

ClothAssembler: a CAD Module for Feature-based Garment Pattern Assembly

Marzia Fontana¹, Alberto Carubelli¹, Caterina Rizzi² and Umberto Cugini³

¹ Università di Parma, marzia@ied.eng.unipr.it

¹ Università di Parma, alberto.carubelli@studenti.unipr.it

² Università di Bergamo, caterina.rizzi@unibg.it

³ Politecnico di Milano, umberto.cugini@polimi.it

ABSTRACT

This work presents a CAD prototype, named ClothAssembler, targeted at complex-shaped apparel design for real manufacturing. The intent is to fill a gap in the current CAD technology for garment design as it is mainly conceived for 2D/3D geometric modelling of cloth shapes, but generally does not provide high level operators that allow interactive and easy design of aesthetic/functional features that characterize the garment pieces, and relations/connections between parts. Though still an academic prototype, ClothAssembler allows to define/choose in an interactive way all the necessary geometric and functional information for the design and finishing of 2D pieces, such as insertion of textile layers, reinforcement lines, pockets, cut lines and pleats, as well as topological information about how pieces are pair-wise connected and assembled, by definition of seams, darts, zips, constraints such as buttons and hooks, etc. A taxonomy and parametrization of cloth tailoring features is discussed, and system functionalities are presented, with applications to garment models of real production.

Keywords: garment design, CAD, tailoring features.

1. INTRODUCTION

The textile and clothing industry had been characterized, in the past, by a certain resistance to automation and use of computer-assisted technology. This is/was due to the creative and skilled-labour dependent nature of cloth design and manufacturing processes. Despite such complexity, CAD tools are nowadays increasingly demanded by cloth manufacturers to support modellers in the 2D and 3D design phases, for stress analysis tests on textiles or for virtual apparel simulation purposes, to evaluate/improve the quality of apparel products through digital models, before (or in place of) any physical prototyping, thus reducing production times and costs. As an incentive to the development of CAD systems for cloth design, there exists a large literature of methods and algorithms for cloth modelling, according to various geometric, physics-based and hybrid techniques [1, 4, 7, 10, 6].

Several CAD systems exist on the market specialized to the clothing sector, e.g., CAD modules from companies such as Lectra Systèmes & Investronica Sistemas, Gerber Technology, PAD System Technologies, Koppermann Computersysteme, Optitex, Browzwear, Digital Fashion, F.K. Group, and others [8]. Beyond certain predictable differences in the GUI appearance, used development tools, data formats or commercial costs, they present many common characteristics. Most of them, in fact, are mainly specialized to the 2D pattern design phase, or in CAD-CAM data transfer with NC machines for textile spreading and cut, and only few include modules for 3D shape modelling and virtual simulation (e.g., the products V-Stitcher™ by Browzwear, OptiTex Runway™ 3D by OptiTex and DressingSym™ by Digital Fashion). Here, we do not focus on 3D cloth modelling and simulation methods, already presented in [5], but rather come back again to the earlier 2D design phase as we feel that, despite such a widespread 2D CAD technology, some aspects need further investigation.

In fact, most of the existing 2D CAD models are conceived for the design and cut of *unassembled* textile parts but, generally, do not provide sufficient information about assembly and finishing rules for effective garment manufacture starting from its single separated pieces. Or, such information can appear in notes/files but not well-organized in a CAD format to enable automatic manufacturing. The intent of the present work is to propose a possible solution to the above limitation. A CAD module has been developed, here presented, to assist the designer/modeller to create his/her

garment model during the 2D design phase by interactive specification of aesthetic and functional *features* that define the peculiar look and function of the garment. A user plays the role of ‘virtual tailor’ and can interactively define elements and details that characterize each cloth piece, e.g., type and location of textile layers and reinforcements, pockets, pleats, cut lines, etc., and fix rules for part connectivity, such as seams, darts, constraints, etc. (Fig. 1). The corresponding geometric and topological information are written in a structured format, available for successive CAD manufacturing stages.



Fig. 1. Seams, pockets, buttons, zips and other finishing details in manufactured apparel.

2. AN INTEGRATED ENVIRONMENT FOR APPAREL MODELLING AND SIMULATION

The present work has been developed in the framework of an Italian PRIN Project (Research Project of National Interest) named VI-CLOTH (Virtual CLOTHing), result of a collaboration between Italian departments at the Università di Bergamo, Politecnico di Milano, Università di Firenze and Brescia, cloth manufacturers and CAD-CAM developers. The general objective is the analysis and development of an integrated 3D CAD system for the virtual prototyping of apparel for real manufacturing purposes. In the next Section we briefly describe the main research goals of the project and the overall platform in which our research activity on apparel design takes place.

2.1 Research Phases in the VI-CLOTH Project

The Vi-CLOTH project aims at the definition of a new computer-assisted design workflow in the clothing industry based on the integrated use of reverse engineering and virtual prototyping techniques. The final goal is the realization of a software environment that allows modellers to design made-to-measure garments, in which the aesthetic and functional validation of the models is done directly in the digital phase before (or in place of) any physical prototyping; this requires accurate physics-based garment simulation in a 3D virtual scene, taking into account properties of materials and interactions with virtual human bodies and the external environment.

Under these objectives, the VI-CLOTH project provides for the integration of different technologies and software resources, for both human body and apparel modelling, based on low-cost equipments as demanded by cloth manufacturers. Briefly, the project's integrated system will include the following modules:

1. a module for *human body data acquisition* by a portable digitalization system;
2. a module for *parametric mannequin modelling*, with 3D models derived from the acquired human body data, which generates a complete anthropometric characterization of the human body and, at the same time, enables local shape/size variation or derivation of other postures for the successive dressing phase;
3. a module for *2D→3D geometric garment modelling*, which generates from 2D unassembled cloth pieces the assembled and finished garment shape in a 3D pre-simulated configuration;
4. a module for *physics-based garment simulation in 3D*, with accurate prediction of the garment shape behaviour when placed on virtual mannequins collocated in a 3D scene.

More information about the project and the various research activities can be read in [2]. In the present work we will focus on results obtained within phase 3, concerning the apparel design and assembly stage.

2.2 Activity on Apparel Design

Fig. 2 shows the high-level architecture of the apparel modelling and simulation stages within the VI-CLOTH project. Two main environments can be distinguished, 2D and 3D, both equally important for our apparel design vision that takes into account manufacturing steps for real garment production. A first 2D CAD environment is used for the initial

design of the 2D pieces that compose a garment, and the successive definition of finishing and assembly information on/among pieces. In a successive 3D virtual environment, the various garment pieces are placed around a virtual human body and, starting from an initial assembled 3D configuration, the final garment shape is then simulated according to some accurate physics-based model.

The first environment includes three essential modules:

1. a *2D design system*, for garment pattern definition and editing of all 2D pieces;
2. an *assembly module*, for the design of features and finishing details on each piece and the definition of assembly/connection elements to relate the various pieces with each other;
3. a *geometric tessellation module*, to locally guide the folding in the successive 2D→3D mapping process.

The second environment includes:

4. a *2D→3D mapping module*, to place the garment in a 3D assembled configuration;
5. a *physics-based modelling module*, based on a garment discrete representation by particle grids;
6. a *3D garment simulation module*, based on a constrained Newtonian dynamics model.

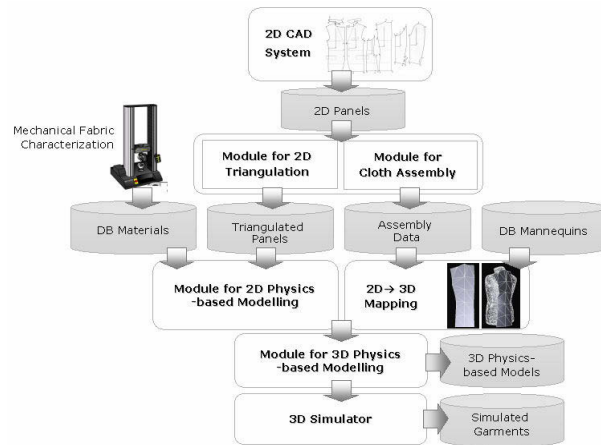


Fig. 2. Apparel modelling and 3D physics-based simulation: the overall system architecture.

Several commercial tools are available for the design of 2D pattern pieces (Section 1): for instance, we used TexWinCAD™ by the Italian F.K. Group company [8], a module for pattern design in which piece contours were drawn as closed polygonal curves connecting vertex points, provided with reference hole points and notches (Fig. 3). This step leads to purely *geometrical* information. An additional level of *topological* and *functional* information comes then by the assembly module, as it will be described in the next Sections. This information is fundamental to define construction details of the garment parts, constraints and connectivity rules, to be used for the successive physics-based modelling phase [5], leading to accurate simulation results of industrial interest only if they originate from a detailed garment model description.

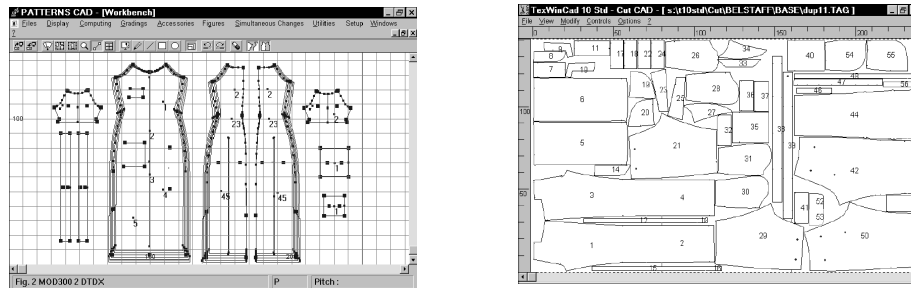


Fig. 3. A commercial 2D CAD system: TexWinCAD™ (Image courtesy by F.K. Group).

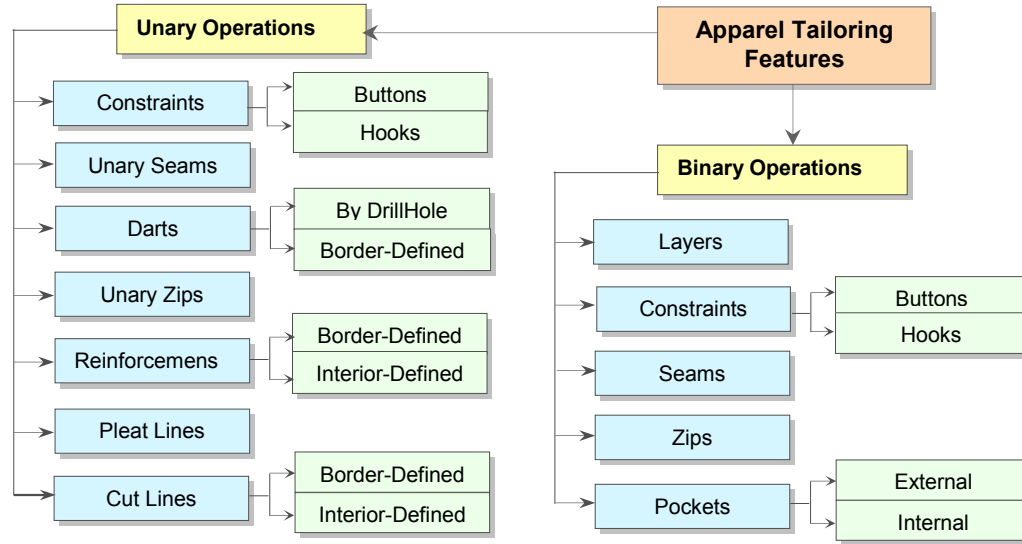


Fig. 4. A taxonomy of tailoring features for garment manufacturing.

3. APPAREL DESIGN AND ASSEMBLY

Real tailored apparel can be particularly complex (e.g., jackets, coats), full of aesthetic and functional elements, inserted in the basic textiles for better characterization, or for connecting parts with each other. As these elements do exist, a CAD system that aims at emulating the designer/tailor activity, or at making it faster, should include them. These are, for instance, layers and reinforcements, seams and stitches, darts, zips, buttons and hooks, holes and cut lines, pleats, various types of pockets, and so on. Our effort has been then to create a graphical tool that allows insertion of these elements within the single cloth pieces or for connecting/sewing parts. In the following, we describe the data structure and the design functionalities that have been conceived for this purpose.

3.1 A Taxonomy of Tailoring Features

Our intent is to define a computer-assisted procedure enabling automatic insertion of styling details or assembly rules by few high level parameters chosen/controlled by an user playing the role of a 'virtual tailor'. Following a feature-based approach, we name these styling/functional entities as *tailoring features*, because they introduce geometric-topological information to single cloth panels or groups of them, as a 'semantic' rule that can be applied, deleted, or repeated on the various panels. Obviously, these features are *context-dependent*, as the semantics underlying each feature class is dependent on the clothing manufacturing domain. We will consider:

- *panel features*, as aesthetic/functional rules applied on a single panel, causing a local modification in the original geometry or material of the panel, e.g., by reinforcements, local treatments, cuts, pleats, etc.;
- *assembly features*, as rules for connecting two (originally unassembled) panels, or (originally separated) parts of the same panel, e.g., by seams, zips, small rigid constraints to fix pieces together, etc.

We first single out the different roles among the various 2D pieces defined in the early 2D CAD phase. In fact, some of them define the 2D geometry for the main visible textiles (named *main panels*), while other pieces will correspond to hidden/internal textile *layers* to be placed below a main panel or any union of them. A man suit, for instance, has its main textiles (e.g., cotton, leather, etc.) exhibited at the exterior sides; yet, these textiles can be composed of several interior textile layers such as linings, stuffings, reinforcements that, though not visible, still contribute to the final style and look of the suit. Our criterion is then to define tailoring features for the main panels; then, the corresponding rules will affect the possible layers located in the region where each feature is imposed.

Resorting to a mathematical abstraction, panel and assembly features correspond, respectively, to *unary* or *binary* relations: panel features, in fact, can be seen as functions $f_a: P \rightarrow P^*$, where P is a main panel and P^* is the modified panel after the feature insertion; while assembly features can be regarded as functions $f_a: P \times Q \rightarrow R$, where P and Q are two main panels, or disjoint subregions of the same panel, and R is the resulting assembled panel or region (for

instance, $R=P \cup Q$ for a sewing process). The various meanings for the new characterized or assembled panels, respectively P^* and R , depend on the type of manufacturing operation. In our first implementation, we considered panel and assembly features, according to the taxonomy presented in Fig. 4. According to that hierarchy, we thus manage:

- location and type of additional *textile layers* stuck or sewn on main textiles;
- different types of assembly rules, such as unary and binary *seams*, or *darts*, to join two disconnected borders of the same panel or two different panels;
- other assembly rules, such as *zips*, to temporarily connect two disconnected panel borders;
- location and type of border or internal *reinforcements lines*;
- location and type of border *cut lines* or internal *cut holes*, to partially disconnect parts, leading thus to a local change in the topological genus of the original panels;
- location and type of *pleating lines* to impose a fold on textiles;
- different types of *point constraints*, e.g., buttons and hooks, imposing a point-wise connection between two (border or internal) points belonging to the same panel or two different panels;
- different types of *internal* and *external pockets*, tight-fitting the main textiles, or large.

Though not yet complete, this classification allows to manage a sufficiently wide class of design details and assembly procedures used for manufacturing daily clothing, such as T-shirts, skirts, blouses, jackets, trousers, coats, and so on. The next step is now to define opportune sets of parameters that allow a complete identification (and, therefore, implementation) of the above defined tailoring features.

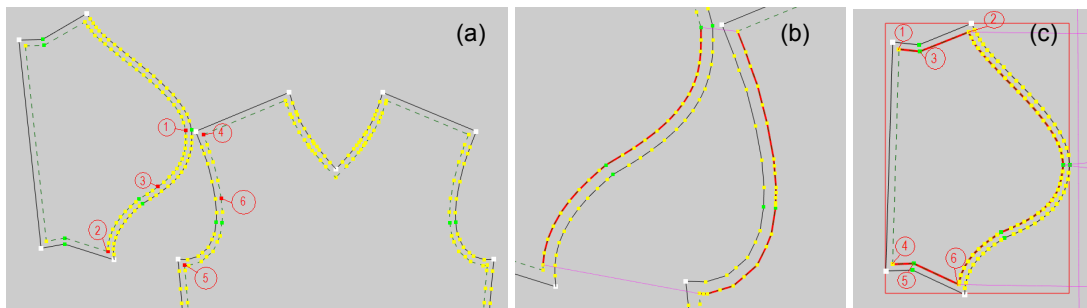


Fig. 5. Assembly between garment panels: (a) global view, (b-c) details of two seams, defined as 1-to-1 mappings between sequences of consecutive border edges.

3.2 Feature Parametrization

In this Section, we describe the parameters that have been considered to characterize each of the tailoring features described in Fig. 4.

Layers are defined as subregions of a main panel composed of a further material added to the main textile. The following list of parameters can be used to describe a layer:

- index id , to identify it as 2D panel in the list of the garment pieces;
- name, a string for layer description;
- index id_p of the main piece above/below which it will be placed;
- index id_l that identifies the closed polyline that describes the layer geometry;
- number N of the layer border vertices;
- type of material (e.g., canvas);
- type of overlapping, i.e., whether stuck or sewn;
- sequence of positions (x_i, y_i) , for $i=1,2,\dots,N$, describing the locations of the layer border vertices within the main panel.

Pleat lines are defined by open lines around which a bending effect is imposed, using these parameters:

- index id , which identifies the panel that will contain the pleat line;
- angle α , describing a forced bending determining the pleat effect;

- orientation, i.e., whether inwards or outwards;
- type of entities defining the reference pleat polyline, i.e., whether node indexes or point coordinates;
- number N of entities (node indexes or xy -coordinates);
- sequence of entities p_i (or x_i, y_i), for $i=1,2,\dots,N$, defining the pleat polyline;
- index of a border node, defining the initial location of the pleat on the panel border.

Reinforcements are material strips added to a main panel, following the direction of a polyline. They require:

- index id to identify the panel that will contain the reinforcement;
- width w of the rectangular reinforcement region around a reference polyline;
- type of material for the reinforcement (or, as equivalent information, mass density of the material);
- type of entities considered to define the polyline, i.e., whether node indexes or xy -coordinates;
- number N of entities;
- sequence of entities p_i (or x_i, y_i), for $i=1,2,\dots,N$, defining the reference polyline.

Border cuts correspond to textile breaks or cuts, defined by:

- index id that identifies the panel that will be partially cut at some border point;
- index p_0 of border node from which the cut originates;
- type of entities considered for the cut polyline definition, i.e., whether node indexes or xy -coordinates;
- number N of entities;
- sequence of entities p_i (or x_i, y_i), for $i=1,2,\dots,N$, defining the polyline driving the cut.

Internal cuts correspond to cut lines or holes, with material removal, caused at the interior of a panel, defined by:

- index id , identifying the panel that will contain the interior cut;
- type of cut, i.e., whether defined by an *open* or *closed* polyline;
- type of entities considered to define the cut polyline, i.e., whether node indexes or xy -coordinates;
- number N of entities (node indexes or xy -coordinates) of the polyline (if closed, the last node/point is not repeated);
- sequence of entities p_i (or x_i, y_i), for $i=1,2,\dots,N$, defining the open or closed polyline driving the cut.

Seams and darts (Fig. 5) are obtained by mappings between borders, and are defined by:

- index id_1 to identify the first panel (or the unique panel for unary seams and darts) that will be sewn;
- (only for binary seams) index id_2 , to identify the second panel to be sewn with the first one;
- angle α describing a forced bending effect to the seam (if null, then the seam is flat, i.e., the two panels are connected with G^1 continuity);
- type of entities considered for seam/dart definition, i.e., whether border node indexes or border edges;
- number N of connected entities (node indexes or edges) for each border;
- sequence of 1-to-1 mappings $p_i \rightarrow q_i$, for $i=1,2,\dots,N$, between the entities of the first and the second panel.

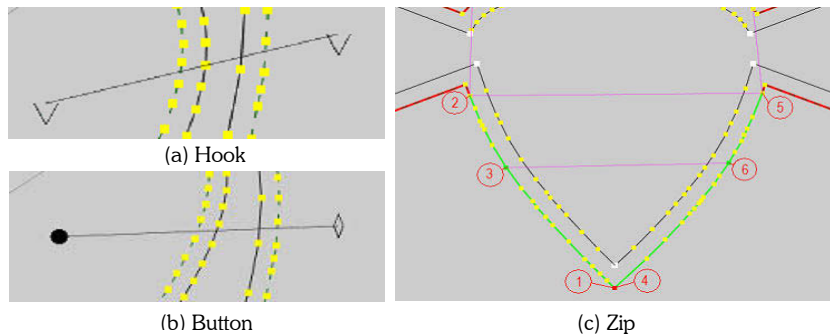


Fig. 6. Other assembly procedures: by point constraints, e.g., (a) a hook, (b) a button; as mappings between border polylines, e.g., (c) a zip.

Note that darts can be regarded as seams, in which the first mapping is such that $p_1 \rightarrow p_1$, as the two border sequences that have to be sewn have, by definition, a first vertex p_1 already in common. Similarly to seams, unary and binary zips (Fig. 6) use the following parametrization:

- index id_1 , to identify the first panel (or the unique panel for unary zips) whose border will have a zip;
- (only for binary zips) index id_2 , to identify a second panel;
- type of entities considered for zip definition, i.e., whether by border node indexes or border edges;
- number N of connected entities (node indexes or edges) for each border;
- sequence of 1-to-1 mappings $p_i \rightarrow q_i$, for $i=1,2,\dots,N$, between the entities of the first and the second panel;
- index of the first entity at the border of the first panel, to define the zip orientation;
- (only for simulation purposes, i.e., to simulate an effective zip configuration) index of the last entity effectively connected by the zip (if closed or semi-closed).

Buttons and *hooks* (Fig. 6) act as constraints between pairs of points and require:

- index id_1 , to identify the first panel (or the unique panel if the constraint is unary);
- (only for binary constraints) index id_2 , to identify a second panel;
- placement of the first panel with respect to the second, i.e., whether inwards or outwards;
- position (x_1, y_1) of the button/hook in the first panel;
- position (x_2, y_2) of the button/hook in the second panel;
- mass m of the button/hook;
- (only for buttons) diameter d of the button.

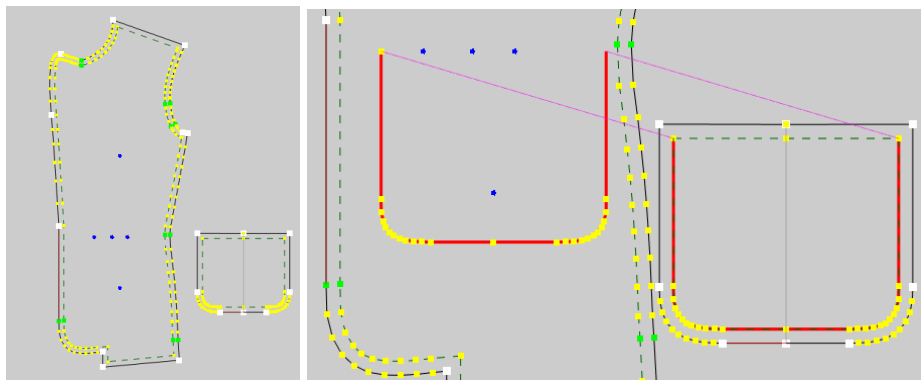


Fig. 7. Pocket, defined by a sewing process guided by reference drill holes that connects the border of a small panel with a polyline defined inside the main panel.

Pockets (Fig. 7) correspond to small regular panels whose borders are sewn along interior polylines of a main panel. They can be external or internal. *External pockets* (e.g., pockets of jeans wear) can be defined as:

- index id_1 , to identify the main panel in which a pocket will be inserted;
- index id_2 , to identify a panel corresponding to the pocket;
- type of entities considered in the main panel, i.e., whether node indexes or xy -coordinates;
- type of entities considered in the pocket panel, i.e., whether border node indexes or edges;
- number N of connected entities while creating a pocket (node indexes or xy -coordinates for the main panel, border nodes or edges for the pocket);
- sequence of 1-to-1 mappings $p_i \rightarrow q_i$, or $(x_p, y_p) \rightarrow q_i$, for $i=1,2,\dots,N$, between the entities of the first panel and the pocket panel.

Internal pockets have a more complex definition, as they can be manufactured in different ways and styles. The simplest internal pocket definition could be done by considering parameters analogously to external pockets and, in addition:

- type of entities considered to define the opening in the main panel, i.e., whether node indexes or xy -coordinates;
- index s_1 or coordinates (x_1, y_1) of the first vertex of a cut line in the main panel defining the pocket opening;
- index s_2 or coordinates (x_2, y_2) of the second vertex of the cut line in the main panel defining the opening.

Of course, there can be alternative parametrizations. Ours is a possible solution. From this classification we understand how, in principle, further design and assembly features could be defined and easily parametrized, or obtained by

composition of these basic classes, most frequently used in real garment manufacturing. In the next Section we describe the module that has been implemented, based on the proposed tailoring features.

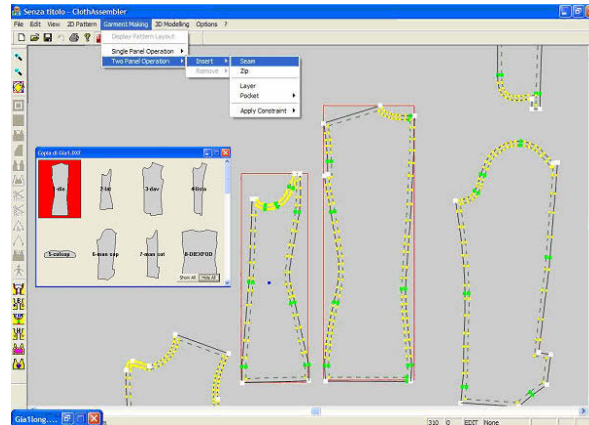


Fig. 8. ClothAssembler's graphical interface.

4. IMPLEMENTATION

A module has been developed, named *ClothAssembler*, provided with a graphical user interface, that allows a user to insert interactively styling/functional elements and assembly rules to the garment pieces of a 2D pattern model, based on the above described taxonomy.

The system reads information of unassembled 2D garment geometries given in a standard DXF/AAMA format, a widely used format for the clothing sector, and returns a detailed description of the geometric and topological information relative to the 2D figures defining the garment model, comprehensive of all data generated from interactive feature insertion, as performed by the user. The output is written in standard XML format with complete attribute specification, whose tags describe: measure units, list of points, list of polygonal figures, list of main pieces and layers, and lists of unary/binary relations according to the hierarchy shown in Fig. 4.

Fig. 8 shows a snapshot of ClothAssembler's graphical interface. The main system includes:

- file management and *edit* functions, e.g., DXF-AAMA file import, XML file export, data and image printing, customization options, and usual editing tools, e.g., undo/redo actions, cut/copy/paste, selection of entities;
- 2D *view* modes, i.e., global layout and detailed views, show/hide and zoom/pan/translate/rotate options;
- functions for 2D *pattern* editing and attribute/property specification, e.g., drill hole/notch insertion, assignment of material, layers and texture;
- the *garment making* main modality, allowing insertion of all tailoring features with selection of entities (pieces, curves, points, notches, etc.), and user-friendly parameter specification;
- functions for panel location in 3D, based on the assignment of anthropometric/Cartesian reference data.

Though still a 2D CAD module, the last modality anticipates functionalities for 3D apparel modelling, that will be embedded in the future version of the system to have integration with the physics-based simulation engine presented in [5].

ClothAssembler is developed in VisualC++, uses MFC (Microsoft Foundation Classes) for visual interface programming, and OpenGL for graphical modelling functions. The implemented classes can be grouped into three main categories:

- import/export classes: e.g., *DXFparser*, to read a DXF-AAMA format, *PieceBuilder*, to generate the piece object from the loaded data, *XMLwriter*, to write output XML files;
- tailoring classes: e.g., *Piece*, *Line*, *Ppoint*, *DrillHole*, *Button*, *ButtonPair*, *Seam*, *UnarySeam*, *Dart*, *Zip*;
- various classes for the graphical interface.

Fig. 9. shows a UML diagram describing the various panel/assembly feature classes with their interdependencies. For further details on implementation aspects, see [3].

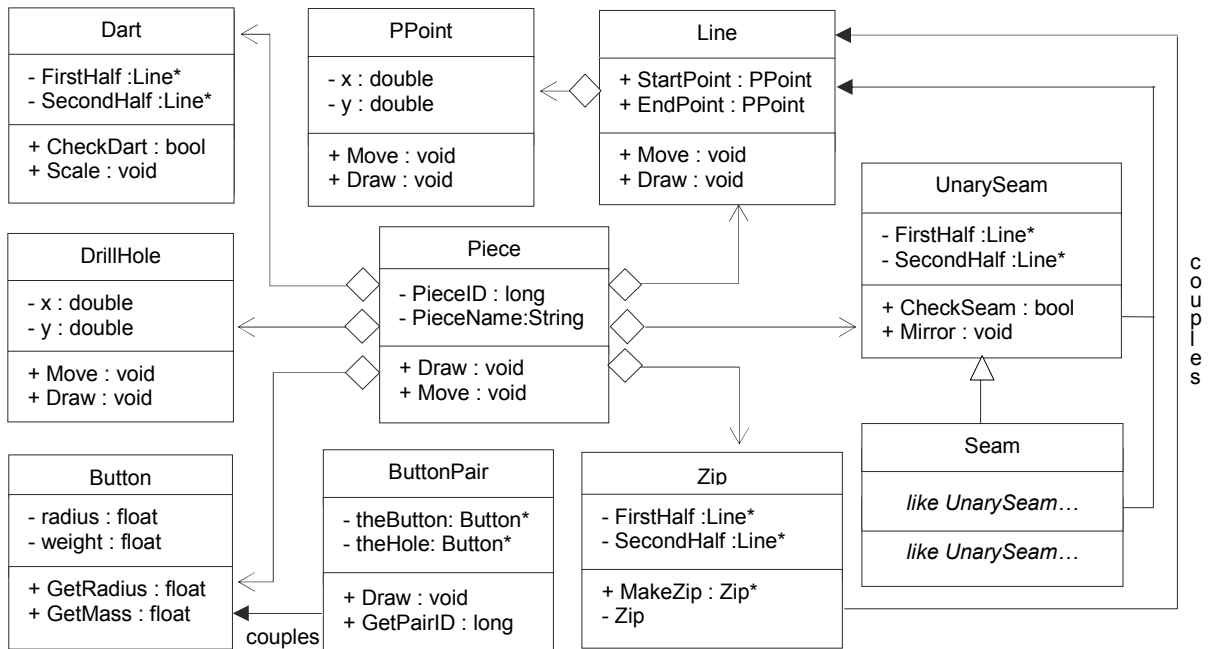


Fig. 9. UML diagram of tailoring classes.

5. TEST CASES

ClothAssembler was validated by testing the system on garment models of real interest and production. As a first test case, a T-shirt model was considered (Fig. 10, left), composed of four pieces and assembled by using ten seams, four of which to connect the front and the back pieces, and six to connect the sleeves to the main body. To make each seam, a user must click on the first and second vertex of each border to be sewn, choose the orientation and a sewing modality, i.e., by specifying if the mapping is built by node or notch correspondence or by looseness distribution; then, the sewing action is performed with information stored for the output. Seams between the front and back pieces were quite easy and intuitive to obtain, as they concern borders at the sides and at the shoulders that have equal or similar lengths between the front and back panels. Sewings between the sleeves and the main body are, on the contrary, particularly difficult operations in garment manufacturing as they connect borders with different lengths (in fact, the sleeve border is longer than the armpit girth): as a tailor manually fixes reference notches along the borders to proportionally distribute the looseness of the seam, the same was done – and easily – by ClothAssembler with automatic notch insertion.

Another, more complex, test case was a skirt (Fig. 10, right) composed of seven pieces, among which the thin waistband and two pockets. This case has required ten seams, two darts, two pocket insertions, and one zip. The lower horizontal side of the waistband, for instance, is sewn with the upper side; then, the entire length is sewn with the four skirt pieces. These are then sewn with each other along the vertical sides. Darts are added to cause a natural 3D effect to the skirt shape as the waist girth is smaller than the hip girth. The zip is located in the upper borders of the two front panels. Pockets require a careful evaluation of the internal polyline onto which they are sewn. ClothAssembler facilitates the pocket localization against the polyline by using reference drill holes (blue, in Fig. 10) along the pocket opening.

The virtual manufacturing of both the T-shirt and skirt required for an average user just few minutes, i.e., about 2'-3' for the T-shirt, and 5'-6' for the skirt. The user had just to be informed of the system functions, and to be a little familiar with basic notions of manual garment manufacturing.

Easy usability of the module and short times required to perform the virtual design/assembly actions encourage us to the validity of our approach. Now we have to extend ClothAssembler with additional functions, in order to use it for the tailoring of garments that are particularly elaborated such as jackets, coats, etc.

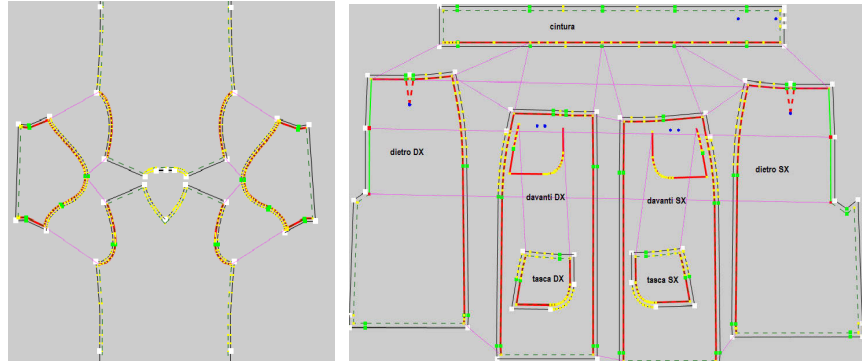


Fig. 10. Test cases: T-shirt and skirt, designed and assembled by ClothAssembler.

6. CONCLUSIONS

A graphical environment has been presented for the computer-aided design and assembly of cloth pieces for real garment manufacturing purposes. The system allows a user to interactively assign sewing and assembly rules between panels, and to insert functional details, such as pockets, reinforcements, pleats, cuts, darts, zips, constraints, etc. The virtual manufacturing process requires short times and can be applied to sufficiently complex apparel. Further tests will be done on more elaborated test cases, e.g., coats and jackets. New tailoring features will be implemented, e.g., virtual emulation of local treatment/deformation of materials (e.g., by ironing, starching, etc.), further seam and pocket types. The system will be integrated in a 3D environment that includes physics-based modelling and simulation of the assembled garment placed in a 3D configuration, to verify the final simulated effect of a garment that is entirely virtually prototyped throughout all 2D and 3D phases.

7. ACKNOWLEDGEMENTS

This work has been carried out in the framework of the national PRIN Project Vi-Cloth, funded by the Italian Ministry MIUR. The authors would like to thank the company F.K. Group Italy, and all colleagues from Università' di Bergamo, Politecnico di Milano, Università' di Firenze and Università' di Brescia, Italy, that are participating to the research project.

8. REFERENCES

- [1] Breen, D.E., Eds., Special issue on Computer graphics in Textiles and Apparel, *IEEE Computer Graphics*, Vol. 16, No. 5, 1996, pp 26-80.
- [2] Cambiaghi, D., Cascini, G., Gorla, C. and Rizzi, C., Vi-Cloth, Virtual Clothing, *Proc. XIV ADM - XXXIII AIAS*, Bari, Italia, 2004.
- [3] Carubelli, A., *ClothAssembler: un modulo per generare parametri per l'assemblaggio in 3D di capi di abbigliamento*, MSc. Thesis, Dipartimento di Ingegneria Informatica, Università' di Parma, 2004.
- [4] Cugini, U., Bordegoni, M., Rizzi, C., De Angelis, F. and Prati, M., Modelling and haptic interaction with non rigid materials, *Eurographics '99 State of the Art Reports*, 1999, pp 1-20.
- [5] Fontana, M., Rizzi, C. and Cugini, U., 3D Virtual Apparel Design for Industrial Applications, special issue on CAD methods for Garment Design, *Computer Aided Design*, Vol. 37, No. 6, 2005, pp 609-622.
- [6] Hauth, M., Eitzmuß, O., Eberhardt, B., Sarlette, R., Sattler, M., Daubert, K. and Kautz J., *Cloth Animation and Rendering*, Eurographics'02 Tutorial, 2002.
- [7] House, D.H., Breen, D.E., Eds., *Cloth modelling and Animation*, A K Peters Ltd, Natick MA, US, 2000.
- [8] Lectra Systèmes, <http://www.lectra.com> & Investronica Sistemas, <http://www.investronica-sis.es>. GerberTechnology, <http://www.gerbertechnology.com>. PAD System Technologies, <http://www.padsystem.com>. Koppermann Computersysteme, <http://www.koppermann.com>. Browzwear, <http://www.browzwear.com>. OptiTex, <http://www.optitex.com>. Digital Fashion, <http://www.dressingsim.com>. F.K. Group, <http://www.fkgroup.com>.
- [9] Kawabata, S., *The Standardization and Analysis of Hand Evaluation*, The Textile Machinery Society of Japan, Osaka, Japan, 1980.
- [10] Volino, P. and Magnenat-Thalmann, N., *Virtual Clothing: Theory and Practise*, Springer-Verlag, Heidelberg, Germany, 2000.