# Application of Listed Data Structures in Knitting Industry CAD/CAM Systems' Development

Elena I. Zaharieva-Stoyanova[1]

*Abstract* – **This paper treats the problem related to data representation in knitting industry CAD systems. Listed data structures are successfully used to code graphic data related to knitting pattern object. Knitting patterns are represented as sets of listed dots, so it is the reason to use listed data structure to describe knitting pattern object graphically. The goal of the paper is to give an approach of usage of listed structures in CAD systems' data representation. In this particular system the structures linked list is used. The data type does not allow using listed structures in MS VC++ collections. That is the reason to create listed structures related to knitting pattern graphic objects. The algorithms operated with list structures are suggested. These algorithms are applied to particular software module Construction. The application can be embedded in whole CAD/CAM system in knitting industry or it can be used as software application creating patterns of knitting products. The MS VC++ and MFC are used to create a software application. The main reason to create such project is that it can be used as software application from fashion designers to make boutique collections of knitting products. The project can be used as an example for CAD system development.**

*Keywords* – **CAD/CAM systems, computer graphics, knitting industry, listed data structures, linked lists.**

## I. INTRODUCTION

CAD/CAM/CAI systems are a part of a huge number of software applications launching to the market. Their development is related to software technologies at all. It means that introduction of new programming environments, software techniques, and new Operation Systems (OS) leads to creation of new CAD/CAM software application and to introduction of new versions of existing ones. One of the most important problems solving in CAD/CAM systems' development refer to data representation [1], [3]. Because this paper treats the problem of data representation in CAD systems in knitting industry, the data type will be discussed at first.

Generally, application of computer graphics in knitting industry CAD/CAM systems has two aspects: knitting structures design and design of knitting pattern shapes. The first one refers to design of knitting structures – intarsia, jacquard, lace, arans. The second one is related to knitting machine's capacity to make products by fully-fashion (FF) method. It means that the machine knits the parts (*cuts*) of

[1]Elena I. Zaharieva-Stoyanova is with the Faculty of Electronic Engineering, Technical University of Gabrovo, 4 H. Dimitar str. , 5300 Gabrovo, Bulgaria, E-mail: zaharieva@tugab.bg.

products, or the whole products. This method allows to avoid cutting the materials, reduces the number of the operations and it leads to reducing the waste products to a minimum.

There is a big difference between data representing knitting patterns objects and data representing knitting structures. While the knitting patterns as graphic objects described by regions or polygons of straight lines and curves – vector graphics, the knitting structures are represented by means of raster graphics – rectangular grid of colored pixels or icons (small bit-map images). The difference between two types graphics objects demands the different approach to the design process and to the data representation, as well. To create a knitting structure a designer needs an editor for raster graphics images. Each item of the raster corresponds to the knitting stitch. A matrix form of an image data representation is suitable [8], [9], [10], [11], [12],[13].

As graphics object the knitting patterns are described as polygons of straight lines and curves. They are determined by set of *control points* (*knots*). Using these knots the computer can generate the whole image [4], [5], [6].

This paper treats the problem related to data representation in knitting industry CAD systems. Listed data structures are successfully used to code graphic data related to knitting pattern object. Knitting patterns are represented as sets of listed dots, so it is the reason to use listed data structure to describe graphically knitting pattern object. The goal of the paper is to give an approach of usage of listed structures in CAD systems' data representation. In this particular system the structures linked list and doubly linked circuit list are proper to use. The data type does not allow using listed structures in MS VC++ collections. That is the reason to create listed structures related to knitting pattern graphic objects. The algorithms operated with list structures are suggested.

## II. FEATURES OF KNITTING PRODUCTS

The goal of this paper is to suggest an approach to CAD development using FF knitting method. The core of such system is editing of constructions and the shape of narrow sections of models, there by. The pattern editor, create different knitting structures would be programmed as additional module incorporated to the system [6], [7], [8].

This paper treats the problems related to CAD system development with possibilities to create models of knitting products; their constructions and variety of sizes. The project suggests an approach to data representation. Knitting structures and patterns design module will be a point of further development.
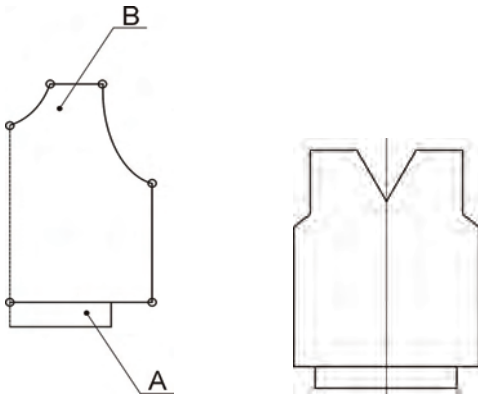
Fig. 1 (a,b). Examples of narrow sections

To choose a method for data representation and pertaining functions, the programmer needs to know the type and the characteristics of designed objects. Figure 1 (a,b) presents examples of knitting products narrow sections. The main difference between them is that figure 1a presents a shape of half symmetric part of the narrow section (cut) while figure 1b shows the whole cut. Section A is a margin – beginning part of the knitted detail. The basic knitted cuts have a margin, so it is a good idea to add it as an extra element. Moreover, the characteristics of this item have more technological aspect. It is obvious that the cuts can be described as a polygon formed by straight lines and curves (fig. 1). It can also be interpreted as a polyline but in general, lines formed knitting cut shape are straight lines and curves, as well [3], [4], [5].

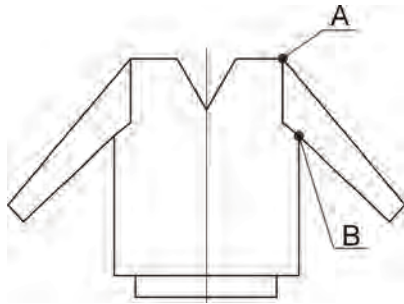There are two manners to represent a cut of knitting product



Fig. 2. Examples of a knitted product construction

(detail): as a set of lines or as a set of points. The advantage of the first one is that it is easy to construct a shape - line by line. However, it is necessary to check if the last point of previous line is equal to the first one of the next line. There is a redundancy of information. If the detail is represented as points, the redundancy will be avoided, but it makes some difficulties in line drawing and detail conversion into knitting rows and stitches. Knit products could be regarded as a set of details. Fig.2 shows an example of knitting products construction. A and B are points belonging to two details and they determine how to join the narrow sections. Such points

fix a connection between details and determine that the distances between them for both details should be equal.

## III. LISTS AS DYNAMIC DATA STRUCTURES

### A. Features of lists as data structures

Lists are very flexible dynamic data structures. Unlike arrays items may be added to them or delete from them at will. It means that an item can be added to the list or removed from the list easy nevertheless of its allocation [2],[7].

Another advantage of the lists as data structures is that a programmer need not worry about how many items a program will have to accommodate because these data structures are dynamic. The memory is allocated when a new item is created. Thus the number of items that may be added to a list is limited only by the amount of memory available. If an item must be removed from the list, its memory allocation is released. This allows us to write robust programs which require much less maintenance. In comparison with array lists are more flexible data structures because the programmer can allocate or release a memory for each list item independently but handling for the arrays is much easier than list handling.

In a linked list, each item is allocated space as it is added to the list. A link is kept with each item to the next item in the list. That way it is accepted for item of the list to be called node. Each node of the list has two elements - the item being stored in the list and a pointer to the next node in the list. The last node in the list contains a NULL pointer to indicate that it is the end or *tail* of the list. This is presented on fig. 3.
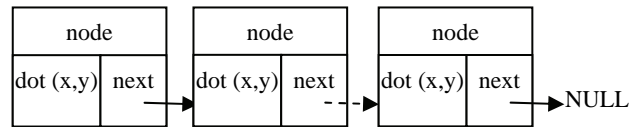


Fig. 3. Data structure **"linked list"**

This paper suggests an approach for particular lists structures application for descriptions of graphics objects. That way go on to describe lists structures instead common term "item", the particular object part "dot" will be used. In this particular case dots are items of the lists.

There are some variations of the lists structures. Such variants are:
- Circuit Linked Lists
- Doubly Linked Lists
- Doubly Linked Circuit Lists

Circularly or Circuit Linked List is that the tail of the list is always pointing to the head – it means that the pointer of the last node points to the first node, which is the beginning of the list. Doubly linked lists have a pointer to the preceding item as well as one to the next. They permit scanning or searching of the list in both directions. To go backwards in a simple list, it is necessary to go back to the start and scan forwards. Doubly linked circuit list is combinations between doubly linked list

and circuit linked list also have a pointer to the preceding item as well as one to the next.

## B. Handle for the lists

The variable (or *handle*) which represents the list is simply a pointer to the node at the *head* (beginning) of the list. Practically, it points to the first node of the list.

The most important operations, applied to the lists structures are: addition to the tail of the list, insert a new item to the list, and deleting an item from the list. The operation *"addition"* is treated as a manipulation to add a new item (node) to the tail of the list, after the last node. The term *"insert"* is used for adding a new item (node) regardless of its new positions. It means to "insert" node, exactly.

Figure 4 illustrates the operation *"insert a new node"* to the list. There are two variants of the operation "Insert before" and "Insert after" according to if the new item is inserted before the selected item or after it.
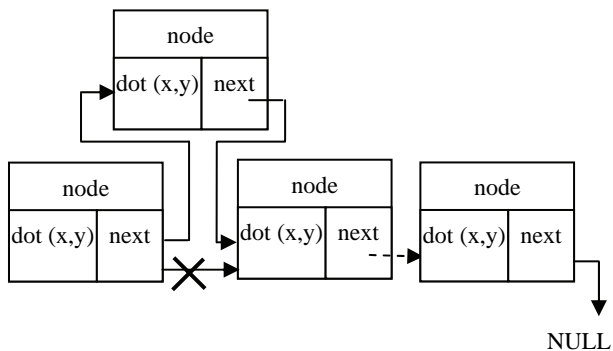


Fig. 4. The operation *"insert a node to the list"*

The simplest strategy (algorithm) for inserting an item to a list is to: allocate space for a new node; copy the item into it; make the new node's **next** pointer point to the next item of the list and make the current node's next point to the newly allocated node.

A delete operation is another important list handler. The simplest strategy (algorithm) to remove a node from the list is to make the previous node's next point to next node (item pointed by the next pointer of the node for deleting) and after that remove allocated memory for the node. This operation is illustrated on fig. 5.
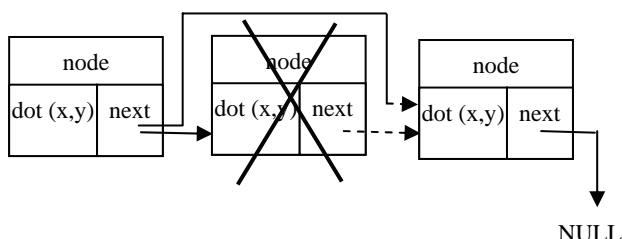


Fig. 5. The operation *"delete a node from the list"*.

## IV. APPLICATIONS OF LISTED DATA STRUCTURES IN KNITTING INDUSTRY CAD SYSTEMS

There are some possibilities to describe a shape of the detail. The most popular are: arrays and listed structures. The former is the easier way for programming whereas the latter leads to better code efficiency and data representation. The data structure can be a linked list or a circuit doubly linked list. Because a detail described as straight lines and curves have to be converted into knitting rows and stitches, the application of the first kind of list or second one depends on the narrow section type [5]. If the narrow section is a graphic representation of a half symmetric part of knitting cut, it is better to use linked list. On the other hand, if the detail is a whole narrow section, the circuit doubly linked list is better decision.

As it already mentioned, the item of the list is a dot with x, y coordinates. The item can be described with an object of CPoint class. The declarations of structure **node** and class **list** are as follows:

```
struct node
{
        CPoint item;
        node *next;
};

class list
{
private:
        node *begin;
        node *current;

public:
        list();
        ~list();
        void Add(CPoint);
        void Insert(node *, CPoint);
        void Delete(node *);
        void DrawAll();
        void EraseAll();
        bool EmptyList();
        node* Search(CPoint);
        CPoint GetCurrentPoint();

};
```

Member-variable **begin** is a pointer to the node at the head of the list; data member **current** points to current dot position. The last one is used as returned value of **GetCurrentPoint()** method. This value is applied in **OnDraw()** method to draw dots and the contour of the detail. Methods of the class are described on Table 1. Coding of methods **Insert()** and **Delete()** follows the diagrams on fig. 4 and fig. 5, respectively.

Construction is an MFC application use list class. The main window of the application is represented on fig. 6. According to the active mode, the operator can add, or insert a dot to the detail construction or delete (remove) a dot from the construction by left mouse button click. These functions could be started using menu item **Draw**. It consist also menu item **Erase All** called **EraseAll()** method. Because the class

**list** destructor needs to delete all list's items, it calls **EraseAll()** method, too.

TABLE I
METHODS OF CLASS LIST

| method | description |
|---|---|
| list(); | constructor |
| ~list(); | destructor |
| void Add(CPoint); | Add a new dot to the end of the list |
| void Intert(node *, CPoint); | Insert a new dot after pointed dot |
| void Delete(node *); | Delete a dot |
| void EraseAll(); | Erase all dots |
| bool EmptyList(); | Check if the list is empty |
| node* Search(dot); | Search a dot |
| CPoint GetCurrentPoint(); | Return Current dot |



Fig. 6. Main window of the application

## V. CONCLUSION

This paper suggests an approach to data representation in CAD system development based on a set of programming modules. Some of these modules can be used in next generation of CAD systems or they can be embedded in another system - CAD software. The main reason to create

such project is that it can be used as software application from fashion designers to make boutique collections of knitting products. Listed data structures are successfully used to code graphic data related to knitting pattern object. Knitting patterns are represented as sets of listed dots, so it is the reason to use listed data structure to describe graphically knitting pattern object.

The goal of the paper is to give an approach of usage of listed structures in CAD systems' data representation. In this particular system the structures linked list is used. The data type does not allow using listed structures in MS VC++ collections. That is the reason to create listed structures related to knitting pattern graphic objects. The algorithms operated with list structures are used. These algorithms are applied to particular software module "Construction". The application can be embedded in whole CAD/CAM system in knitting industry or it can be used as software application creating patterns of knitting products. The MS VC++ and MFC are used to create a software application. The main reason to create such project is that it can be used as software application from fashion designers to make boutique collections of knitting products. The project can be used as an example for CAD system development.

REFERENCES

[1] Angelova Y., E. Zaharieva-Stoyanova, Precision of the shape of garment pieces at Fully Fashion Knitting, *Textile & Cloths*, 7/2006.
[2] Baczynski M., Data Structures: Linked List.
[3] Zaharieva-Stoyanova, Computer Aided Design of Knitting Products Using FF Method, *International Scientific Conference UNITECH'2007,* 23-24 November, Gabrovo, Bulgaria.
[4] Zaharieva-Stoyanova E., Methods of Graphic Representation of Curves in CAD systems in Knitting Industry, *XLII International Scientific Conference on Information, Comunications and Energy Systems and Technologies ICEST 2007*, 24-27 June, 2007, Ohrid, Macedonia.
[5] Zaharieva-Stoyanova E., Application of Bezier Curves in Knitting Industry CAD/CAM systems, *XL International Scientific Conference on Information, Communication and Energy Systems and Technologies – ICEST 2005*, June 29 – July 1, Nis, Serbvia and Montenegro, 2005.
[6] Zaharieva-Stoyanova E., Application of Bezier Curves in Knitting Industry CAD/CAM systems, *XL International Scientific Conference on Information, Communication and Energy Systems and Technologies – ICEST 2005*, June 29 – July 1, Nis, Serbvia and Montenegro, 2005.
[7] Zaharieva-Stoyanova E., *Algorithm for Computer Aided Design Curve Shape Form Generation of Knitting Patterns*, *International Conference on Automation, Quality and Testing, Robots AQTR 2006(THETA 15),* May 25-28 2006, Cluj-Napoca, Romania.
[8] Data Structures and Algorithms, http://www.cs.auckland.ac.nz/software/AlgAnim/lists.html
[9] *Trend Collection 2004/2005*, Stoll, Department Fashion & Technology
[10] Sirix 110/210, Instructions, Part I, II, Reutlingen, 2001.
[11] DeSignaKnit, Demo, www.knitcraf.com.
[12] Stitch&MotifMaker, www.softwareknitting.com
[13] SweaterWizard2001, Demo, www.softwareknitting.com
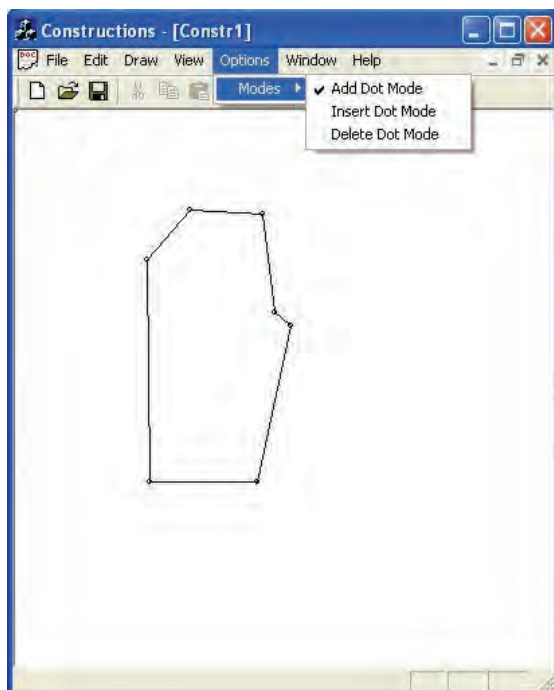[14] WiseTex, Demo, www.mtm.kuleuven.oc.be