Object-Oriented Approach to CAD System for Knitting Industry Application

Elena Zaharieva-Stoyanova
TU-Gabrovo, Gabrovo, Bulgaria

***Abstract**: This paper suggests an approach to CAD system development based on its creation as 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 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. The project shows how to create classes for CAD system as software application. Data types plus function pertaining to data are likewise defined. Suggested CAD system would be used as an application for making boutique collections by fashion designers or as a part of another system.*

***Keywords:** CAD/CAM systems, knitting industry, data structures, object-oriented programming.*

1. INTRODUCTION

The development of CAD/CAM systems for knitting industry automation is conditioned by technological improvement of knitting machines and by the advance in computer science, as well. Software rapidly changes along with hardware technologies development. New Operation Systems are continuously being introduced to the market plus a number of platforms and programming environments.

Normally, the companies manufacturing knitting machines also put on the market CAD/CAM/CAI software packages delivered with whole hardware equipment. Non-standard equipment was usually in use before due to the than existing a trademark protection policy. However this increases CAD/CAM systems' prices several times [6], [7], [8], [9], [10], [11].

At present, companies prefer standard computer hardware. Therefore, the CAD/CAM/CAI systems' developers make efforts to create new software packages according to demands of new version Operation Systems. Since software development demands a lot of efforts and human resources it is better to use relevant modules from previous CAD/CAM systems versions. Moreover, knitting manufacturing technologies are not so frequently modified, as is the case with software development. Accordingly, it is important to develop CAD/ACM software in modules, which can be alternatively used in other systems [1], [7],[8],[9].

The goal of this paper is to offer an approach to a specific CAD system development, which is based on a relevant set of special purposes programming modules. Some of these modules can be used in next generation of CAD systems or they can be embedded in another CAD software. [2]

Another main reason to create such software project is that it can be used as software application by fashion designers to make boutique collections of knit products. CAD/CAM/CAI software delivered with knitting machines is custom developed. The user of such kind of a system must be well familiar with knitting machine features and knitting technologies process as a whole. For a fashion designed such software is hard to handle. The project aims at more user-friendly software.

2. DEMANDS TO CAD SOFTWARE FOR KNITTING PRODUCTS DESIGN

All contemporary flat-knitting machines can manufacture tailor-mades of knitting products. This method is called "Fully Fashion" (FF). To knit a part of products (narrow section) the machine is supplied with a narrowing and widening device. Generally, the design of knit products has two aspects: Knitting structures design and Design of knitting patterns (cuts) shapes. These two aspects are related to each other.

According to the conventional design, the model of a knit product (jacket, sweater, etc.) is created based on some standard constructions. The term "construction" determines the whole style of the product; the number of its parts and their shape [6], [10].

Each model can be manufactured in several sizes determined by the Standards of knit products. They define the sizes of knitting products' cuts. To create original and authentic models designers make use of different kinds of knitting techniques such as jacquard, intarsia, gusset, lace and other knitting structures. The techniques employed are involved as a part of the model. It means that creation of knitting structures is a part of a whole process of knitting design [6],[7].

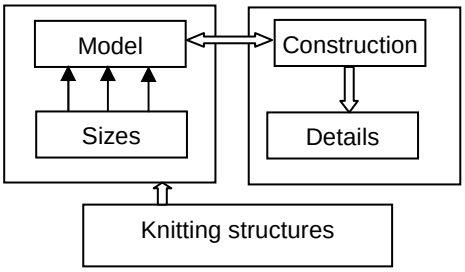

Fig. 1: Aspects of a knitting design process.

All aspects of a knitting design process and their relations to each other is shown on fig.1. As a process, knitting products design is to cover all these aspects, so that the demands to CAD/CAM systems for knitting industry in general can be described as follows:

- Capacity to create and edit a construction of model including all its parts (cuts) of the product.
- To support a set of standard constructions of knitting models.
- To create a knitting model including all sizes variation.
- To create different types of knitting structures according to machine abilities to manufacture different knitting techniques (jacquards, intarsia, gusset, lace, and so on).
- Automatic generation of knitting control programs implementing the models.

The last one is a CAM module function; the rest is demands to CAD software. As a software application, a CAD system can be determined as a graphics editor creating and describing knitting structures and cuts of knit products [7], [8].

3. FEATURES OF COMPUTER AIDED DESIGN 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 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 presents a project for CAD system development with possibilities to create models of knitting products; their constructions and variety of sizes. Knitting structures and patterns design module will be a point of further development.

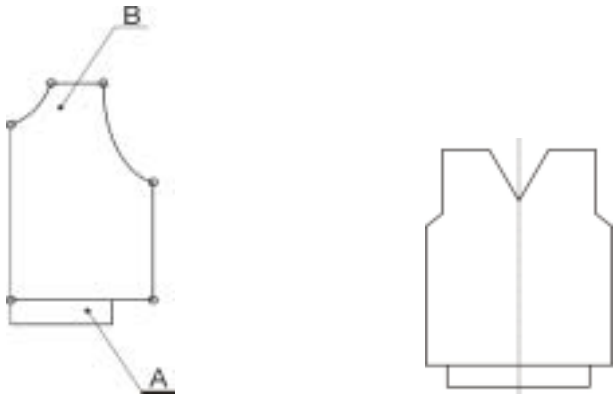

Fig. 2 (a,b): Examples of narrow sections.

To create a whole set of classes determining data and their pertaining functions, the programmer needs to know the type and the characteristics of designed objects. Figure 2 (a,b) presents examples of knitting products narrow sections. The main difference between them is that figure 2a presents a half symmetric part of the narrow section (cut) while figure 2b 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. 2). 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].

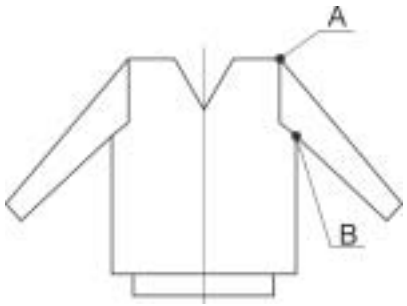

Fig. 3: Examples of a knitted product construction.

There are two manners to represent a cut of knitting product (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. For this reason, the first manner of coding is better. Knit products could be regarded as a set of details. Fig.3 shows an example of knitting products construction. A and B are points belonging to two details. Such points fix a connection between details and determine that the distances between them for both details should be equal.

4. REPRESENTATION OF DESIGNED OBJECT AS SLASSES AND DATA TYPES

4.1. Representation of knitting products' details

As it has already been mentioned the graphic representation of knitting products details is in the form of a polygon by straight lines and curves. Because both items are lines, an abstract base class Line would describe them. The start and end point are data-members of this class. Class Line is a base class that is why data-members belong to protected section.

Class StraightLine and class Curve are derived classes inheriting the base class Line. They describe a straight line and a curve. Tables 1 present the data-members and member functions of the abstract base class Line and the derived classes – StraightLine and Curve. All methods of the abstract class are pure virtual and they are overloaded in derived classes.

Tab. 1: Lines describing classes.

abstract base class Line		
	Name	Description
Data Members	CPoint StrartPoint; CPoint EndPoint;	*First point of line/curve.* *End point of line/curve.*
Methods	virtual void DrawLine()=0; virtual void InputDots()=0;	*Draws line or curve.* *Inputs first and end points.*
Derived class StraightLine		
Data Members	StraightLineType typeLine;	*Defines line type.*
Derived class Curve		
Data Members	CurveType typeLine; int numberKnots; CPoint* Knots;	*Defines curve type.* *Number of knots.* *Array of knots.*

Methods	void Bezier2();
	void Bezier3();
	void BSpline();
		Draws a Bezier curve using 2 knots.
		Draws a Bezier curve using 3 knots.
		Draws spline curve.

The type StraightLineType is an enumeration for the following kind of lines: horizontal, vertical, slash. Enumeration CurveType determines the used curved representation method.

There are some possibilities to describe a whole detail: array of lines and list of lines. The former is the easier way for programming whereas the latter leads to better code efficiency and data representation. The list 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.

Class Detail is formed as a linked list of lines. It represents a half of narrow section. Class Detail is a base class for class DoubleDetail describing a whole narrow section. It corresponds to a circuit doubly linked list. The derived class methods are overloaded. All methods are virtual functions. The data-members and methods of classes Detail and DoubleDetail are shown on Table 2.

Tab. 2: Detail describing classes.

Base class Detail		
	Name	Description
Data Members	char* nameDetail; Line* begin; Line* next;	*Determine a kind of detail.* *The beginning of the list.* *Next list's item.*

Methods	virtual Line& InsertLine(Line*);
	virtual void DeleteLine(Line*);
	virtual void AddLine(Line*) ;
	virtual void DrawDetail();
	virtual void ConvertStitches (double, double);
		Inserts a new line in the list.
		Removes a line from the list.
		Add a line to the list end.
		Draws a narrow section.
		Describes the narrow section as stitches and rows.

Derived class DoubleDetail		
Data Members	Line* previous;	*Previous list's item.*

4.2. Representation of knitting products' construction

Classes describing a knitting construction and models are represented on Table 3. The data members of class Construction are object array of class Detail and an array corresponding to table giving information about their connections.

Tab. 3: Class Construction.

	Name	Description
Data Members	char*nameConstruction; int numerDetails; Detail* narrowSections; int* DetailType int** tableConnections;	*Determine the kind of construction* *Number of narrow sections.* *Array of details.* *Determine the list type.* *Table describing connections between details.*

Methods	void InsertDetail();
	void DeleteDatail();
	void EditDetail();
	void CopyDetail();
	void DrawConstruction();
		Creates a new detail.
		Removes an existing detail.
		Edit an existing detail.
		Copies a detail and its properties to another construction.
		Draws construction.

Tab. 4: Class Model.

	Name	Description
Data Members	char*nameModel; int numnerSizes; char** nameSizes; int** sizesTable; int** margins	*Determine the kind of model.* *Defines the number of sizes* *Define the sizes of details* *Table with sizes of details.* *Define margins of details.*

Methods	void InsertSize();
	void DeleteSize();
	void EditSize();
	void ShowSizeTable();
	void GenerateSizes();
	void InsertMargin();
	void DeleteMargin();
	void EditMargin();
		Inserts a new size.
		Deletes an existing size.
		Edit an existing size.
		Shows size table.
		Generates sizes by using standard automatically.
		Inserta a margin.
		Deletes a margin.
		Edits a margin.

Array DetailType determine the type of the used list: a linked list or a circuit doubly linked list. It determines a kind of detail: a half symmetric part of the narrow section or whole narrow section. Class Model inherits class Construction. The own data-member of class Model is a size table describing the sizes of the details. Table 4 represents this information.

Data members of the class Construction belong to protected section. Normally, member-functions belong to public section. The derived class Model can be accessed all data-members of the base class Construction. Data about margins (borders) of the details and function operated with them are included, too.

5. CONCLUSION

This paper suggests an approach to CAD system development based on its creation as 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. The object-oriented programming is used. The project shows how to create classes for CAD system as software application. The data type, data and function operated with them are defined. Using inheritance as software technique creates the hierarchy of classes. The abstract base classes are applied, too. Most of the methods are virtual functions. The CAD system would be used as an application for making boutique collections by fashion designers or as a part of another system. The project can be used as an example for CAD system development.

6. REFERENCES

[1] Angelova Y., E. Zaharieva-Stoyanova, Precision of the shape of garment pieces at Fully Fashion Knitting*, Textile & Cloths*, 7/2006.
[2] Zaharieva-Stoyanova, Computer Aided Design of Knitting Products Using FF Method, *International Scientific Conference UNITECH'2007,* 23-24 November, Gabrovo, Bulgaria.
[3] 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.
[4] 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.
[5] 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.
[6] *Trend Collection 2004/2005*, Stoll, Department Fashion & Technology.
[7] *Sirix 110/210, Instructions*, Part I, II, Reutlingen, 2001.
[8] DeSignaKnit, Demo, www.knitcraf.com.
[9] Stitch&MotifMaker, www.softwareknitting.com
[10] SweaterWizard2001, Demo, www.softwareknitting.com
[11] WiseTex, Demo, www.mtm.kuleuven.oc.be.