetry mark. This function is implemented on dedicated hardware because it is a short-amplitude high-frequency pulse whose detection through software is complex and expensive.

Target recognition is implemented from a simplified and modified version of a Radial Basis Function Neural Networks (RBFNN) based on the neuronal processor ZISC78 [4–6,13–15] of IBM (Figure 14) which detects similarities between the information from the radar echoes and a variable set of training patterns representing the profiles of different objectives on the radar.

In this neuronal architecture each pattern representing a wave profile of an object is stored as a prototype in each neuron. Evaluating all the neurons in parellel the one having the prototype with a certain margin of similarity reacts sending a code to the controller of the net representing the belonging of the pattern to a determined category. Representing with characteristic prototypes the different objects to be recognized it is possible to detect them in each radar echo.

7.3 Data Visualization

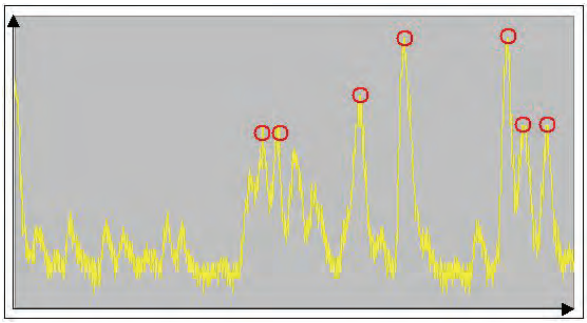
Different presentations have been programmed and put into practice. Most of them are variations of the presentations type B which locate on the display different identified objectives on a cartesian representation according to the information of distance (axis Y) and azimuth (axis X) (Figure 15(a)). Presentations type A were also implemented, these show on the display a cartesian representation according to the information of amplitude of the signals received through the antenna (axis Y) and distance (axis X) [1] (Figure 15(b)).

8. Acknowledgments

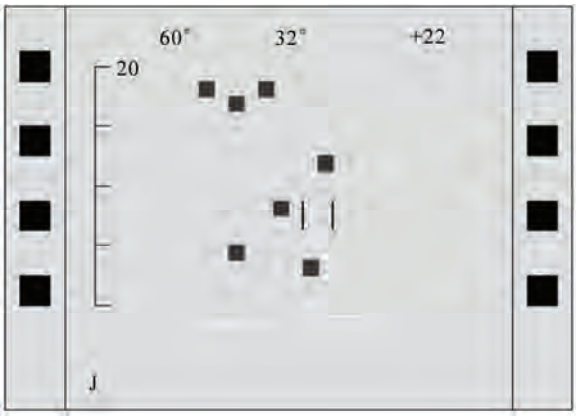
We thank F. Oyarbide, F. Cutropía, L. Burgos, R. Cela, and C. Mayol, Redimec SRL staff. We are also grateful for the financial collaboration of Redimec SRL, National Agency of Scientific and Technological Promotion through the ANR 111/03 granted to Redimec SRL for the development of part of this project, Commission of Scientific Investigations of Buenos Aires Province for the scholarship to Martín F.Mezzanotte and all the technical staff of the institutions that have allowed us to work with their equipment.

9. Conclusions

Based on the tests made it is possible to conclude that the system developed can be successfully applied to different systems of existing radars to implement changes, updatings or improving with different adaptation grades and costs. This software was used on a Panel PC (Figure 16). [16].



(a): Presentation "A"



(b): Presentation "B"

Figure 15. Two kinds of radar displays

Figure 16. System tested on Panel PC with a LCD 12.1" TFT

The proposed architecture proved satisfactory to guarantee the fullfillment of functional and nonfunctional requirements. Anyway it is critical to apply this system on different technological platforms to test the functionalities in different application domains.

In a short time it is important to implement different signal processing algorithms. First, automatic recognition and tracking of objectives and/or paths algorithms must be developed.

The development of new presentations is also important since allows to adapt the present system to new and different application domains.

REFERENCES

[1] S. A. Hovanessian, "Radar detection and tracking systems," Artech House, Inc., 1973.

[2] L. Sha, *et al.*, "Evolving dependable real-time systems," Technical Report CMS/SEI95 -TR-005, CMU, Software Engineering Institute, 1995.

[3] J. R. Y. Johnson and R. W. Johnson, "Challenges of computing the fast fourier transform," Optimized Portable Application Libraries (OPAL) Workshop, Kansas City, Junio de 1997.

[4] G. Noone, "Radar pulse train parameter estimation and tracking using neural networks, in proc," IEEE ANNES'95, IEEE, November 1995.

[5] E. Mingolla, W. Ross, and S. Grossberg, "A neural network for enhancing boundaries and surfaces in synthetic aperture radar images," Neural Networks 1999; 12 499—511.

[6] R. C. Krishnamohan and P. S. Mmoharir, "Radar signal design problem with neural network processing," Sadhana, Vol. 26, Part 3, pp. 237–241, June 2001.

[7] R Perry, *et al.*, "Trellis structure approach to multitarget tracking. adaptive sensor array processing workshop," Marzo de 1999, MIT Lincoln Laboratory.

[8] G. Welch and G Bishop, "An introduction to the Kalman filter," Technical Report: TR95-041, University of North Carolina, 1995.

[9] Q. Zhao and Z. Bao, "Radar target recognition using a radial basis function," Neural Networks, Vol. 9, No. 4, pp. 709–720, Elsevier, 1996.

[10] Xilinx, Inc., Virtex-4 User Guide V2.2, www.xilinx.com, 2007.

[11] Xilinx, Ltd., Virtex-4 Family Overview, DS112 (V2.0), January 23, 2007.

[12] K. Chapman, Pico Blaze—Amplifier and A/D converter control for Spartan-3E Starter Kit, Xilinx Ltd, February 23, 2006.

[13] Solanki Gautam V., Neural network and its application in pattern recognition, Seminar Report, Indian Institute of Technology, 2004.

[14] Y. H, Liao, Neural networks in hardware: a survey, Department of Computer Science, University of California, 2001.

[15] Silicon Recognition, 1150 Industrial Avenue, Suite C, Petaluma, CA 94952, URL: www.silirec.com.

[16] Acosta *et al.*, "Desarrollo de un Visualizador de Señales de Radar," CACIC, October 2006, San Luis, Argentina.

J. Software Engineering & Applications, 2010, 3: 160-166
doi:10.4236/jsea.2010.32020 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Development of a Web-Based Decision Support System for Cell Formation Problems Considering Alternative Process Routings and Machine Sequences

Chin-Chih Chang

Department of Information Management, Jen-Teh Junior College of Medicine, Nursing and Management, Taiwan, China.
Email: chinju.chang@gmail.com

Received November 16th, 2009; revised December 3rd, 2009; accepted December 12th, 2009

ABSTRACT

In this study, we use the respective advantages of the tabu search (TS) and the Web-based technologies to develop a Web-based decision support system (DSS) for cell formation (CF) problems considering alternative process routings and machine sequences simultaneously. With the assistance of our developed Web-based system, the CF practitioners in the production departments can interact with the systems without knowing the details of algorithms and can get the best machine cells and part families with minimize the total intercellular movement wherever and whenever they may need it. To further verify the feasibility and effectiveness of the system developed, an example taken from the literature is adopted for illustrational purpose. Moreover, a set of test problems with various sizes drawn from the literature is used to test the performance of the proposed system. Corresponding results are compared to several well-known algorithms previously published. The results indicate that the proposed system improves the best results found in the literature for 67% of the test problems. These show that the proposed system should thus be useful to both practitioners and researchers.

Keywords*: *Web-Based, Cell Formation, Tabu Search, Decision Support System, Alternative Process Routings

1. Introduction

In response to various and diversified customer demands, companies must adopt innovative manufacturing strategies and manufacturing technologies to achieve an efficient and flexible manufacturing system. Group technology (GT) is one such approach that meets the requirements of system flexibility and product variations. GT is a manufacturing philosophy, which determines and divides the components into families and the machines into cells by taking advantage of part similarity in processing and design functions. Studies show that 30%–75% of the product cost is due to materials handling [1]. Cellular manufacturing (CM) is the application of group technology (GT) in manufacturing systems. The implementation of cellular manufacturing system (CMS) design has been reported to result in significant benefits such as reductions in material handling costs, work-in-progress inventory, throughput times and set-up times, simplified scheduling and improved quality [2]. Cell formation (CF) is the crucial element in designing CMS. However, it has been known that the CF in CMS is one of the NP-hard combinational problems [3], as it becomes difficult to obtain optimal solutions in an acceptable amount of time, especially for large-sized problems. While considering alternative process routings and machine sequences to CF problems, making the problems more realistic; however, it leads to a more complex problem than the simple CF problem. Thus, development of an effective computer-aided manufacturing CF support system is necessary. In this regard, many models and solution approaches have been developed to identify machine cells and part families. These approaches can be classified into three main categories [4]: 1) mathematical programming (MP) models [5–8], 2) heuristic/meta-heuristic solution algorithms [9–11], and 3) similarity coefficient methods (SCM) [12,13].

Due to their excellent performance in solving combinatorial optimization problems, meta-heuristic algorithms such as genetic algorithm (GA), simulated annealing (SA) and tabu search (TS) have been the most successful solution approach to efficiently solve the CF problem and its variants with good results. Among the meta-heuristic algorithms, TS has been successfully used to solve many problems appeared in manufacturing system including cell formation problems [14]. Hence, we adopt it as a solver to solve the CF problem considering alternative

process routings and machine sequences simultaneously in the development of our CF Web-based decision support system (DSS).

In addition, CF algorithms are usually expressed in mathematical terms, which may only be understood by domain experts, but not by most of the CF practitioners in the production departments. With the assistance of CF DSS, users can interact with the systems without knowing the details of algorithms. Moreover, due to an increasing global competition, companies are now shifting to a geographically distributed manufacturing environment. Besides, the information flow nowadays requires reliability, efficiency and security. With the emergence of information technology, the traditional way of communication of information flows between companies and between internal parties can now be replaced by the interconnected network [15]. The Internet provides an open environment for companies to connect with their business partners as well as to serve as a medium for internal information flows [16]. Moreover, manufacturing systems have migrated to integrate with the Internet to provide a remote access and control system with the characteristics of quick response and real line monitoring [17].

In this study, we use the respective advantages of the TS and the Web-based technologies to develop a Web-based DSS for CF problem considering alternative process routings and machine sequences simultaneously. An example taken from the literature is adopted for illustrational purpose. To further verify the feasibility and effectiveness of the system developed, ten test problems with various sizes drawn from the literature are used to test the performance of our proposed CF solver. Corresponding results are compared to several well-known algorithms previously published.

The remainder of this article is organized as follows: Section 2 describes the problem definition including cell formation and the mathematical model; Section 3 details the implementation of our Web-based CF DSS; Section 4 verifies the performance of the proposed system and methodology; and Section 5 concludes the paper.

2. Problem Definition

2.1 Cell Formation

In a simple CF problem, cell formation in a given 0–1 machine-part incidence matrix involves rearrangement of rows and columns of the matrix to create part families and machines cells, in which the cellular movement can be minimized and the utilization of the machines within a cell can be maximized. Two matrices shown in Figure 1 are used to illustrate the concept. Figure 1(a) is an initial matrix where no blocks can be observed directly. After rearrangement of rows and columns, two blocks can be obtained along the diagonal of the solution matrix in Figure 1(b). For those 1's outside the diagonal blocks, they are called "exceptional elements"; while those 0's inside the diagonal blocks are called "voids".

When parts have alternative process routings (APR) is called the generalized CF problem. Such as the case shown in Table 1, part #1 has two process routings R1 and R2. Kusiak [5] first described the problem and presented an integer-programming model to solve the problem. While introducing APR to CF problems, the grouping of parts can be more effective due to the flexibility of the routes; however, it leads to a more complex problem than the simple CF problem. Under this circumstance, not only the formation of part families and machine cells must be determined but also the selection of routings for each part has to be determined to achieve decision objectives such as the minimization of intercellular movement. For instance, Table 2 provides a feasible solution to the sample problem of Table 1 with a total intercellular movement of 215.

2.2 Mathematical Model

The decision objective of their research is to minimize the sum of total intercellular movement. The 0–1 integer programming model that they formulated is given below, and the notations are introduced first.

(a)

Machines \ Parts	P1	P2	P3	P4
M1	1	0	0	1
M2	0	1	1	0
M3	1	0	0	0
M4	0	1	1	0

(b)

Machines \ Parts	P2	P3	P1	P4
M2	1	1	0	0
M4	1	1	0	0
M1	0	0	1	1
M3	0	0	1	0

Figure 1. Rearrangement of rows and columns of matrix to create cells: (a) initial matrix and (b) matrix after rearrangement

Table 1. Initial machine-part matrix where alternative process routings are allowed

PV	75		130		110		145		110		105		140		115	
PN	P1		P2		P3		P4		P5		P6		P7		P8	
RN	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2	R1	R2
M1	*1		1		1	1	1	1		1		1	1		1	
M2		1		1					1		1			1		1
M3					2	2	2	2								
M4	2		2			3	3	3		2			2			
M5	3		3	2	3			4					3			
M6		2												2	2	2
M7									2	3	2	2				
M8		3	4		4	4	4	5			3	3				
M9	4			3					3	4			4	3		

PV: Production Volume; PN: Part Number; RN: Routing Number; * Process Sequence

	P5 R1	P6 R1	P8 R2	P1 R1	P2 R1	P3 R1	P4 R1	P7 R1
M2	1	1	1					
M6			2					
M7	2	2						
M1				1	1	1	1	1
M3						2	2	
M4				2	2		3	2
M5				3	3	3		3
M8		3			4	4	4	
M9	3			4				4

Figure 2. Final machine-part matrix of Table 1

Notations:

m	Number of machines
p	Number of parts
NC	Number of cells
V_i	Production volume for part i
Q_i	Number of routings for part i
U_m	Maximum number of machines in each cell
L_m	Minimum number of machines in each cell
K_{ij}	Number of operations in routing j of part i; the operations of part i along route j are processed on a machines' set of $\left\{u_{ij}^{(1)}, u_{ij}^{(2)}, ..., u_{ij}^{(k)}, u_{ij}^{(k+1)}, ..., u_{ij}^{(K_{ij}-1)}, u_{ij}^{(K_{ij})}\right\}$
$u_{ij}^{(k)}$	Machine index for the k-th operation of part i along route j
Y_{kl}	1, if machine k locates in cell l; 0, otherwise
Z_{ij}	1, if routing j of part i selected; 0, otherwise

The 0–1 integer programming model is as follows:

$$\text{Min} \quad ICM = \sum_{i=1}^{p}\sum_{j=1}^{Q_i}\sum_{k=1}^{K_{ij}-1}\sum_{l=1}^{NC} V_i Z_{ijl} Y_{(u_{ij}^{(k)})l}\left(1 - Y_{(u_{ij}^{(k+1)})l}\right) \quad (1)$$

st:

$$\sum_{j=1}^{Q_i} Z_{ij} = 1 \qquad \forall i \in \{1, 2, ..., p\} \quad (2)$$

$$\sum_{l=1}^{NC} Y_{kl} = 1 \qquad \forall k \in \{1, 2, ..., m\} \quad (3)$$

$$L_m \le \sum_{k=1}^{m} Y_{kl} \le U_m \quad \forall l \in \{1, 2, ..., NC\} \quad (4)$$

$$Y_{kl}, Z_{ij}, \in \{0,1\} \quad \forall i, j, k, l \quad (5)$$

In the above model, Equations (1) is the objective function, which show the calculation of the total inter-cellular movement. Equation (2) indicates that only one process routing will be assigned to each part. Equation (3) provides a restriction that each machine will be assigned to exactly one cell. Equation (4) assigns the upper and lower limits of the cell size. Equation (5) indicates that Y_{kl} and Z_{ij} are 0–1 binary decision variables.

The large number of binary variables and constraints makes it difficult to obtain optimal solutions in an acceptable amount of time. Developing an effective computer-aided manufacturing CF support system is more appropriate than using the exact method in terms of solution efficiency, especially for large-sized problems. This paper, thus, presents an efficient Web-based DSS for CF problem. The developed system is described and explained in detail in the next section.

3. System Development

In this study, we dedicated to develop a Web-based CF DSS. With this system, the users can upload the CF data to Web serve and then with the CF solver executed, they can get the best machine cells and part families with maximize grouping efficacy wherever and whenever they may need it. The system architecture for CF DSS is shown in Figure 3. From the figure, we can know that the system consists of five elements. They are the clients (i.e., users), the Web-based user interface, the Web server, the CF solver and the database. All of them are linked up with the Internet but may be located in different geographical places. We will describe them below.

3.1 Client and Web-based User Interface

Web browsers are clients that connect to Web servers and retrieve Web pages for display. Using appropriate Web browsers, such as Netscape's Navigator or Microsoft's Internet Explorer, users can input data or view the CF results through a dynamic hypertext user interface. Because of the PHP is a widely-used general-purpose scripting language that is especially suited for Web development and can be embedded into HTML. Hence, we use PHP to making dynamic and interactive Web pages for the Web-based user interface which consists of three buttons on the top of the screen. The framework of the user interface is shown in Figure 4 which is simple and Framework of Web-based user interface

Upload Data: The client users can upload production data to Web serve by using this button. The production

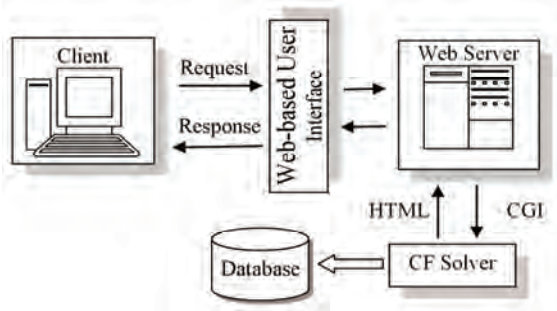


Figure 3. System architecture for CF DSS

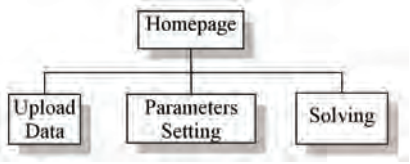


Figure 4. Considered to be user-friendly, we will describe them below

data are given through a text document readable with any text viewer (*.txt).

Parameters Setting: The client users can set up the parameters for CF solver.

Solving: With this button be pressed, the CF solver will be executed to minimize the total intercellular movement.

3.2 Web Server

The Web server is a computer that serves requested Web pages. The Web server interacts with the individual user's Web browser and accepts external Hypertext Transfer Protocol (HTTP) requests from the browser. An Application Programmer's Interface (API) is distributed, along with most of the commercially available browsers, such as Netscape's Navigator or Microsoft's Internet Explorer. Application programs, such as PHP, ASP and JSP, can be written using these APIs to enhance the capabilities of a browser. Because of the Apache HTTP Server has been the most popular Web server on the Internet since April 1996. Therefore, we use the Apache server as Web server in this study.

3.3 CF Solver

The CF solver was developed using Visual C++ programming languages. It consists of two stages: the initial solution construction and the improvements. The Single Linkage Clustering Algorithm (SLCA) of McAuley [13] is adapted in the first stage to produce good initial solutions, while the TS continuously improves and generates more effective solutions through the TS algorithm in the second stage. The proposed generic framework for the CF solver is shown in Figure 5 which is actually consists of the following seven steps:

1) Initialization of computational parameters;
2) Construction of initial solution;
3) Searching of improving neighborhood solutions;
4) Update of tabu list;
5) Update of better solutions found;
6) Check of timing for directing searching toward diversified solution space by applying mutation operator;
7) Check of solution stagnancy.

Note that the first five steps are the same as the TS algorithm, while Step 6 generates new solutions with higher degree of diversification in order to increase the probability of finding the global optima, and Step 7 avoids spending too much computational efforts in order to have a balance between the computational effectiveness and efficiency.

For the CF solver, the insertion strategy is applied to produce a new neighborhood solution and the values of parameters are given below: tabu list size is equal to 7, a limit of iterations for each run is set to 1000 and the mutation probability for each gene is set at 0.8.

3.4 Database

A database server machine may be physically different from the Web server that maintains the database. Due to the MySQL is the world's most popular open source database. Hence, we use MySQL server as the database-server in this study. This remote database is accessed

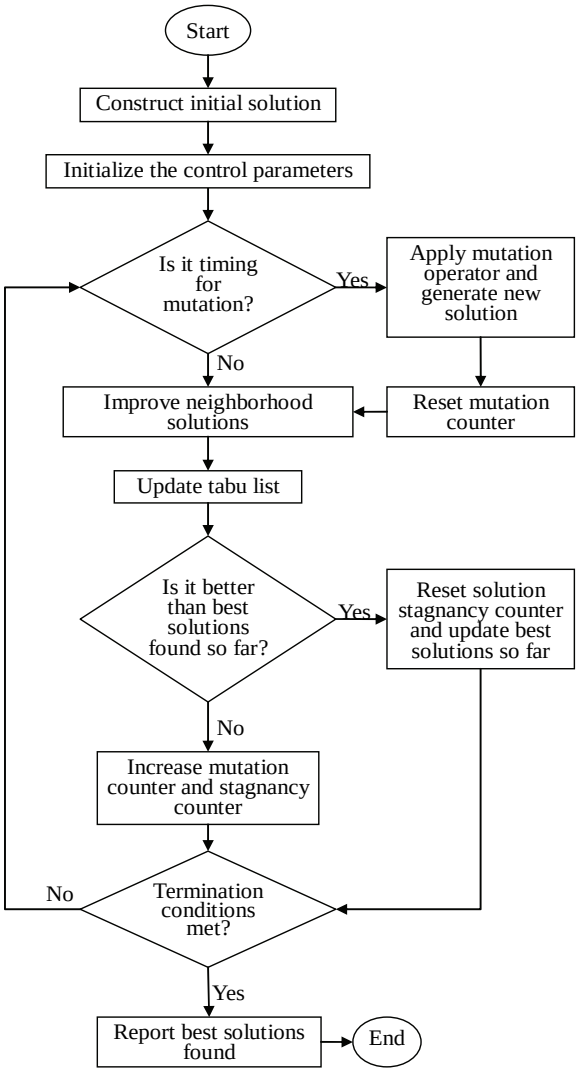


Figure 5. Flowchart of CF solver

through the Open Database Connectivity (ODBC) gateway to insert, delete or update information in the database.

5. Research Results

The research results consist of two parts. They are the numerical illustration and the comparative study. We will describe them below.

5.1 Numerical Illustration

We applied a numerical example, which was drawn from Sofianopoulou [9], to test the performance and usability of our developed system. The step-by-step procedures are described as follows:

Step 1: Press the "Upload Data" button to upload the production data to Web server, as shown in Table 2, which consists of 4 machines and 5 parts.

Step 2: Set the parameters and constrains for CF solver by pressing the "Parameters Setting" button (See Figure 6).

Step 3: Press the "Solving" button to execute the CF solver to groups the machines into machine cells and parts into part families with minimize the total intercellular movement. As shown in Figure 7, the CF solver only took 0.015 seconds to get the final configuration for the cell formation with two cells and without any inter-cell movement.

Table 2. Production data for the numerical example

Part No.	Production volume	Routing number	Process sequence
1	50	1	4 3
		2	2 4
		3	2 1
2	30	1	2 3
		2	3 1
3	20	1	4 1
		2	4 2
4	30	1	1 4
		2	1 3
5	20	1	3 4
		2	1

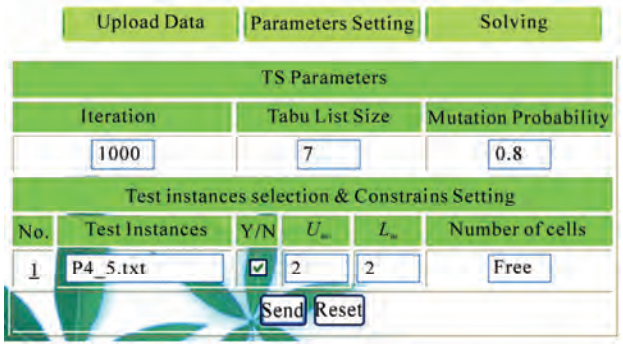


Figure 6. Web-based input interface for setting parameters and constrains

Part (PV)	Routing	Maching			
		2	4	1	3
1(50)	2	1	2		
3(20)	2	2	1		
2(30)	2			2	1
4(30)	2			1	2
5(20)	2			1	
Number of cells(NC)					2
Inter-Cell Movements(ICM)					0
CPU(s)					0.015

Figure 7. Web-based output interface for displaying CF results

Table 3. Test Instances From The Literature

No.	Source	Size(m×p×r)
1	Nair and Narendran [12]	8×20×20
2	Sofianopoulou [9]	12×20×26
3	Wu [18]	13×13×13
4	Sofianopoulou [9]	14×20×45
5	Gupta and Seifoddini [19]	16×43×43
6	Sofianopoulou [9]	18×30×59
7	Harhalakis *et al*. [10]	20×20×20
8	Nagi *et al*. [20]	20×51×51
9	Nair and Narendran[12]	25×40×40

5.2 Comparative Study

In order to evaluate the computational characteristics of our proposed CF solver with other approaches, ten test instances from the literature, as shown in Table 3 were used. The proposed CF solver was coded in Visual C++ programming languages and implemented on an Intel(R) 1.66 GHz personal computer with 1GB RAM. Table 4 shows the comparisons of our proposed CF solver with other approaches from the literature, that is, the GABB [10], the MIP [8] and the SA [9]. The bold characters indicate the best values obtained for each test problem. It can be seen from Table 4 that our proposed CF solver are better than or equal to those reported results. To be more specific, CF solver obtains for 3 problems values of the total intercellular movement that are equal to the best results found in GABB, MIP, and SA methods and improves the values for the rest 6 (67%) problems. It is worth to mention that our proposed CF solver was able to find the optimal solution in 1.547 seconds, illustrating the superiority of CF solver in solution efficiency.

Table 4. Performance of the proposed approach compared to other approaches

Test instances			Other approaches			Proposed approach		
No.	L_m	U_m	GABB[10]	MIP[8]	SA[9]	CF solver		
			ICM	ICM	ICM	NC	ICM	CPU (s)
1	2	4	13	-	-	2	13	0.273
2	2	5	-	-	29	3	29	0.500
3	2	6	-	1800	-	3	1260	0.360
4	2	5	-	-	25	3	25	0.578
5	2	6	-	34979	-	3	27416	0.907
6	2	7	-	-	34	3	32	0.774
7	2	5	18	-	-	5	17	0.953
8	2	5	86	-	-	5	82	1.789
9	2	4	39	-	-	7	33	1.547

6. Conclusions

Considering alternative process routings and machine sequences to cell formation (CF) problems makes the problems more realistic. New technologies, especially the World-Wide Web (WWW) technologies, have created many opportunities for research about Decision Support Systems (DSS). In this study, a Web-based CF DSS considering alternative process routings and machine sequences simultaneously has been proposed. With the assistance of CF DSS, the CF practitioners in the production departments can interact with the systems without knowing the details of algorithms and can get the best machine cells and part families with minimize the total intercellular movement wherever and whenever they may need it. An example taken from the literature was adopted for illustrational purpose. To further verify the feasibility and effectiveness of the system developed, a set of test problems with various sizes drawn from the literature have be used to test the performance of the CF solver. Corresponding results were compared to several well-known algorithms previously published. The results indicated that the proposed CF solver improved the best results found in the literature for 67% of the test problems and the CPU times took by our proposed CF solver to find the optimal solution were in 1.547 seconds. These show that our developed system should be very effective, efficient and practical.

REFERENCES

[1] S. Heragu, "Facilities design," Massachusetts: PWS Publishing Company, Boston, 1997.

[2] U. Wemmerlov and N. Hyer, "Research issues in cellular manufacturing," International Journal of Production Research, Vol. 25, pp. 413–431, 1987.

[3] A. Kusiak, "Intelligent manufacturing systems," New Jersey: Prentice Hall, Englewood Cliffs, 1990.

[4] Y. Yin and K. Yasuda, "Similarity coefficient methods applied to the cell formation problem: A taxonomy and review," International Journal of Production Economics, Vol. 101, pp. 329–352, 2006.

[5] A. Kusiak, "The generalized group technology concept," International Journal of Production Research, Vol. 25, pp. 561–569, 1987.

[6] M. S. J. Ameli and J. Arkat, "Cell formation with alternative process routings and machine reliability consideration," International Journal of Advanced Manufacturing Technology, Vol. 35, pp. 761–768, 2008.

[7] F. Boctor, "A linear formulation of the machine-part cell formation problem," International Journal of Production Research, Vol. 29, No. 2, pp. 343–356, 1991.

[8] Y. Y. Won and K. C. Lee, "Group technology cell formation considering operation sequences and production volumes," International Journal of Production Research, Vol. 39, No. 13, pp. 2755–2768, 2001.

[9] S. Sofianopoulou, "Manufacturing cells design with alternative process plans and/or replicate machines," International Journal of Production Research, Vol. 37, pp. 707–720, 1999.

[10] M. Boulif and K. Atif, "A new branch-&-bound-enhanced genetic algorithm for the manufacturing cell formation problem," Computers and Operations Research, Vol. 33, pp. 2219–2245, 2006.

[11] T.-H. Wu, S.-H. Chung, and C.-C. Chang, "Hybrid simulated annealing algorithm with mutation operator to the cell formation problem with alternative process routings," Expert Systems with Applications, Vol. 36, pp. 3652–3661, 2008.

[12] G. J. Nair and T. T. Narendran, "CASE: A clustering algorithm for cell formation with sequence data," International Journal of Production Research, Vol. 36, pp. 157–179, 1998.

[13] J. McAuley, "Machine grouping for efficient production," Production Engineer, Vol. 52, pp. 53–57, 1972.

[14] S. Lozano, B. Adenso-Diaz, and L. Onieva, "A one-step Tabu search algorithm for manufacturing cell design," Journal of the Operational Research Society, Vol. 50, pp. 509–516, 1999.

[15] G. Y. Tian, G. Yin, and D. Taylor, "Internet-based manufacturing: A review and a new infrastructure for distributed intelligent manufacturing," Journal of Intelligent Manufacturing, Vol. 13, No. 5, pp. 323–338, 2002.

[16] J. Lee, "E-manufacturing systems: Fundamental and tools," Robotics and Computer-Integrated Manufacturing, Vol. 9, No. 6, pp. 501–507, 2003.

[17] C. S. Smith and P. K. Wright, "CyberCut: A world-wide web-based design-to-fabrication tool," Journal of Intelligent Manufacturing, Vol. 15, No. 6, pp. 432–442, 1996.

[18] N. Wu, "A concurrent approach to cell formation and assignment of identical machines in group technology,"

International Journal of Production Research, Vol. 36, pp. 2099–2114, 1998.

[19] T. Gupta and H. Seifoddini, "Production data based similarity coefficient for machine-part grouping decisions in the design of a cellular manufacturing system," International Journal of Production Research, Vol. 28, pp. 1247–1269, 1990.

[20] R. Nagi, G. Harlarakis, and J. M. Proth, "Multiple routings and capacity considerations in group technology applications," International Journal of Production Research, Vol. 28, pp. 2243–2257, 1990.

J. Software Engineering & Applications, 2010, 3: 167-175
doi:10.4236/jsea.2010.32021 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

A Novel Method of Using API to Generate Liaison Relationships from an Assembly

Arun Tom Mathew[1*], C. S. P. Rao[2]

[1]Research Scholar, Department of Mechanical Engineering, National Institute of Technology, Warangal, India; [2]Professor, Department of Mechanical Engineering, National Institute of Technology, Warangal, India.
Email: aruntom123@gmail.com

Received November 17th, 2009; revised December 3rd, 2009; accepted December 20th, 2009

ABSTRACT

A mechanical assembly is a composition of interrelated parts. Assembly data base stores the geometric models of individual parts, the spatial positions and orientations of the parts in the assembly, and the relationships between parts. An assembly of parts can be represented by its liaison which has a description of its relationships between the various parts in the assembly. The problem is to not only make the information available but also use the relevant information for making decisions, especially determination of the assembly sequence plan. The method described in this paper extracts the feature based assembly information from CAD models of products and build up liaisons to facilitate assembly planning applications. The system works on the assumption that the designer explicitly defines joints and mating conditions. Further, a computer representation of mechanical assemblies in the form of liaisons is necessary in order to automate the generation of assembly plans. A novel method of extracting the assembly information and representing them in the form of liaisons is presented in this paper.

Keywords*: *Assembly, Mate Entities, Liaisons, Solidworks API

1. Introduction

The deployment of product models for planning assembly processes has received significant attention over the years and considerable research is happening in the area of assembly planning over the years. But assembly planning still poses a challenge like the description of assembly data and information specifically. There is much interest in reducing the cost of assembly activities. Assembly costs account for 10–30% of total industrial product labor costs [1], and as much as 50% of product manufacturing cost [2,3]. One way of achieving this is to improve assembly planning which aims to identify and evaluate the different ways to identify and evaluate the different ways to construct a mechanical object from its component parts. Due to frequent changes in product design and manufacturing methods, it is desirable to automate and computerize the planning activity. Sequence generation plays an important role in designing and planning the product assembly process. The choice of the assembly sequence in which parts of assembly are put together can drastically affect the efficiency of the assembly process. Identifying part interdependencies in assemblies and planning the process of assembly are examples of complex decision making activities. Assemblies contain a very large amount of information and complex relationships. An assembly planner is a system based on the geometric description of an assembly model identifies the parts that are involved in the construction of the assembly and generates the assembly plan. The model should provide a representation of parts and relationships such as contacts, degree of freedom among parts of assembly. Relational models represent geometric relations in the form of mating features between individual parts or subassemblies called liaisons. Of late, these parts or subassemblies are being designed using CAD programs, therefore the shape of each part and geometric information are already available in computer database. Since an effective description of assembly knowledge is very necessary, this information if extracted will be beneficial in identifying interdependencies between parts of the assembly and represent them in the form of a liaison diagram. In this paper, a novel approach of using the Automatic Programmable Interface (API) of the CAD software to extract the information which is then used to generate the liaison diagram which will be useful to generate assembly plans more efficiently.

The organization of the paper is as follows: Section 2 presents a literature review of assembly sequence generation; Section 3 describes the Generation of Assembly Relationships. Section 4 summarizes a different approach to the generation of assembly relationships using API

and building of a liaison diagram; Section 5 describes the system interface and the algorithm to extract relationships and generate liaisons; Section 6 gives an example of the approach and system and Section 7 gives concluding remarks.

2. Literature Survey

Over the years, a considerable progress has been made in the area of assembly planning specifically in the generation of assembly sequences. In general, assembly sequence planning consists of assembly modeling and assembly sequence generation. A lot of relevant information regarding the assembly could not be modeled and stored in the product while the assembly operation is done. The efficiency of assembly planning depends on the way the assembly information is modeled. Assembly information modeling is the base in this research, for generating assembly sequences. In the modeling of an assembly, the relation between the connected components must be established. The most commonly used method of assembly modeling is graph based called part mating graph [4] which represents the topological relationship between components of the assembly, where in the nodes represent the components and the arcs establish the relationship between the components. The mating conditions between two components provided by the designer are captured by a Virtual link mating graph [5]. Relational model graph includes parts, contacts and attachment relationships in a model [6]. De Fazio and Whitney called these mating graphs as liaison graphs [7]. Commercial CAD systems interpret assembly modeling as a means of providing functionality to the designer to easily position components with respect to each other [8]. Various detailed assembly representations have evolved including feature based [9], kinematics based [10] and geometry based [11]. Gottipolu and Ghosh [12,13] generated relationships by analyzing contact and mobility constraints. Laperriere and ElMaraghy [14] generated relationships using geometric and accessibility constraints. Generation of assembly relationships was also attempted using solid models by Chang [15]. Linn and Liu [16] described an algorithm developed to identify part liaison relationships presented in the commercial package, I-DEAS where the program processes geometric and topological data. Completely disassembling an assembly component based on the geometric contact relations results in the components explosion graph. Chen used the "contact above" concept to construct the Above Graph and then derive the Explosion Graph [17]. In the joint-based method [18], the assembly constraints are assigned on the components, but not on the geometric elements of the components. The method generates assembly models from kinematic joint constraints by extracting feasible joint mating features for each mating component, and then generates the assembly configuration for a set of joint constraints. Geometry-based representations capture the surface mating constraints like fit, coplanar, etc to establish the relations of precedence and feasibility [19]. The connection-semantics based assembly tree hierarchy provides a way to consider both geometric information and non-geometric knowledge of the assembly and obtain the degree of freedom of the mating entities [20]. Product semantic information model which is made of semantic information is structured into a three level semantic abstract, from which the relevant information is retrieved [21]. During assembly operation the designer specifies the relative location and the orientation of the components and surface mating constraints in order to accomplish the desired joints. All information regarding relationships between parts should be captured and used during the assembly planning process. In this paper, a novel method to extract this information between the assembly, subassembly and components using the API of the CAD modeling package and building the liaison matrix of the assembly is proposed.

3. Generation of Assembly Relationships

SolidWorks, the commercial CAD system, is used as the main feature-based design environment. The benefit of using SolidWorks is, it includes an entire API with functions that can be called from Visual Basic. In addition, SolidWorks shares the same solid modeling engine as Unigraphics and several other CAD systems like the Pro/Engineer and Catia. In addition, these CAD systems account for large user and application bases.

The description of the relationships among the features of various parts is required for an assembly component. These features can be classified into assembly features and primal features. It is the primal features that participate in assembly constraints. The assembly module automatically determines which relationship is meant by the user based upon the features involved in the relationship and updates the degrees of freedom accordingly. The primary mating conditions are align, mate, mate entity, align offset, insert, orient etc. The align condition requires that the axial centre lines of two parts be collinear. The mate condition requires that the two mating faces lie in the same plane with their outward normal opposing each other. The offset condition requires that the two faces lie in parallel planes with their outward normal in the same direction. The relationships between a pair of parts are specified by the user in terms of their features and the mating conditions between them. The individual parts in an assembly are created before the assembly module is invoked. The assembly modeling module requires information about the relationships between the part features. The information specified for each mating condition includes the ID of the mating feature and the type of mating conditions.

To build a list of all the characteristics of an assembly the assembly format is developed to store all the charact-

Figure 1. Flow of information

eristics in an assembly as its signature. The method explores in depth the assembly tree and extracts assembly related information for each part.

a) The method retrieves the constraints used to specify the position of the part.

b) It identifies which entities are used to constrain the part or subassembly.

c) It identifies the parent features and part of each geometrical entity in use.

4. Application Programming Interface (API)

Application programming Interface is an interface that defines the ways by which an application program may request services from libraries and/or operating systems. An API determines the terminology and calling conventions the programmer should employ to use the services. It may include specifications for routines, data structures, object classes and protocols used to communicate between the requesting software and the library.

An API may be Language-dependent, available only in a given programming language, using the syntax and elements of that language to make the API convenient to use in this context. It can also be Language-independent, written in a way that means it can be called from several programming languages. An API itself is largely abstract in that it specifies an interface and controls the behavior of the objects specified in that interface. The software that provides the functionality described by an API is said to be an execution of the API. An API is typically defined in terms of the programming language used to build the application. The API functions used in this paper are SolidWorks functions. The API functions are essential for developing the application software. The names of the mate features, the types, identities and the types of the mate surfaces, the mate clearances and the reference features etc are included in the mate information. There are three main SolidWorks document types namely Part Document, Assembly Document and Drawing Document. Each document type has its own object (PartDoc, DrawingDoc and AssemblyDoc) with its own set of related functions. For example, the AssemblyDoc::AddComponent4 method exists on the AssemblyDoc object because adding components is specific to assembly documents. The SolidWorks API also has functions that are common to all document types. For example, printing, saving, or determining the file name associated with a document would be common operations. To expose common document-level functions, the SolidWorks API uses the ModelDoc2 object. The ModelDoc2 object provides direct access to the PartDoc, DrawingDoc, and AssemblyDoc objects. As a general rule, the AssemblyDoc object provides access to functions that perform assembly operations; for example, adding new components, adding mate conditions, hiding and exploding components. The SolidWorks API also includes functions that are common to all document types; for example, determining the file name associated with a document is a common operation. To expose common document-level functions, the SolidWorks API uses the ModelDoc2 object. The structure of the assembly document is shown in Figure 2. The AssemblyDoc object is derived from the ModelDoc2 object. Therefore, an AssemblyDoc object has access to all of the functions on the ModelDoc2 object. The following objects related to the mate information like Mate, Mate Feature, Face and Surface. The Mate Object allows access to various assembly mate parameters. The MateEntity object enables access to mated objects and the assembly mate definition. The Feature Object allows access to the feature type, name, parameter data and to the next feature in the FeatureManager design tree. The mate information accessed through these objects in the FeatureManager design tree.

5. Mate Information

Most CAD systems represent the assembly using constr-

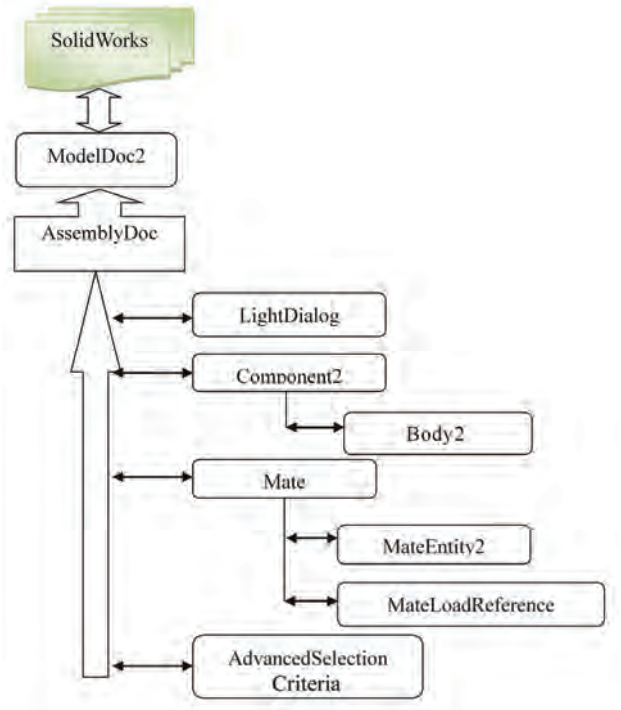


Figure 2. Detailed structure of SolidWorks assembly document

aint relationships between parts. A mate feature tree is obtained after the parts are assembled together using SolidWorks. With the help of SolidWorks API functions the mate constraints information is extracted by traversing the mate feature tree object. The information generated is represented in an object oriented way to generate assembly strategies especially assembly sequence plans.

A mate surface is a geometric surface of a part model that has mate relationships with other part models. A part model is composed of many surfaces, but only a few surfaces have mate relationships with other parts. Each surface is mated with the other using the mate entities (Table 1) generating a constraint relation. The constraint list includes all the geometrical constraints defined in the CAD model; each constraint in the list includes the constraint type, such as concentric, against and collinear, and mate tolerance, etc and its corresponding code as displayed in the Table 2.

The interface consists of a provision to load Solidworks.exe and create part models and assemble them. The part models are saved as .sldprt file and assembly models are saved as .sldasm file. A provision is provided to assemble the part models directly if they already exist.

5.1 Assembly Mate Extraction (AME) Algorithm

An algorithm is developed and executed using Visual Basic for Applications in order to access feature tree of the Solidworks assembly module.

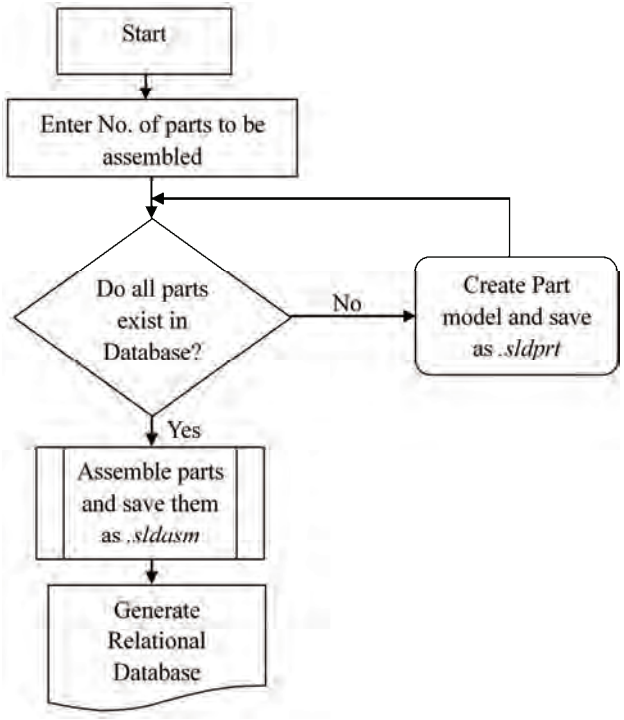


Figure 3. Process flow to generate relational database

The relationships between parts in the assembly are extracted using the Assembly Mate Extraction Algorithm and stored in a database. In addition to the mate type and mate entities types, the algorithm returns the follo- wing array of doubles: [*pointX, pointY, pointZ, vectorI, vectorJ, vectorK, halfangle, radius*]

where

- *pointX* is the X location of this mate entity in the assembly model space
- *pointY* is the Y location of this mate entity in the assembly model space
- *pointZ* is the Z location of this mate entity in the assembly model space
- *vectorI* is the i component of the assembly mate vector
- *vectorJ* is the j component of the assembly mate vector
- *vectorK* is the k component of the assembly mate vector
- *halfangle* is the value for the half angle
- *radius* is the value for the radius

Table 1. List of mate entity types

MATE TYPES	CODE
swMateUnsupported	0
swMatePoint	1
swMateLine	2
swMatePlane	3
swMateCylinder	4
swMateCone	5
swMateSphere	6
swMateCircle	7

Table 2. List of mate types

MATE TYPES	CODE
swMateCOINCIDENT	0
swMateCONCENTRIC	1
swMatePERPENDICULAR	2
swMatePARALLEL	3
swMateTANGENT	4
swMateDISTANCE	5
swMateANGLE	6
swMateUNKNOWN	7
swMateSYMMETRIC	8
swMateCAMFOLLOWER	9
swMateGEAR	10

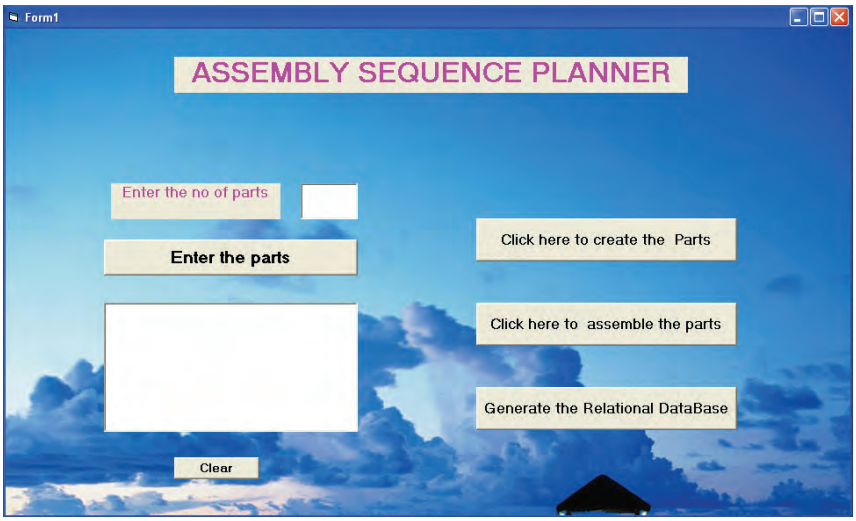


Figure 4. User interface between SolidWorks and Visual Basic

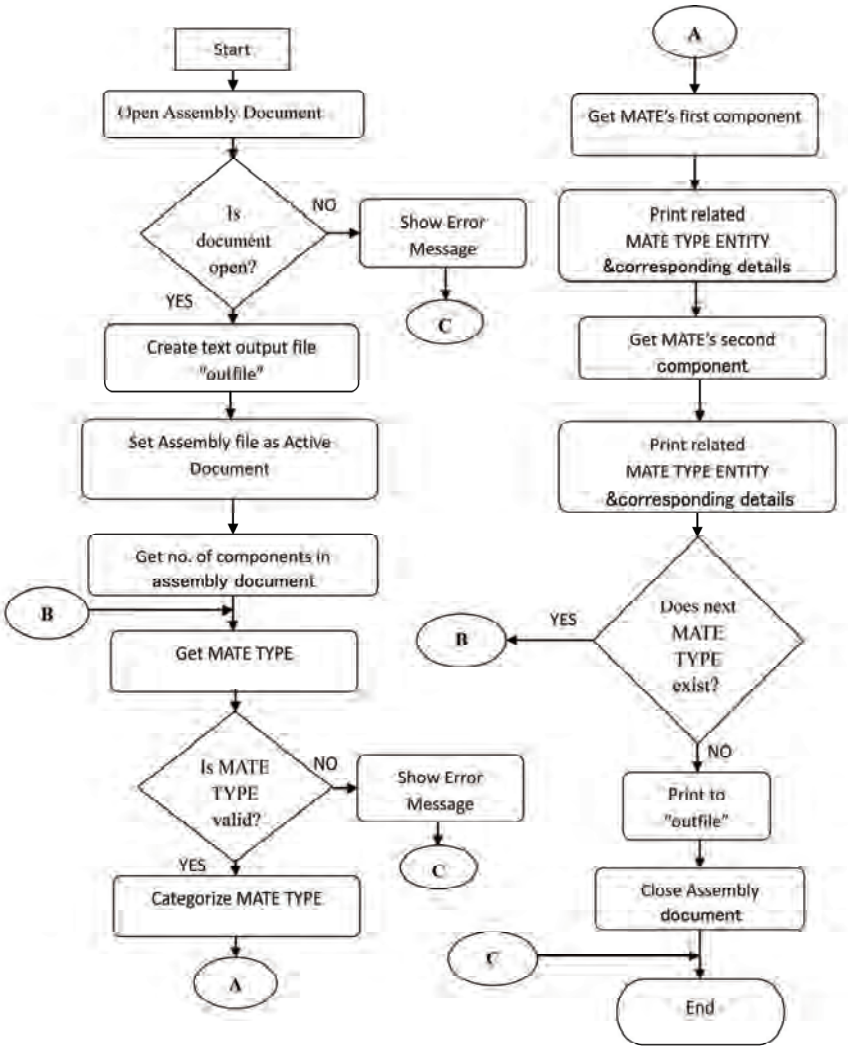


Figure 5. Flowchart for AME algorithm

To define the mate entity, the following information shown in Table 3 is returned based on the mate type. All coordinate information is given in terms of the assembly coordinate system where the mate resides.

The database contains the relations (or the Mategroups in Solidworks) between various parts of the assembly. There can be more than one relation between parts such as the axis of the two parts is aligned to each other and one face of the first part is coincident to the other. In addition to the basic relations, the database also contains the type of relation or mate whether it is a point contact, line contact or a plane contact as in the case. It can also be inferred whether the component is a cylindrical component or a prismatic component. If the program returns a value for the radius then the component is a cylindrical component else it is a prismatic component. The mass properties such as the weight, volume of individual components taking part in the process of assembly is also considered. The above information is necessary to assist in the selection of the base component for the assembly and to analyze the assembly plan with respect the Design for Assembly.

5.2 Symbolic Representation of Liaisons

Liaison graph can portray different logical and physical contact relations among the parts of the assembly. Liaison graph is a two-tuples $G = (P,L)$, where P is a set of nodes that represent parts, and L a set of edges that represent any of certain user defined relations between parts called liaisons. Given the liaisons graph, a decomposition method is used to systematically generate the assembly plans. The graph representation is difficult to process using a computer, but can easily handle the information in a matrix form. The table of liaisons or the liaison matrix is used to represent the contact information between the components. If one or more relation is found to exist between 2 pairs of components of the assembly, then that pair will have a value one else zero.

Table 3. Information is returned based on the mate type

Mate Type	Information Returned
swMatePoint	*pointX, pointY, pointZ*
swMateLine	*pointX, pointY, pointZ, vectorI, vectorJ, vectorK* where the point is a point on the line and the vector represents the line direction.
swMatePlane	*pointX, pointY, pointZ, vectorI, vectorJ, vectorK* where the point is a point on the plane and the vector represents the plane normal.
swMateCylinder	*pointX, pointY, pointZ, vectorI, vectorJ, vectorK, halfangle* where the point is a point on the cylinder axis and the vector represents the cylinder axis.
swMateCone	*pointX, pointY, pointZ, vectorI, vectorJ, vectorK, halfangle, radius* where the point is a point on the cone axis and the vector represents the cone axis.

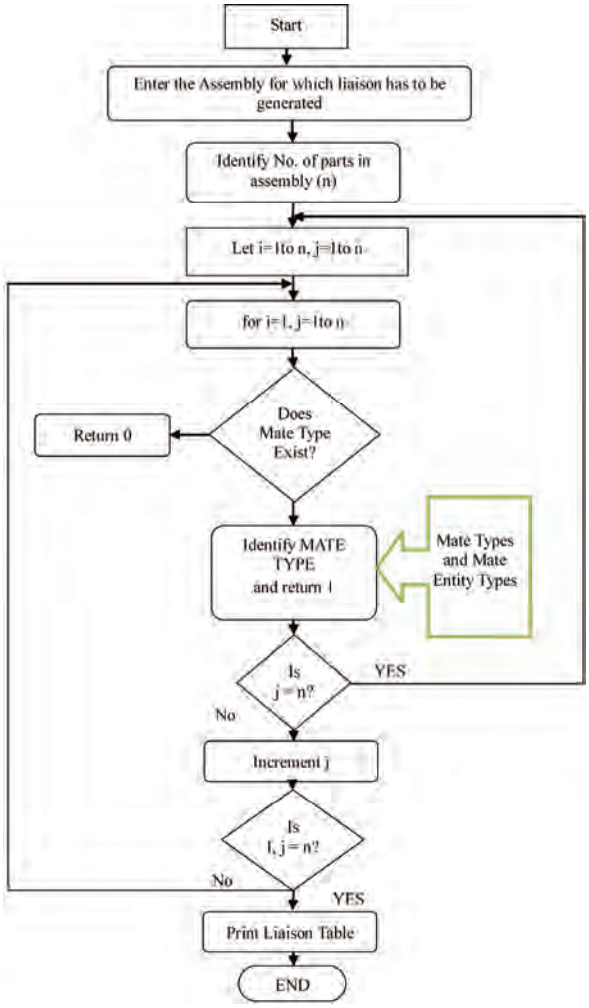


Figure 6. Flowchart for LTG algorithm

i.e, if $L_{i,j} = 1$, then relation exist between component 'i' and 'j'

if $L_{i,j} = 0$, then no relation exist between component 'i' and 'j'

The Matrix Cell *ij* contains the value of L as 0 or 1.

It assumed that between two parts there is only one edge representing a liaison that includes all contacts such as collinear, coincident etc. There is no liaison joining a vertex to itself because it is impossible to assemble a component to itself. A component belongs to the product it has at least a liaison with another component of the assembly. If a component p_i can be assembled to p_j, the reverse is also true. An algorithm named Liaison Table Generator (LTG) Algorithm is developed to generate the liaison matrix based on the relationships existing among parts and is described in Figure 6.

6. Implementation with an Example

The clamping fixture (Figure 7) taken for the present study has 7 components namely the base, plunger, link, knob, lever, clamp-end and pin. SolidWorks software was used to model the assembly. The individual components were created as separate geometric models in the part models in the part mode and saved as ".sldprt" files. Next, the assembly modeling mode is invoked and the Base is taken as the support component. After specifying the assembly constraints, the assembly was built by adding the remaining components to the casting. All the components are assembled using the mate attributes like the Coincident, Parallel, Perpendicular, Tangent, Concentric, Distance and Angle. The completed assembly model is then saved as a ".sldasm" file.

The feature tree of the clamping fixture modeled and assembled is generated in the assembly module after adding all the mating conditions is shown in Figure 8. This feature tree appears on the left side of the Assembly module. The mate relationship information stored in this tree is extracted using the AME algorithm and stored in a notepad file as shown in Figure 9. The AME algorithm is implemented in a computer program coded in Visual Basic for Applications (VBA) in the Microsoft Windows platform. VBA is chosen, essentially for availability and portability reasons in addition to the ability to interface with SolidWorks. The machine used to run the program was a notebook with a Pentium Centrino Duo 1.6 GHz processor, with 1 GB DDR RAM.

Once the notepad file is generated, it acts as a database, from which the liaisons between the various components of the assembly can be generated. The LTG algorithm is applied to the clamping fixture assembly and the liaison table with relations is generated. The LTG algorithm is implemented JAVA and resultant liaison table is shown in Figure 10. The output of the program gives the number of parts in the assembly, mating relationships that exist between various parts of the assembly and finally displays the liaison diagram or table showing the relations.

7. Conclusions

The liaison information contains relationships between parts is the basic information needed for assembly sequence planning. Although many modeling packages provide information about the solid models, the information regarding to the relationships between parts in the assembly is not explicitly available. In this paper, a novel method of determining the relationship liaison diagram is proposed. The method first extracts the assembly information, processes it and then generates liaisons. The mating feature matrix of two contacting parts in the assembly is established. The technique is fully automated, simplifies the process of extracting geometrical constraints for any given assembly considering the relationships between components of a CAD model using the

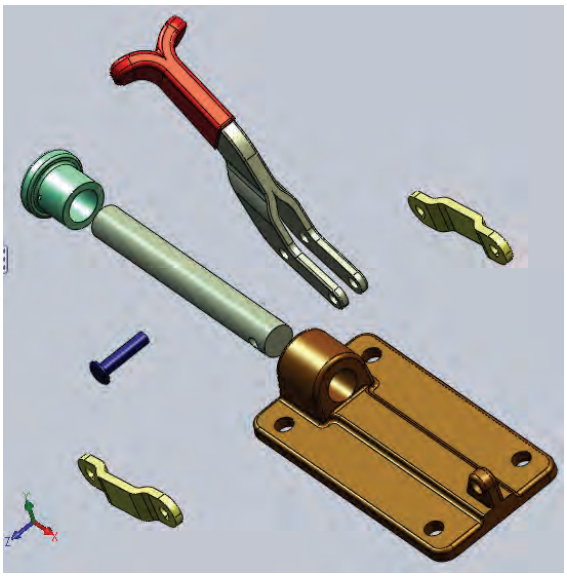


Figure 7. Example of clamping fixture (exploded view)

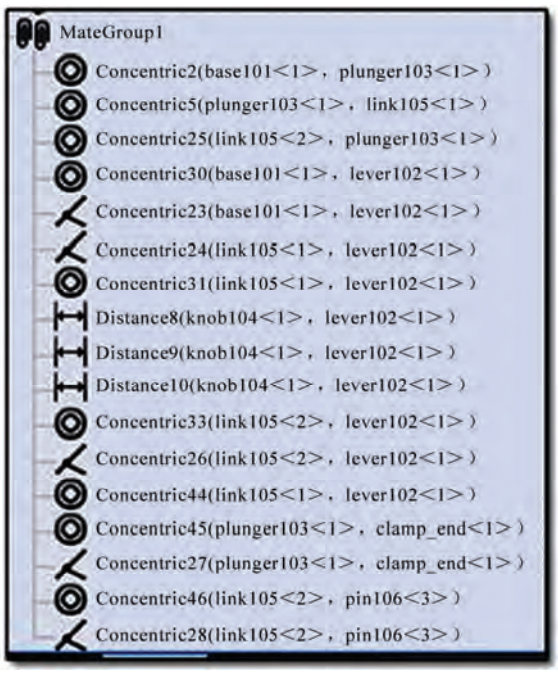


Figure 8. Feature tree of the assembly created in SolidWorks

Automated Programmable Interface of the CAD software. The main constraint in this approach is that the designer should model the assembly components and define the mating conditions while assembling the product. Once the database containing the relationships is extracted, it can be used to generate the liaison diagram and liaison tree. Taking the precedence relations into consideration, feasible assembly sequences of a given assembly can be generated.


```
" File=D:\SolidWorka Working Directory\clamping fixture \clamping_fixture.sldasm "
" MateGroup1
"  "
"  " Concentric2 "
"       Type             =1 "
"       AlignFlag        =1 "
"       CanBeFlipped     =False "
"  "
"       Component        =base101-1 "
"       MateEntType      =4 "
"       (x,y,z)          =(-6.09316913443831E-02, 2.60376887513812E-02, -1.62423420159542E-16) "
"       (i,j,k)          =(1,0,0) "
"       Radius 1         =0.00635 "
"       Radius 2         =0 "
"  "
"  " Concentric2 "
"       Type             =1 "
"       AlignFlag        =1 "
"       CanBeFlipped     =False "
"  "
"       Component        =plunger103-1 "
"       MateEntType      =4 "
"       (x,y,z)          =(-2.06503960267056E-02, 2.60376887513812E-02,-1.62423420159542E-16) "
"       (i,j,k)          =(-1,0,0) "
"       Radius 1         =0.00635 "
"       Radius 2         =0 "
"  "
"       - - - - - - - - - - - - - - - - - - - - - - - - - - - - "
"  " Concentrics "
"  "
"    Concentrics "
"       Type             =1 "
"       AlignFlag        =1 "
"       CanBeFlipped     =False "
"  "
"       Component        =plunger103-1 "
"       MateEntType      =4 "
"       (x,y,z)          =(-2.70003960267056E-02, 2.60376887513812E-02, -6.350000000000E-03) "
"       (i,j,k)          =(0, 0,1) "
"       Radius 1         =2.38125000000001E-03 "
"       Radius 2         =0 "
"  "
"  " Concentrics "
"       Type             =1 "
"       AlignFlag        =1 "
"       CanBeFlipped     =False "
"  "
"       Component        =link105-1 "
"       MateEntType      =4 "
"       (x,y,z)          =(-2.70003960267056E-02, 2.60376887513812E-02,-6.350000000000E-03) "
"       (i,j,k)          =(1.15754264840233E-16,-5.18983244566464E-16,-1) "
"       Radius 1         =0.00238125 "
```

Figure 9. Snapshot of the relational database generated

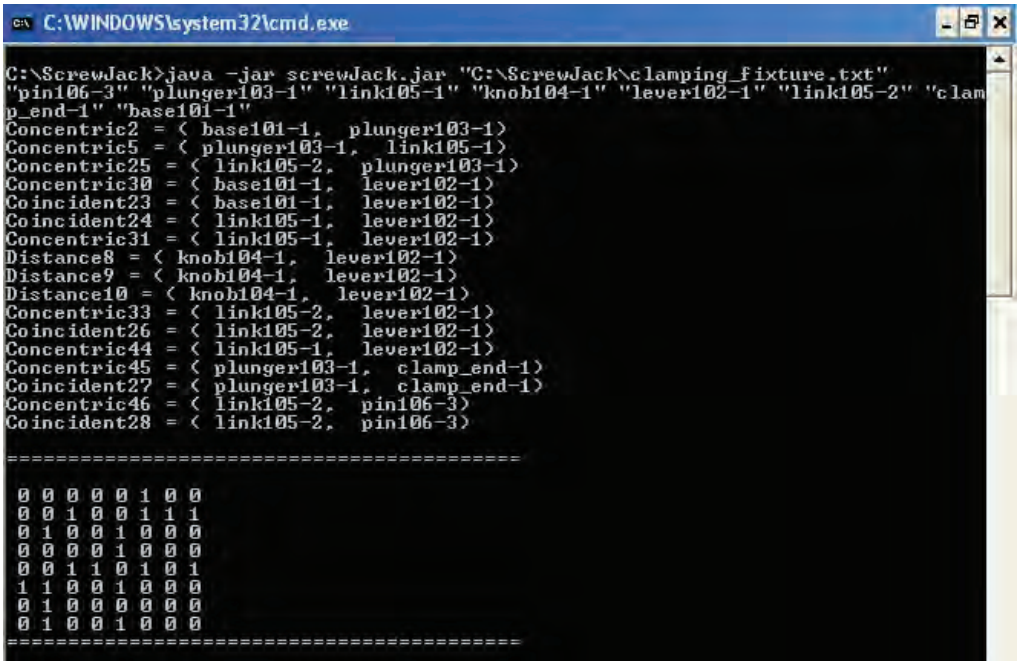


Figure 10. Liaison table generated

REFERENCES

[1] J. L. Nevins and D. E. Whitney, "Concurrent design of product and processes," McGraw-Hill, New York, 1989.

[2] U. Rembold, C. Blume, and R. Dillmann, "Computer-integrated manufacturing technology and systems," Marcel Dekker, New York, 1985.

[3] S. S. F. Smith, "Using multiple genetic operators to reduce premature convergence in genetic assembly planning," Computers in Industry, Vol. 54, Iss. 1, pp. 35–49, May 2004.

[4] C. M. Eastman, "The design of assembly," SAE Technical Paper Series 0148-7191/81/0223-0197, 1981.

[5] H. Ko and K. Lee, "Automatic assembly procedure generation from mating conditions," Computer Aided Design, Vol. 19, pp. 3–10, 1987.

[6] L. S. Homen De Mello, and A. C. Sanderson, "Representations of mechanical assembly sequences," IEEE Transactions on Robotics and Automation, Vol. 7, No. 2, pp. 211–227, 1991.

[7] D. F. Baldwin, T. E. Abell, M. C. M. Lui, T. L. D. Fazio, and D. E. Whitney, "An integrated computer aid for generating and evaluating assembly sequences for mechanical products," IEEE International Conference on Robotics and Automation, Vo1. 7, No. 1, pp. 78–94, 1991.

[8] K. W. Lyons, V. N. Rajan, and R. Sreerangam, "Representations and methodologies for Assembly Modeling," NIST Int. Rep., Gaithersburg, MD, 1996.

[9] J. J. Shah and M. T. Rogers, "Assembly modeling as an extension of feature based design," Recent Trends in Engineering Design, Vol. 3, No. 3 & 4, pp. 218–237, 1993.

[10] C. Mascle, "Features modeling in assembly sequence and resource planning," In Proceedings IEEE International Symposium on Assembly and Task Planning, Pittsburgh, PA, pp. 232–237, 1995.

[11] R. Anantha, G. A. Kramer, and R.H. Crawford, "Assembly modeling by geometric constraint satisfaction," Computer Aided Design, Vol. 28, No. 9, pp. 707–722, 1996.

[12] R. B. Gottipolu and K. Ghosh, "An Integrated approach to the generation of assembly sequences," International Journal of Computer Applications in Technology, Vol. 8, No. 3–4, pp. 125–138, 1995.

[13] R. B. Gottipolu and K. Ghosh, "A simplified and efficient representation for evaluation and selection of assembly sequences," Computers in Industry, Vol. 50, pp. 251–264, 2003.

[14] L. Laperriere and H. A. ElMaraghy, "Assembly sequences planning for simultaneous engineering applications," International Journal of Advanced Manufacturing Technology, Vol. 9, pp. 231–244, 1994.

[15] A. C. Lin and T. C. Chang, "3D MAPS: Three dimensional mechanical assembly planning system," Journal of Manufacturing Systems, Vol. 12, No. 6, pp. 437–456, 1993.

[16] R. J. Linn and H. Liu, "An automatic assembly liaison extraction method and assembly liaison model," IIE Transactions, Vol. 31, pp. 353–363, 1996.

[17] R. S. Chen, K. Y. Lu, and P. H. Tai, "Optimizing assembly planning through a three-stage integrated approach," International Journal of Production Economics, Vol. 88, pp. 243–256, 2004.

[18] J. S. Kim, K. S. Kim, J. Y. Lee, and J. H. Jeong, "Generation of assembly models from kinematic constraints," International Journal of Advanced Manufacturing Technology, Vol. 26, pp. 131–137, 2005.

[19] R. Sudarsan, Y. H. Han, S. Foufou, S. C. Feng, U. Roy, F. Wang, R. D. Sriram, and K. Lyons, "A model for capturing product assembly information," Journal of Computing and Information Science in Engineering, Vol. 6, No. 1, pp. 11–21, 2006.

[20] T. Dong, R. Tong, Ling, and J. Dong, "A Knowledge based approach to assembly sequence planning," International Journal of Advanced Manufacturing Technology, Vol. 32, pp. 1232–1244, 2007.

[21] H. Wang, D. Xiang, G. Duan, and L. Zhang, "Assembly planning based on semantic modelling approach," Computers in Industry, Vol. 58, pp. 227–239, 2007.

J. Software Engineering & Applications, **2010, 3: 177-184**
doi:10.4236/jsea.2010.32022 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Feature Extraction and Diagnosis System Using Virtual Instrument Based on CI

Renping Shao, Xinna Huang, Yonglong Li

School of Mechatronics, Northwestern Polytechnical University, Xi'an, China.
Email: shaorp@nwpu.edu.cn, huangxn@nwpu.edu.cn

Received August 9th, 2009; revised September 7th, 2009; accepted September 30th, 2009.

ABSTRACT

Through investigating intelligent diagnosis method of Computational Intelligence (*CI*) *and studying its application in fault feature extraction, a gear fault detection and Virtual Instrument Diagnostic System is developed by using the two hybrid programming method which combines both advantages of VC++ and MATLAB. The interface is designed by VC++ and the calculation of test data, signal processing and graphical display are completed by MATLAB. The program converted from M-file to VC++ is completed by interface software, and a various multi-functional gear fault diagnosis software system is successfully obtained. The software system, which has many functions including the introduction of gear vibration signals, signal processing, graphical display, fault detection and diagnosis, monitoring and so on, especially, the ability of diagnosing gear faults. The method has an important application in the field of mechanical fault diagnosis.*

Keywords**:** *Virtual Instrument* (*VI*), *Computational Intelligence* (*CI*), *Fault Diagnosis*, *Feature Extraction*, *Gear System*

1. Introduction

At present, a gear transmission is one of primary driving-forms in mechanical transmission and is broad applied in practical engineering. Gears are used in most of equipments such as aero-engine, vehicle and machine tool. However, studies have shown that 60% faults in gear system cases are caused by gear failure and 90% gear failures are due to partial failure, such as crack, wear abrasion of tooth and so on. Therefore, the condition monitoring and fault diagnosis for gears can not only avoid the decline of equipment's accuracy during working in order to reduce or to cease the appearance of accident, but also adequately exert the potential gear work. It has an important significance in both of economic and social benefit [1].

After combining the analysis method of time-domain, frequency-domain, and some intelligent diagnosis methods such as the ones based on support vector machine, artificial neural network and so on, in the field of gear fault diagnosis, it is necessary to develop a virtual instrument system of gear failure analysis so that these analysis methods can are expediently applied in gear fault diagnosis [2]. In the traditional time-domain and frequency-domain analysis and the ones based on Support Vector Machine, artificial neural network, pattern recognition, it is related to the portion such as numerical calculation, signal processing, graphical display etc. Therefore, it is necessary to synthesize the above studies mentioned to empolder a virtual instrument diagnosis system. Wan and Tong designed a fault diagnosis system based on virtual instrument, which was successfully applied to SDF-9 generator [3]. Lv and Zhang used virtual instrument technique designed a fault diagnostic system, which solved the problems of state prediction and trouble-mode recognition of warships equipment [4]. The literature [5] studied the remote fault diagnosis for complex equipment based on virtual instrument (VI) technology. Xu *et al.* introduced a fault testing and diagnosis system of bearing in armored vehicle based on information fusion technology [6]. However, above the most are subject to the restrictions of their software system, the diagnostic functions of system are not very comprehensive, and the diagnosis systems lack the functions of online real-time detecting and monitoring.

Virtual instrument (VI), which brings the fault identification and diagnosis to a higher level, is an important development trend with combination of testing technology and fault diagnostic techniques. It considers computer software as its core. It possesses all the functions of signal collection, analysis, processing, display, identification, diagnosis, alarm, monitoring, and it is the developing direction of intelligent detection and diagnosis. Wang and Gao investigated the design, optimization, and

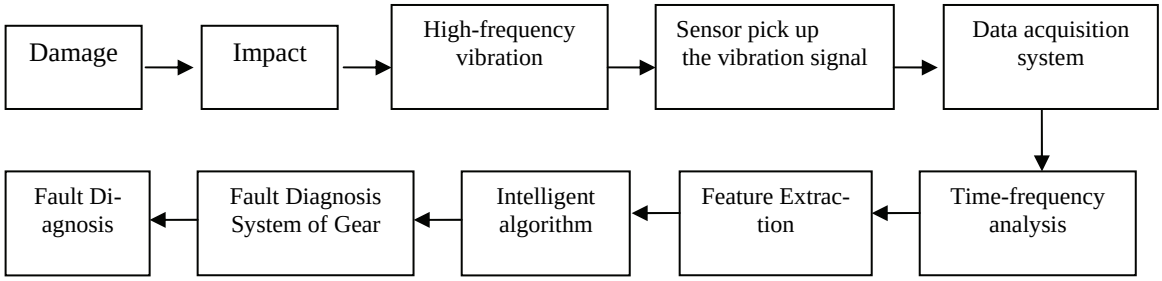


Figure 1. Block diagram of signal extraction and fault diagnosis

implementation of virtual instrument (VI), which was an essential part of integrated bearing condition monitoring system [7]. The literature [8] introduced an approach combining virtual instrument with rough set theory (RST) for FPC on-line fault diagnosis. Betta *et al.* analyzed the use of Support Vector Machines (SVMs) in software-based instrument fault accommodation schemes for automotive systems [9]. The literature [10] researched the application of virtual instrument and grey theory in the fault diagnostic system. However, nowadays the virtual instrument systems have Labview, Labwindows/CVI, HP VEE, and so on. But due to these software systems' limitation, it is difficult to run independently without the inherent system respectively.

MATLAB, which has strong graphical display capability, is a very powerful matrix and mathematical software package for engineering and scientific calculations with advanced file I/O. VC++ is a very useful language widely used in computer-controlled measurement system as well. The combination of MATLAB and VC++ can play their respective advantages with powerful functions such as: analysis and processing of signal, visualization, diagnosis and monitoring, etc. The interface developed by VC++ is friendly to user, and the system developed with MATLAB has the following functions, such as numerical analysis and calculation, graphics display, testing and detecting, etc. The literature [11] introduced a MATLAB and VC++ mixed programming method, which is used in their differential optic absorption spectroscopy (DOAS) atmospheric pollution monitoring system. Xu *et al.* studied a hybrid programming method, which is combined with VC++ and MATLAB, applied to power system fault analysis and identification system [12]. Chen *et al.* investigated several common mixed programming methods between VC++ and MATLAB based on their respective characteristics [13].

Therefore, this paper based on technology platform of virtual instrument will empolder the main program of gear failure analysis software systems by using MATLAB software tools and VC++ language. The interface of between VC++ and MATLAB is developed. Using the two hybrid programming method, the calculation of test data, signal processing and graphics display are completed by MATLAB. Utilizing MATCOM tool, it transforms program written the MATLAB into program functions called VC++ which is used to design the main interface and achieve the sub-interface's call, then develops a gear failure analysis software system. It can run independently without MATLAB, and possesses the functions of virtual instrument diagnosis system, included many functions, such as signal collection, signal analysis, processing, display, identification, on-line diagnosis, alarm and monitoring.

2. Fault Diagnosis and Feature Extraction Mechanism

Vibration diagnosis is a basic method in fault diagnosis technology, it commonly used in the mechanical equipment condition monitoring [14]. Vibration will be increased when an exception occurs in the machine. Therefore, according to the measurement and analysis for the mechanical vibration signal, we can know the estate of their deterioration and the property of failure without stopping running and disassembly. The process of fault diagnosis using vibration signal is shown in Figure 1.

The diagnosis method is a very active research field in the fault diagnosis, and the key of diagnosis method is the research of extraction mechanism of failure feature. Namely, which measure method has been taken and what kind of sensitive signals have been extracted to improve the identification accuracy and detecting ability of system. How to achieve the most effective real-time monitoring and diagnosis will be the key of modern fault diagnosis and monitoring. It has become a hot topic in the current research on how to detect mechanical fault in time and forecast the developing trend of equipment operation condition. It is the key of diagnostic success in how to develop a compositive system which is combined with software and hardware to achieve real-time, on-line diagnosis for mechanical drive system. The rapid development of virtual instrument technology, which introduces the mechanical fault diagnosis and testing technology into the higher level field, it is an important trend of development of testing technology

and fault diagnosis technology [15,16].

Computational Intelligence (CI), which developed from Fuzzy, AI, Neuron (a joint name "FAN"), is a burgeoning modern signal processing and control theory with decennary development. It is a strongly all-around advanced theory with a unique advantage which is related to many fields such as AI (artificial intelligence), Fuzzy, Neural Network, GA (genetic algorithm), Chaos, Fractal, Granular Computing, and Biomedicine. Using CI can extract coupling highly order useful features from random non-linear and non-stationary data and remove the effect of adscititious noise. It has become a brand-new method for pattern recognition and fault diagnosis of complex dynamic system, and a hot spot in current international research [17] as well.

Virtual instrument (VI), a new instrument concept put forward in early 90s of 20th century, is a combination of computer and instrument with a breakthrough for traditional instrument concept. Using adequately the intelligent computer-function such as computing, storage, playback, call, display and document management, complex information processing, it has achieved the specialized function of traditional instrument by software. The new instrument, whose appearance and function are completely identical with the traditional hardware instruments, fully share the computer intelligence resources. It has become the developing direction of instruments and equipment. Therefore, software development is the key of the virtual instrument exploiture [18,19].

To establish a fault detection system needs to do as follows: including modern signal processing technology with artificial neural networks, support vector machines, genetic algorithms and other computational intelligence methods to extract failure feature of mechanical systems, then diagnose the typical fault of mechanical system and establish the fault detection system [20,21]. With the investigation of fault diagnosis system based on the virtual instrument technology and establishment of the software and hardware system of virtual instrument, which can collect signal, analyze and diagnose signal, the development of entire fault diagnosis system is completed. It can also achieve the on-line detection and diagnosis of failure for mechanical drive system. Consequently, a low-cost, highly intelligent, detection and diagnosis system is introduced to meet the needs of practical engineering.

In this paper, the detection and diagnosis system based on the VI platform include two major parts, hardware and software are developed. Hardware takes PC (or platform) and I/O interface devices, as well as data acquisition cards to complete the task of signal acquisition. Software is the primary part of entire systemic development to complete the signal analysis and detection, fault diagnosis and results display. Under the support of hardware platform considering computer as the core, the instrument function is achieved by software programming (such as the use of MATLAB and C++ language, etc.) design, and multifarious testing and fault diagnosis analysis functions are achieved by the combination of different software modules of the test function.

3. Conceiving of Virtual Instrument Diagnosis System

The two major characteristics of MATLAB are powerful matrix calculation and graphical display. It integrates numerical analysis, matrix computation, signal processing and graphical display into a convenient, user-friendly environment. However, some shortcomings of MATLAB itself restrict its application:

1) MATLAB is an interpretative language, so its real-time efficiency is very poor.

2) MATLAB program cannot run without their environment, so it can not be used for commercial software development.

3) The source code of MATLAB can be seen directly, so it doesn't avail to confidentiality of algorithm and data.

The object-oriented visual programming of VC++ is used to develop software applications from the bottom to the user-oriented software and other. By utilizing its various practical tools, developers can easily develop powerful high-performance Windows applications program. However, in practical engineering development, compared with MATLAB:

1) VC++ is not as good as MATLAB in numerical disposal analysis and algorithms tool and other aspects.

2) VC++ is not as good as MATLAB in accurately and expediently mapping data Graphics (data visualization).

Therefore, if we combine the advantage of MATLAB in numerical calculation, algorithm design and data visualization and other areas with VC++ application system, it can not only fully meet the need in data calculation and display of system, but also improve system efficiency and stability to reduce the difficulties of achieving algorithm, shorten the cycle of software development and improve software quality. It has highly valuable in practice. This system designedly makes use of its two major characteristics to achieve analysis process, diagnose and related graphics display of gear vibration signal.

The software, which could be run without MATLAB, can be developed by MATCOM transform method. VC++ is responsible to open the document

and for the main interface, while MATLAB is responsible for numerical computation, signal processing and graphics display and other functions. The M-file can be converted into transferable cpp-file and relevant h-file of VC++ by MATCOM. Finally, the development of system is completed. The system, which has multi-functional interfaces, includes its chief: the main function interface, the various interfaces of gear vibration signal analysis, the various interfaces of gear vibration signal diagnosis and identification, the interfaces of gear vibration signal detection and monitoring. Each interface includes corresponding sub-interface possessing functions of analysis and diagnosis, as shown in Figure 2.

4. Exploitation of Virtual Instrument Diagnosis System

The software supports to collect the vibration acceleration signal data of gear system, including 4-channel data and 2-channel data. Because the program in signal analysis interface has been converted into cpp-file and corresponding h-file, so the software can run without MATLAB software environment in the Windows operating system and has been successfully tested under Windows XP.

4.1 Data Acquisition System

Gear system testing device is shown in Figure 3. In the figure, electromotor drive the entire system, and the coupling transfer the power to reducer gear, and after reducer export the power which pass the gear-coupling and torque speed sensor, the power is transmitted to the magnetic loader. The common experiment method is via testing the vibration signal of gear box, which the sensor stick on, to test the dynamic characteristics of system and gear, however, with a large system noise, the characteristics of gear fault signal will be weakened when it reach gear box. To highlight the characteristics of test signals to minimize the effects of noise on gear failure signal, in this experiment the acceleration sensors should be fixed on the gear and the sensors should be near the symmetrical location of gear fault. Driving-gear is set as fault-gear with the use of transmission ring, because of which the signal acquired from the rotary acceleration sensor is transmitted out fully. Because the signal acquired from the acceleration sensor is comparatively faintly, the signal has to be amplified by amplifier to collect. In the test, by using control cabinet, we observe and control the magnitude of the torque and speed, and observe the magnetic loader to control the magnitude of load given by system.

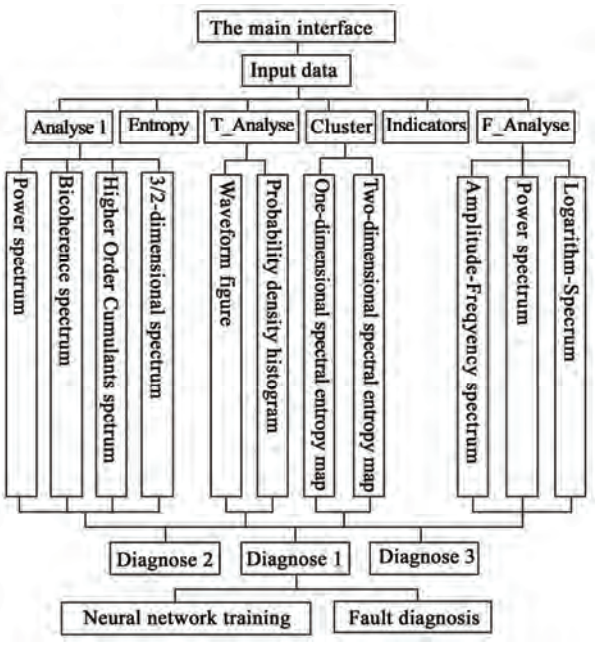


Figure 2. Software system structure

The running of gear system has four levels of speed: no-load with speed of 300r/min, load 10N.m with 300r/min, load 8N.m with 900r/min, load 6N.m with 1200r/min, load 6N.m with 1500r / min. Three typical failures (crack at the gear root, crack at the gear's reference circle and the wear abrasion fault on tooth surface) and various composite faults are set on driving-gear and tested at above four levels of speed. CRAS signal acquisition software system is used for signal acquisition and analysis, and the algorithms mentioned above are used for feature extraction and diagnosis.

4.2 The Introduction and Test of Signal

The main interface is shown in Figure 4. It mainly achieves the signal acquired from the actual gear transmission testing system in order to facilitate subsequent analysis, identification, and diagnosis. The top column is used to show the physical address of imported data; "ChannelNum" is used to set the total channel number of data files; "Channel" is used to set the channel sequence of importing data; "DataLength" is used to display imported data length; "Fs (Hz)" is used to set the sampling frequency. After the parameters have been set, pressing the "InputData" button, we can open the dialog frame of data file to select the data which will be analyzed. If the data format is correct, the bottom graph-displayed frame will display the waveform of the imported data, so it shows the success of importing data. The right column of buttons, whose functions and manipulations would be introduced one by one in the latter parts, is used for signal processing, calculation and analysis, diagnostic tests.

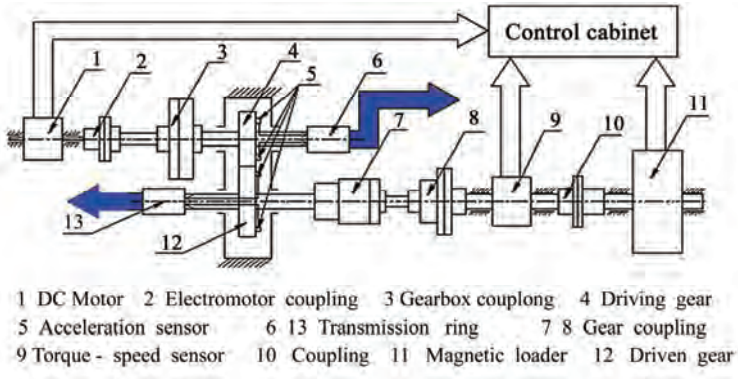


Figure 3. Testing equipment and data acquisition of gear system

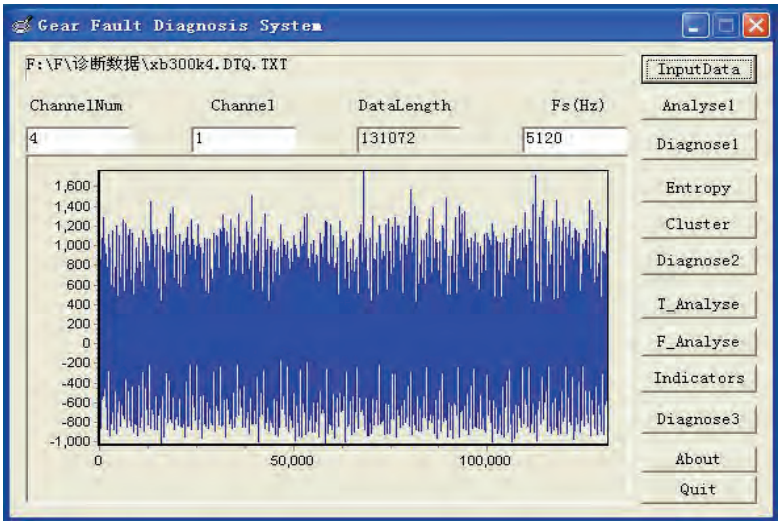


Figure 4. The main interface of software system and the display of interface after importing data

4.3 Signal Analysis and Process

Signal process and analysis include six buttons of Analyse1, Entropy, Cluster, T_Analyse, F_Analyse, Indicators. The following introduce respectively their function and detailed operation.

1) The function and operation of Analyse1

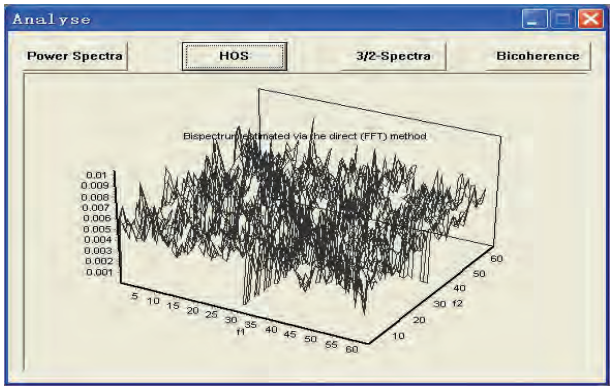
Analyse1's part of the interface is shown in Figure 5. Function: It can realize the analysis calculation and figuring graph of power spectrum, higher-order cumulant spectrum, 3/2 dimensional spectrum, bicoherence spectrum. Data power spectrum is plotted by clicking the Power Spectra button; 2 dimensional (contour map) and 3 dimensional (grid chart) spectrogram of higher-order cumulant of data can be displayed by clicking HOS button, as shown in Figure 5(a); Clicking the 3/2-Spectra button is used to achieve the drawing of 3/2 dimensional spectrum; 2 dimensional (contour map) and 3 dimensional (grid chart) spectrogram of bicoherence spectrum of data can be displayed by clicking Bicoherence button, as shown in Figure 5(b).

2) The function and operation of Entropy

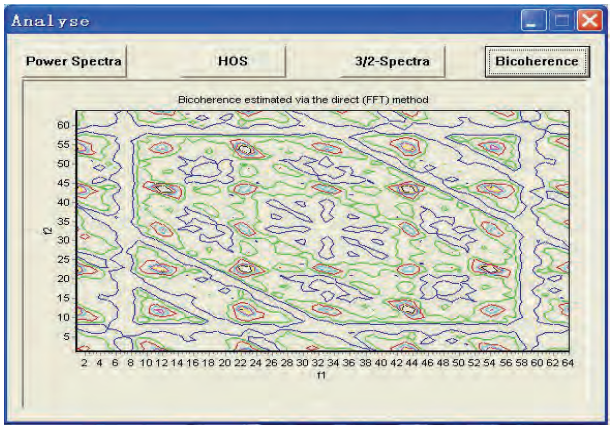
Entropy's part of the interface is shown in Figure 6. Function: It calculates the spectral entropy of the imported data owning numbers of cycles. "Mean" denotes the mean of spectral entropy, "Max" denotes the max of spectral entropy, "Min" denotes the min of spectral entropy, "Beta" denotes the kurtosis of spectral entropy, and "S" denotes the variance of spectral entropy. The underside graph display frame shows the distribution of calculated value of spectral entropy.

3) The function and operation of Cluster

Cluster's part of the interface is shown in Figure 7. Function: It is used to cluster analysis for the imported data, including the K-mean method and the den-grid method. In the figure, "•" denotes wear abrasion faults, "×" denotes crack faults, "*" denotes the feature of without faults. It is obvious that this method can accurately cluster for different signals of gear system.



(a) Three-dimensional grid spectrum of higher-order cumulant



(b) Two-dimensional contour line of bicoherence spectrum

Figure 5. The display interface of Analyse1

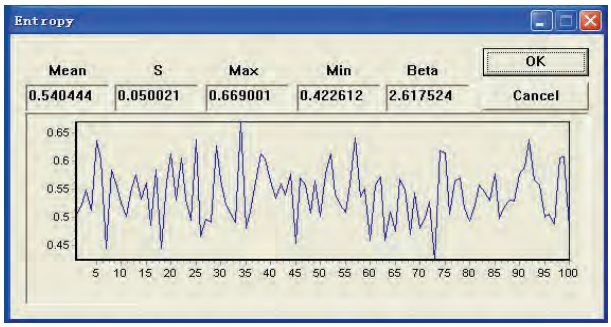


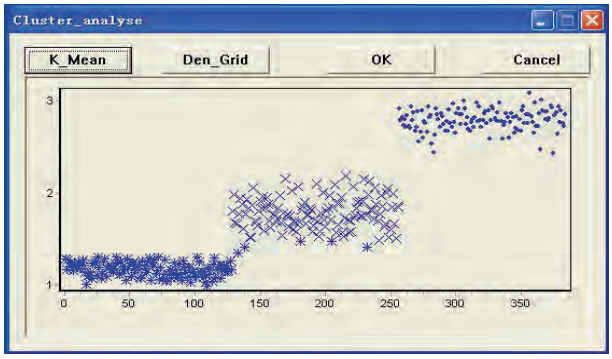
Figure 6. The display interface of Entropy

4) The function and operation of T_Analyse

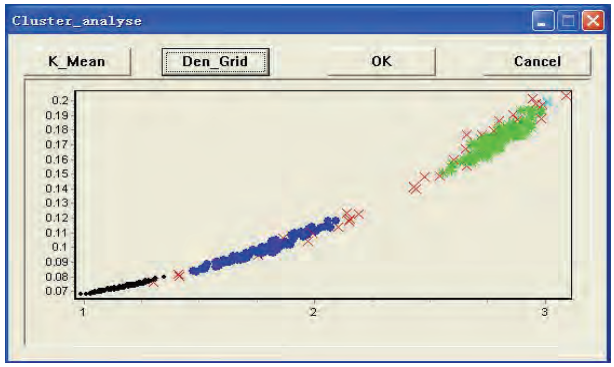
T_Analyse's part of the interface is shown in Figure 8. Function: It displays the time-domain waveforms and probability density histogram after that the imported data have been amplified.

5) The function and operation of F_Analyse

F_Analyse's part of the interface is shown in Figure 9. Function: To display amplitude-frequency diagram of the imported data of gear vibration signal (abscissa is the frequency, unit is Hz, ordinate is the amplitude of vibration acceleration), Power spectrum diagram (abscissa is the frequency, unit is Hz,

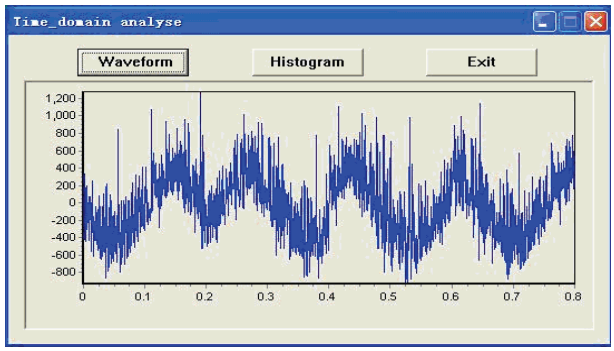


(a) The analysis results by K_Mean method

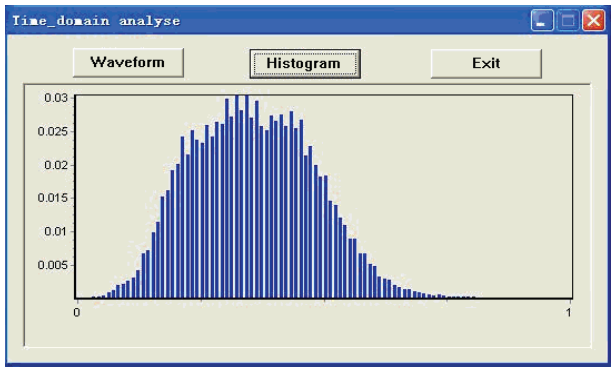


(b) The analysis results by Den_Grid method

Figure 7. The display interface of Cluster



(a) Time-domain waveform



(b) Probability density histogram

Figure 8. The display interface of T_Analyse

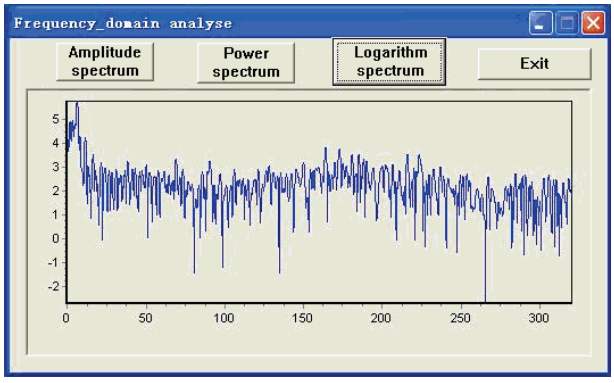


Figure 9. The display interface of F_Analyse

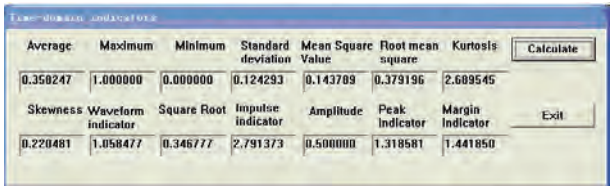


Figure 10. The display interface of Indicators

ordinate is the power of vibration acceleration), logarithm spectrum diagram (abscissa is the frequency, unit is Hz, ordinate is the amplitude, converted into logarithm, of vibration acceleration).

6) The function and operation of Indicators

Indicators' part of the interface is shown in Figure 10. Function: It calculates and shows the time-domain parameters of the various indicators.

4.4 Fault Diagnosis and Identification

Signal diagnosis includes Diagnose 1, Diagnose 2, and Diagnose 3. Functions are as follow:

The interface of Diagnosis 1 is shown in Figure 11. Functions: Neural Network Algorithm is used to train and identify the imported data, finally, to get the fault type of the imported data. Producing the data sample, training the data sample, introducing the trained weight, diagnosing and getting test result, it can identify three kinds of faults: the crack at the gear root, the crack at the gear's reference circle and the wear abrasion on tooth surface.

After the Diagnose 1 button is clicked, a new window (Figure 11(a)) appeared. In the new window, it should be set the corresponding parameters. StartPT is the starting position of the interception segment, SampLen is the data length of each piece, SampNum is the number of data segment. Select the training sample fault type in the drop-down menu including three kinds: crack at tooth root, crack at teeth top and tooth wear abrasion. Select New if create a new sample file, select Old if add samples into old sample files. BlockNum and L are same to the part above; P is time-serial_model order; Variance is used to direct time-serial analysis error. After the setting is finished, click the MakeSamps button, a save location window of the created samples pop-up to select store directory in it, and then click OK in the following window. The creation of sample is complete after clicking OK button in the prompt window. Then, click Train button, select the created sample to train. After training is completed, click GetWeight button, select the trained weight, and finally click the Diagnose button to get the diagnostic results. As shown in Figure 11(b).

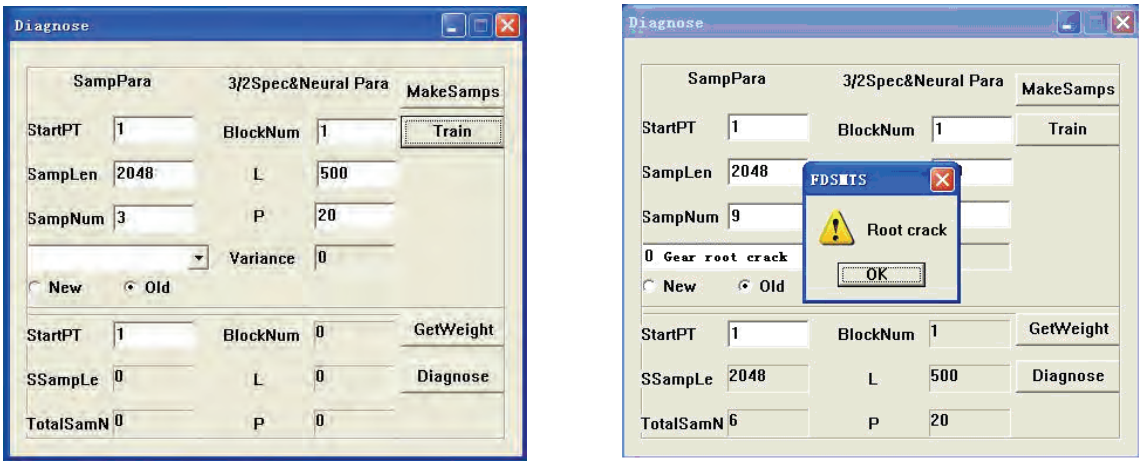
The interface of Diagnose 2 is shown in Figure 12. Function: The faults can be recognized by the spectral entropy algorithm. By importing vibration signal at different running condition, its time-frequency characteristics are analyzed, and the spectral entropy of the signal are calculated. Because of the cluster analysis, the fault of the gear can be diagnosed. As shown in Figure 12, Speed is used to set the speed of running system, Cycle-Num is used to set the tested sample number, Entropy is used to display spectral entropy value of the tested sample, Fault-style is used to output diagnostic results. After clicking Diagnose 2 button, setting every parameter in the prompt window and clicking OK button, the diagnosis can be done.

The interface of Diagnose 3 is shown in Figure 13. Function: Through analysis, according to the rotational speed and the measured sample number, the proposed system can be effectively used in diagnosis of different faults of gear system, such as the crack, and the wear abrasion fault. As shown in Figure 13, clicking Diagnose 3 button and setting the speed and number of the tested samples in the prompt dialog box, then clicking the fault identification button, we can directly get the tested samples fault type. It can process, diagnose and identify three kinds of data measured from experiment: normal gear, crack gear and wear gear.

Via above analysis and test diagnosis, it shows the accuracy and feasibility of the fault analysis, detection and diagnosis system of the gear transmission system. At the same time, it also shows that the virtual instrument diagnostic system possesses the basic functions of gear transmission vibration signal processing, such as, analysis, display, identification, diagnosis and so on, and the results are very good.

5. Summaries

This paper discusses the requirements, functions and development of gear failure analysis software system. Via VC++ and MATLAB mixed programmer, virtual instrument Diagnosis System of Gear Fault Analysis, which can run under Windows platform, is successfully developed; by importing the vibration signal data measured at practical test, it demonstrates the functions of



(a) The main interface of Diagnose 1

(b) The diagnostic result of using neural network

Figure 11. The display interface of Diagnose 1

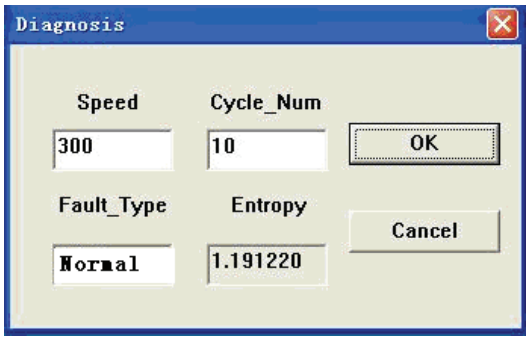


Figure 12. The interface of Diagnose 2

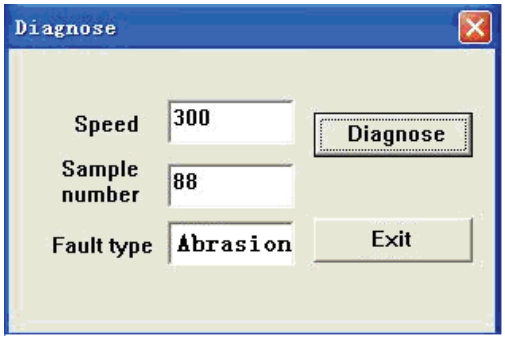


Figure 13. The interface of Diagnose 3

the virtual instrument system in analysis, processing, diagnosis and identification of gear vibration signal. It validates the correctness and effectiveness of this system in identification and diagnosis of gear vibration signal, and gets the relative graphics and calculation results. It also shows that this software system can be used for analysis, processing, diagnosis and identification of vibration signal of gear system, and the proposed method can be effectively used in engineering diagnosis of gear different faults.

6. Acknowledgments

This research is sponsored by the Natural Science Fund of China (NSFC) (Grant No. 50575187) and the fund of Chinese aviation science (01I53073) and the fund of Natural science of ShanXi province in China (2004E219).

REFERENCES

[1] R. P. Shao, H. Y. Liu, and Y. Q. Xu, "Fault detection and diagnosis of gear system based on higher order cumulants," Chinese Journal of Mechanical Engineering, Vol. 44, No. 6, pp. 161–168, 2008.

[2] J. Rafiee, F. Arvani, A. Harifi, and M. H. Sadeghi, "Intelligent condition monitoring of a gearbox using artificial neural network," Mechanical Systems and Signal Processing, Vol. 21, No. 4, pp. 1746–1754, 2007.

[3] S. T. Wan and H. X. Tong, "Design of the fault diagnosis system of generator rotor winding inter-turn short circuit based on virtual instrument," The Eighth International Conference on Electronic Measurement and Instruments, Vol. 3, pp. 576–580, 2007.

[4] S. J. Lv, M. H. Zhang, Y.F. Li, and X. J. Yu, "Design on the fault diagnostic system based on virtual instrument technique," Second International Workshop on Knowledge Discovery and Data Mining, pp. 304–307, 2009.

[5] Y. F. Liu, D. Miao, Y. H. Peng, and F. Meng, "Remote fault diagnosis based on virtual instrument technology," Proceedings of the 10th International Conference on Computer Supported Cooperative Work in Design, 2006.

[6] Z. H. Xu, X. B. Wu, and Y. Guo, "Fault testing and diagnosis system of armored vehicle based on information fusion technology," The Eighth International Conference on Electronic Measurement and Instruments, Vol. 3, pp. 708–711, 2007.

[7] C. T. Wang and Robert X. Gao, "A virtual instrumenta-

tion system for integrated bearing condition monitoring," IEEE Transactions on Instrumentation and Measurement, Vol. 49, No. 2, pp. 325–332, April 2000.

[8] Y. F. Li, "Research on fault diagnosis of FPC based on virtual instrument and rough set theory," Proceedings of the 6th World Congress on Intelligent Control and Automation, Dalian, China, pp. 3891–3895, June 21–23, 2006.

[9] Giovanni Betta, Andrea Bernieri, Domenico Capriglione, and Mario Molinara, "SVM-based approach for instrument fault accommodation in automotive systems," IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems (VECIMS), Giardini Naxos, Italy, pp. 145–150, July 18–20, 2005.

[10] M. H. Zhang, Y. Wang, S. Zhu, S. J. Chen, and J. Y. Bao, "Research on application of virtual instrument and grey theory in the fault diagnostic system," Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, Nanjing, China, pp. 1343–1346, November 18–20, 2007.

[11] Y. Z. Yu, S. Y. Yang, and J. Q. Yao, "VC++ and MATLAB mixed programming in DOAS air pollution monitoring system," Journal of Tianjin University Science and Technology, Vol. 36, pp. 548–552, 2003.

[12] X. X. Xu, D. C. Liu, and Y. Huang, "Power system fault reoccurrence and analysis system based on hybrid programming of VC++ and MATLAB," Electric Power Automation Equipment, Vol. 26, pp. 38–40+44, 2006.

[13] B. Chen, S. B. Chen, and T. Lin, "Displaying weld pool's 3-D shape by mixed programming between VC++ and MATLAB," Material Science and Technology, Vol. 14, pp. 105–108, 2006.

[14] Markus Timusk, Mike Lipsett, and Chris K. Mechefske, "Fault detection using transient machine signals," Mechanical Systems and Signal Processing, Vol. 22, No. 7, pp. 1724–1749, 2008.

[15] R. P. Shao, X. N. Huang, and J. H. Hu, "Analysis of data mining of clustering and its application to mechanical transmission fault diagnosis," Journal of Aerospace Power, Vol. 23, No. 10, pp. 1933–1938, 2008.

[16] H. Y. Liu, "Research of mechanical transmission fault diagnosis system based on HOC virtual instrument," Northwestern Polytechanical University, 2006.

[17] Y. He and Z. F. Li, "The research and application of intelligent fault diagnosis methods," Journal of Zhejiang University (Agric. & Life Sci.), Vol. 29, No. 2, pp. 119–124, 2003.

[18] X. Chen and W. X. Li, "Fault detection & diagnosis system for rolling bearings based on virtual instrument," Journal of WUT (Information & Management Engineering), Vol. 19, No. 1, pp. 41–44, 2007.

[19] Y. Lv, Y. R. Li, and Z. G. Wang, "Virtual instrument and its application to machinery fault diagnosis," Journal of Wuhan University of Science and Technology (Natural Science Edition), Vol. 25, No. 2, pp. 135–137, 2002.

[20] M. H. Zhang, Y. Wang, S. Zhu, *et al.*, "Research on application of virtual instrument and grey theory in the fault diagnostic system," Proceedings of 2007 IEEE International Conference on Grey Systems and Intelligent Services, pp. 1343–1346, November 18–20, 2007.

[21] B. P. Tang, F. B. Cheng, and A. J. Yin, "Research on virtual instrument for wavelet transform," IEEE Workshop on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, pp. 649–652, September 5–7, 2005.

J. Software Engineering & Applications, 2010, 3: 185-189
doi:10.4236/jsea.2010.32023 Published Online February 2010 (http://www.SciRP.org/journal/jsea)

Research on LFS Algorithm in Software Network

Wei Wang, Hai Zhao, Hui Li, Jun Zhang, Peng Li, Zheng Liu, Naiming Guo, Jian Zhu, Bo Li, Shuang Yu, Hong Liu, Kunzhan Yang

Information Science and Engineering Northeastern University, Shenyang, China.
Email: wangweiwin1@163.com, zhhai@neuera.com

Received October 11th, 2009; revised October 26th, 2009; accepted November 5th, 2009.

ABSTRACT

Betweenness centrality helps researcher to master the changes of the system from the overall perspective in software network. The existing betweenness centrality algorithm has high time complexity but low accuracy. Therefore, Layer First Searching (*LFS*) *algorithm is proposed that is low in time complexity and high in accuracy. LFS algorithm searches the nodes with the shortest to the designated node, then travels all paths and calculates the nodes on the paths, at last get the times of each node being traveled which is betweenness centrality. The time complexity of LFS algorithm is O*(*V*2).

Keywords*: *LFS, Software Network, the Shortest Path, Betweenness Centrality

1. Introduction

It is over fifteen years since Norman Fenton outlined the need for a scientific basis of software measurement. Such a theory is a prerequisite for any useful quantitative approach to software engineering, although little attention has been received from both practitioners and researchers. Measurement is the process that assigns numbers or symbols to attributes of real-world entities. Unfortunately, many empirical studies of software measurements lack a forecast system that combines measurements and parameters in order to make quantitative predictions. How can we overcome these limitations?

Here we present a new approach to software engineering based on recent advances in complex networks. As a typical parameter and an important global statistics, betweenness centrality can meet the needs of researchers to know the inter-reaction of software network from the overall perspective. The definition of betweenness centrality is the times of a node being traveled in all the shortest paths of the software network and it reflects the influence of the node in the whole network.

The existing betweenness centrality algorithms are either based on the shortest path or merely approximate algorithm [1–3], and both have its own shortcomings. First, the traditional shortest path algorithm has high time complexity and costs too much time to be available in large network, which makes it impossible to put the algorithms into practice. The research on approximation of betweenness centrality is unavailable for low accuracy which brings great obstacle for further study [4,5].

As mentioned in [6], the average path length is similar to normal distribution with the mean of 15.21. LFS algorithm is proposed based on this method.

The dissertation contains three parts: First, take Dijkstra algorithm as example to conclude shortcomings of traditional algorithms. Then, propose LFS algorithm. At last, compare these two algorithms to show the advantages of LFS algorithm.

2. Traditional Dijkstra Algorithm

The complexity of betweenness centrality comes from calculating the shortest path between each two nodes in network, and the time complexity of Dijkstra is O(n3). The existing algorithms based on the shortest path contain Dijkstra, Floyd-Warshall, and Johnson. Dijkstra is the most popular and classic algorithm.

The idea of Dijkstra is like this. Abstract the network into a graph, Put the isolated nodes (the nodes with out-degree and in-degree being both 0) in set V and the nodes having been traveled in set S, then calculate the shortest path from vi to any node in the graph. Move the node vk with the shortest path from V-S to S till V-S is empty.

① Initialization: Set up a two-dimensional array a to mark whether the shortest path has been found out between the two nodes. Set up a none-dimensional array to store the betweenness centrality. Set up a adjacency matrix arcs with 1 if there exist edge between the nodes, else with ∞. V is a set of all nodes and S is a set of all marked nodes. The value from vi to vj is initializes as

follows:

D[j]=arcs[vi][j] vj∈V

Then put vi into S.

② Pick up vk which satisfies

D[k]=min{D[j] | vj∈V-S}

vk is the end point of the shortest path starting from vi. Put vk into S.

③ Calculate the length of the shortest path from vi to each accessible node in set V-S

D[k]+arcs[k][m]<D[m]

and revalue the D[m] as

D[k]+arcs[k][m]=D[m]

④ Add 1 to betweenness centrality of nodes if only they are on the shortest path from vi to any node in the graph. Then mark these nodes being travelled in the two dimensional array a. Repeat ②, ③ n-1 times. At last, it gets all the shortest paths from vi to the other nodes in the graph.

⑤ It is the end.

Repeat the process for n times and get the shortest path between each two nodes. Since each time contains a two-cycle, the time complexity of Dijkstra is O(V3). The space complexity is T(V2).

3. LFS Algorithm

According to the description to Dijkstra, it has a three-cycle and find out the shortest path between each two nodes which, then add 1 to between centrality of the nodes being on the path. The application range is limited for its high time complexity and the time consumption is unavailable when it is applied into large network of thousands of nodes. To solve this problem, the dissertation proposed a new algorithm (LFS) with the help of the features of software network.

As a number of papers mentioned [7–9], the length of shortest path between each two nodes wouldn't exceed a constant. For example, [1] says the average of the length is 15.21. Based on this method, LFS is proposed. Starting from a node in the network, put the connected nodes with the shortest path of 1 into array, and then put that of 2 into the array and so on till that of n in the array but there's no connected node with shortest path of n+1. Add 1 to the nodes on the paths which just have been found. The time complexity of LFS is the summation of length of all the shortest path between each two nodes, that is O(V2). Compared with Dijkstra, the time complexity of LFS is reduced obviously.

The preparation of LFS is similar to Dijkstra: abstract the network into a graph, set up an adjacent list which makes single-link lists for all non-isolated nodes in the graph and the i-list contains the nodes directly connected to the non-isolated node vi. Each node is composed by two parts: neighboring nodes field (adjvex) and linking field (nextarc). The neighboring nodes field marks where the nodes connected to node vi are in the graph and the linking field marks the next node. Each single-link has a head node which is composed of data field (data) and linking field (firstarc). Data field marks the number of vi in the graph and liking filed marks the first node that is connected to vi.

①Initialization: Set up a two-dimensional array c initialized maximum to store the length of the shortest path between each two nodes in the graph, then set up a one-dimensional array bc initialized 0 to store the betweenness centrality of each node. V is the number of non-isolated nodes in the graph.

② Set up array a and b. a is used to restore the end node of the shortest path. b is used to restore the number of nodes with the same starting node and the same length and put the starting point into a and put la=0 into b. At last, set the relative element 0 in array c.

③Judge whether it is the end of array a. If it is, turn to ⑧; else turn to④.

④Check out the number of nodes with the length of shortest path la, and set it to be n.

⑤Travel the next node in array a and the find out all child nodes which are connected to this node (parent node) directly. If the value from starting node to this node in array c is bigger than la, set the value from the starting node to this node and the value from this node to starting node in array c to be la+1, then put the id of parent node and the id of child node into a. The same nodes being put into an m times means that there are m shortest paths from the starting node to this node. Add 1 to num which is the number of nodes on layer la+1. Because the id of parent node can be found by the id of child node, the shortest paths from the starting node to the other nodes in array a can been found, then add 1 to the nodes on each shortest path.

⑥repeat ⑤ n-1 times.

⑦put num into array b, la=la+1, turn to ③.

⑧repeat ②-⑦ V-1 times.

⑨It is the end.

According to the description above, the time complexity of LFS is the summation of all the shortest path in the network, that is O(V2). And V is the number of non-isolated nodes in the network. The space complexity is T(V2) which is equal to that of Dijkstra.

The comparison between the time complexity and space complexity of Dijkstra and LFS:

Table 1. Time complexity and space complexity of Dijkstra and LFS

algorithm	time complexity	space complexity
Dijkstra	O（V3）	T(V2)
LFS	O（V2）	T(V2)

4. Performance Evaluations

Dijkstra takes breadth-first method to travels all the nodes in the software network, find out all the shortest paths, obtain the nodes on the shortest path, [10–12] and calculate betweenness centrality. Dijkstra has a three-cycle which makes the time complexity is so high that it is a fatal shortcoming when applied into large-scale software network.

LFS only has one-cycle, and travels nodes in the network one by one, then find out the shortest path from starting node to the other nodes. The time complexity of LFS is O(V2) and the space complexity is T(V2). Depending on what mentioned above, LFS has great advantages both in time complexity and in space complexity.

With the development of computer science, the computer memory becomes bigger and bigger which can satisfy all kinds of demands and no longer need to be considered. So we spend no more time on discussing space complexity and merely pay attention to time complexity.

We get satisfying result with the help of HP computer which is composed of Core Duo 6300 CPU, 1.86GHz Frequency, DDR2 667 1GB Memory and Windows XP SP2 Operation System.

In order to verify that LFS has great advantages in time complexity, 22 samples are selected and sorted ascending which can justify whether LFS is applicable to software of different sizes. The comparison of time consumption between Dijkstra and LFS is shown as follows:

DT is the time cost by calculating betweenness centrality of the software by Dijkstra, and WT is that by LFS. Time units are seconds. DT/WT which marks advantages of LFS in time consumption is the ratio of DT and WT.

Since both Dijkstra and LFS are related to the number of non-isolated nodes in the software network, the relation between these two kinds of algorithm is shown through the number of non-isolated nodes by the following figures. Figure 1 shows the time cost by these two algorithm when measure the same software. Figure 2 shows the multiples that LFS saved when compared to Dijkstra.

As shown in Figure 1, when the number of non-isolated nodes goes up, the time consumption of Dijkstra appears a clear upward trend, while the time consumption of LFS changes smoothly. And the larger the number of non-isolated nodes the more obviously the difference appears, which approves that the times consumption would not increase as quickly as the number of nodes, but Dijkstra is on the contrary.

As shown in Figure 2, Dijkstra cost more than 10 times even tens of times of time than LFS when they are applied into the same software. With the growth of the number of nodes, the curve has a clear upward trend although it is fluctuant sometimes. It approves that the advantages of LFS in time complexity appears more and more obviously as the software becomes larger and larger under the same circumstance, which says that LFS is especially fit for large-scale software network.

Table 2. The comparison of time consumption between Dijkstra and LFS

software	number of nodes	number of edges	number of non-isolated nodes	DT（s）	WT（s）	DT/WT
waimea	116	193	86	0.359	0.032	11.22
kicad	212	300	180	0.609	0.110	5.54
ktorrent	263	335	217	3.313	0.250	13.25
rhythmbox	366	342	252	8.349	0.531	15.72
filezilla	431	577	358	5.500	0.563	9.77
licq	574	633	491	12.110	1.282	9.45
freemind	713	933	562	53.172	1.812	29.34
espgs	1339	1271	955	150.094	7.063	21.25
abiword	1300	2124	1167	384.391	11.531	33.34
ArgoUML	2031	2217	1731	747.093	31.718	23.55
kdegraphics	2014	3498	1749	1036.781	44.688	23.20
mysql-5.0.56	3132	3837	2182	1685.828	54.453	30.96
mysql_6.0.6	3793	5368	2889	3131.318	104.531	29.96
kdepim	3518	4447	3008	2933.594	136.047	21.56
koffice	4580	5892	3883	4853.296	185.891	26.11
linux	7343	6045	4238	6756.612	296.313	22.80
resin	5076	7875	4261	10613.281	389.072	27.28
node	5418	11451	5418	13693.187	553.985	24.72
firefox	9261	15533	5781	18167.438	725.530	25.04
firefox	7100	48236	7100	24267.391	942.793	25.74
Mozilla	8354	13878	7195	37298.863	1315.750	28.35
firefox	10115	17469	8830	120818.089	3152.580	38.32

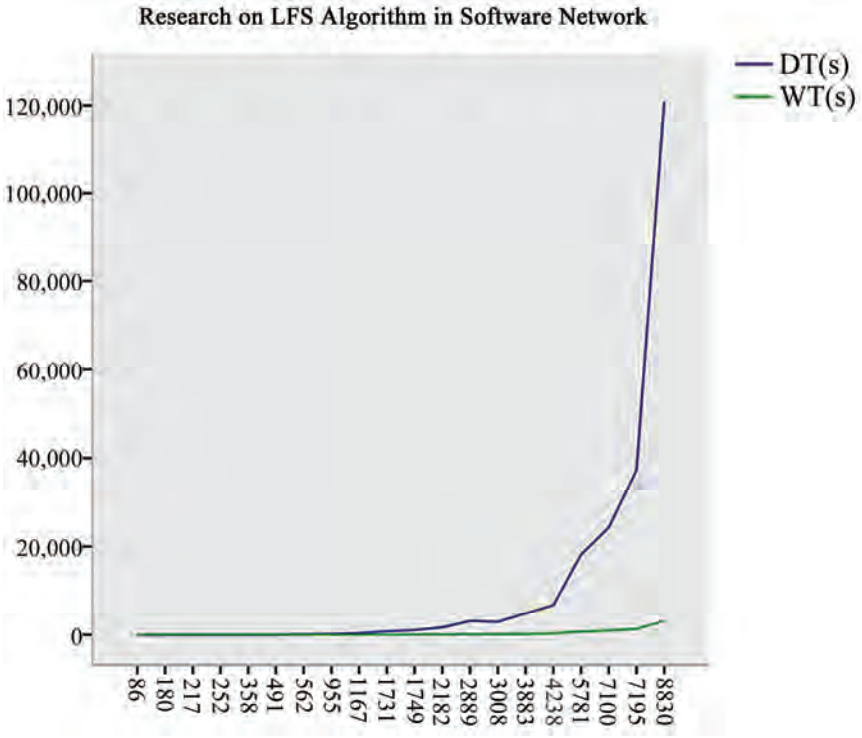


Figure 1. The time cost by Dijkstra and LFS

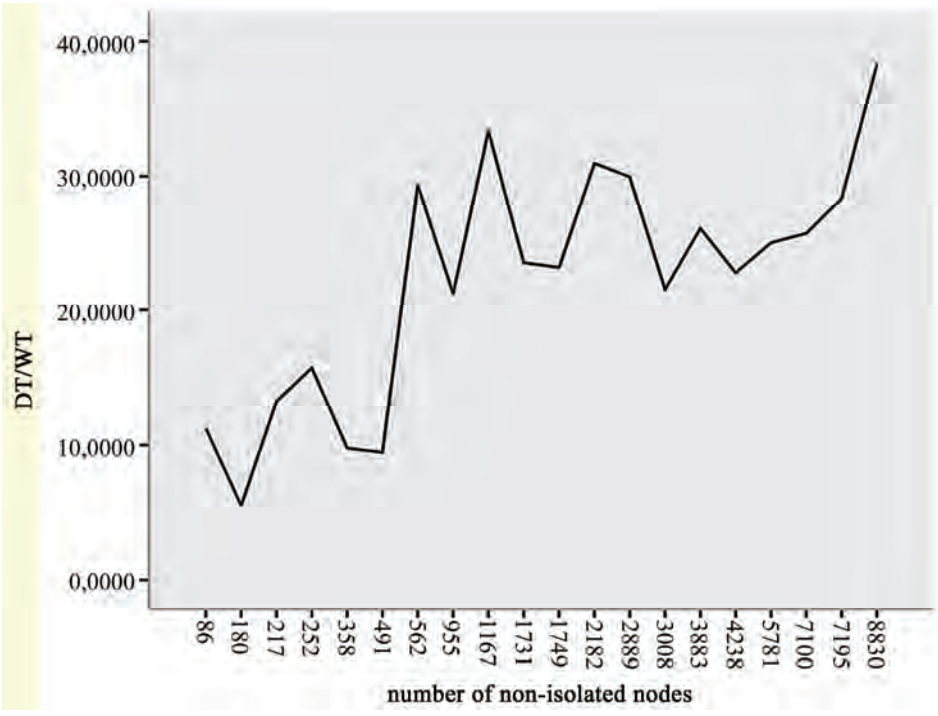


Figure 2. The multiples that LFS saved when compared to Dijkstra

As shown in Figure 2, the value of DT/WT shows an upward trend in volatility. Due to the relationship between the time complexity of LFS algorithm and the complexity of the network itself, the value of DT/WT depends on the size of the network which is equals to the sum of the length of the entire shortest path. When

the software network is a little more complex and the lengths of shortest paths between some nodes are relative longer, LFS algorithm's time complexity will increase. For the same reason, when a more uniform distribution of the network and the lengths of shortest paths between some nodes are relative shorter, the advantage of LFS is fully reflected that the time complexity of LFS drops to relative small and the time consumption comes down too, but Dijkstra doesn't have such feature. Therefore, the value of DT/WT shows an upward trend in volatility. However, the curve appears an upward trend from the whole.

As mentioned above, the low time complexity of LFS can meet the needs of starting research on large-scale software and solve the problems brought by time complexity in the software measurement. At the same time, the accurate data obtained will bring accurate and reliable conclusion in practical research work.

5 Conclusions

LFS can solve a series of problems brought by the traditional algorithm. It has advantages both in time complexity and accuracy which are so important in practical research work that may result in disaster conclusion without it. LFS improve the efficiency and the accuracy to calculate the betweenness centrality, which ensures the further research to be continued smoothly.

REFERENCES

[1] M. Pióro, A. Szentesi, J. Harmatos, A. Juttner, P. Gajowniczek, and S. Kozdrowski, "On open shortest path first related network optimization problems," Performance Evaluation, Vol. 48, pp. 201–223, May 2002.

[2] E. P. F. Chan and N. Zhang, "Finding shortest paths in large network systems," Proceedings of the 9th ACM International Workshop on Advances in Geographic Information Systems (ACMGIS2001), Atlanta, Georgia, pp. 160–166, 2001.

[3] D. Awduchen, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao, "Overview and principles of internet traffic engineering," RFC 3272, May 2002.

[4] D. Torrieri, "Algorithms for finding an optimal set of short disjoint paths in a communication network," Communications, IEEE Transactions, Vol. 40, No. 11, pp. 1698–1702, 1992.

[5] B. Fortz and M. Thorup, "Internet traffic engineering by optimizing OSPF weights," in Proceedings IEEE INFOCOM, pp. 519–528, 2000.

[6] X. Zhang, H. Zhao, W. B. Zhang, and C. Li, "Research on CFR algorithm for Internet," Journal on Communications, Vol. 27, No. 9, September 2006.

[7] B. Fortz and M. Thorup, "Optimizing OSPF/IS-IS weights in a changing world," IEEE Journal on Selected Areas in Communications, Vol. 20, No. 5, pp. 756–767, May 2002.

[8] G. Rétvári and T. Cinkler, "Practical OSPF traffic engineering," IEEE Communications Letters, Vol. 8, No. 11, pp. 689–691, November 2004.

[9] A. R. Soltani, H. Tawfik, J. Y .Goulermas, *et al.*, "Path planning in construction sites: Performance evaluation of the dijkstra, a* and GA search algorithms," Advanced Engineering Informatics, Vol. 16, No. 4, pp. 291–303, 2002.

[10] Z. Wang, "Internet QoS: Architectures and mechanisms for quality of service," Academic Press, CA, San Diego, 2001.

[11] M. Pióro and D. Medhi, "Routing, flow, and capacity design in communication and computer networks," Morgan Kaufmann, CA, San Diego, November 2004.

[12] W. Ben-Ameur and E. Gourdin, "Internet routing and related topology issues," SIAM Journal on Discrete Mathematics, Vol. 17, No. 1, pp. 18–49, 2003.

[13] Gamma Erich, Helm Richard, Johnson Ralph, and Vlissides John, "Design patterns: Elements of reusable object-oriented software," Addison-Wesley Longman Publishing Co., Inc. Boston, USA, 1995.

[14] M. M. Lehma and J. F. Rmail, "Software evolution and software evolution processes," Annals of Software Engineering, Vol. 14, No. 1, pp. 275–309, 2002.

[15] B. Dougherty, J. White, C. Thompson, and D. C. Schmidt, "Automating hardware and software evolution analysis," Engineering of Computer Based Systems (ECBS), 16th Annual IEEE International Conference and Workshop on the[C], pp. 265–274, 2009.

[16] S. N. Dorogovtsev and J. F. Mendes, "Scaling properties of scale-free evolving networks: Continuous approach," Physical Review E, Vol. 63, No. 5, pp. 56125, 2001.

[17] N. Zhao, T. Li, L. L. Yang, Y. Yu, F. Dai, and W. Zhang, "The resource optimization of software evolution processes," Advanced Computer Control International Conference on [C] (ICACC), pp. 332–336, 2009.

[18] B. Behm, "Some future trends and implications for systems and software engineering processes," Systems Engineering, Vol. 9, No. 1, pp. 1–19, 2006.

[19] Lollini Paolo, Bondavalli Andrea, and Di Giandomenico Felicita, "A decomposition-based modeling framework for complex systems," IEEE Transaction on Reliability, Vol. 58, No. 1, pp. 20–33, 2009.

[20] Y. Ma and K. He, "A complexity metrics set for large-scale object-oriented software systems," in Proceedings of 6th International Conference on Computer and Information Technology, pp. 189–189, 2006.

[21] K. Madhavi and A. A. Rao, "A framework for visualizing model-driven software evolution," IEEE International Advance Computing Conference (IACC), pp. 1628–1633, 2009.

[22] S. Valverde and R. V. Sole, "Network notifs in computational graphs: A case study in software architecture," Physical Review E, Vol. 72, No. 2, pp. 26107, 2005.

