

## An algorithm based approach for the apprenticeship of Computer Aided Design (CAD) software

Professor Alain Desrochers  
Department of Mechanical Engineering  
Université de Sherbrooke  
Sherbrooke, J1K 2R1, Canada  
E-mail : alain.desrochers@gme.usherb.ca

### Introduction

Contemporary CAD systems are increasingly becoming easier to learn, simpler to use and more productive. Several reasons can be accounted for this. Among those are the development of sketch based approaches for the generation of 3D geometry, the construction of product specification trees and the use of familiar Windows® based interface, toolbars and context dependent menus.

However, at the same time, the application spectrum of CAD systems and digital mock up is widening, with the number of specialized functions increasing accordingly. Therefore, the learning challenge for the novice user is now to know *what* function, application or module to use rather than *how* to use them.

Unfortunately, traditional CAD training material are still highly structured as guided tutorial, focusing on sequences of interaction required to achieve a given result and providing great details on how to fill out dialog boxes in very specific instances.

The proposed approach aims at providing the user with a generic design methodology which is ultimately tied to the various features of a CAD software. The methodology is presented as a series of algorithms of increasing depth, leading to appropriate application modules and finally specific CAD functions. The details relating to the use of the functions themselves are left for the user to apprehend as they are often intuitive, given proper knowledge of CAD basic principles and theory. Moreover, the algorithms are viewed as a general front end to existing training tools, providing a more comprehensive view of the CAD software by exposing its logical structure.

### Training material provided by CAD systems suppliers

Before going on to presenting the proposed learning approach, we will proceed to enumerate the main features of most training material developed by CAD software manufacturers. Their characteristics will then be compared against learning tools requirements in engineering curricula.

As to meet the requirements of their key market, the CAD manufacturers have developed training materials which have been designed with the industrial user in mind. As mentioned in the introduction, these training material are structured as detailed tutorial to be closely followed and leading to the quick realization of parts, thus providing a feeling of productivity and easy learning curve. In addition to this, these training material also feature the following characteristics:

- The learning process requires minimal efforts from students and is almost independent of their technical backgrounds;
- Exercises provided in the training often require hours and sometimes days, even for basic tasks;
- This approach requires enormous amounts of documentation and data to cover a complete product and these have to be continuously updated;
- Students are typically each assigned a computer and classes generally feature a high instructor to student ratio;
- Understanding of the software structure and general principles follows rather than precedes its use in tutorials.

#### Apprenticeship of CAD software in engineering curricula

Learning CAD software in engineering curricula is usually not an end in itself. Indeed, CAD software apprenticeship alone is generally not being recognized as “units” by engineering accreditation boards in North America, unless it is coupled as laboratory to more traditional formal courses on CAD theory and principles. Also, the definition of units and the many objectives of CAD courses make it such that a limited number of hours can effectively be devoted to CAD software apprenticeship. At the same time, CAD systems have become an essential tool for creating digital products, a now common input source for an increasing number of computational and simulation engineering software. Because of that, the proper knowledge of CAD software use is therefore, nevertheless, essential.

Additionally, CAD systems constantly evolve, continuously requiring increasingly powerful computers to run. Typically, a CAD laboratory will be dedicated to them with costly computers featuring huge amounts of RAM memory, high performance graphical adapters and large screens. These high performance machines are moreover in great demand by students who are using them in an increasing variety of course and projects requiring work on the digital mock up. Therefore, time spent by students in front of these computers must be as productive as possible such as to maximize their availability to other students.

Clearly, these facts and constraints are not compatible with the features of traditional training material, as provided by the CAD manufacturers. For these reasons, a novel approach has been developed at the University of Sherbrooke to address this issue.

#### Requirements for a CAD apprenticeship aid

Given the context outlined above, several requirements and desirable features can be formulated for a CAD apprenticeship aid within an engineering curricula:

- Competencies objectives must be attained in less time (in front of the computer);
- Students must have the tools to become autonomous quickly in a context where there are more students per instructors than in industrial training;
- The apprenticeship aid must provide a broad and comprehensive view of the software capabilities even though the student will ultimately not be trained on all of these;
- The learning aid must also serve to inform on the limitations of the software while remaining as independent as possible from software continuous evolution;
- Finally, CAD systems apprenticeship tool must be targeted towards methodology as well as software use.

## Algorithm based apprenticeship

The proposed approach or solution to the above mentioned problems and requirements is the construction of a specific, algorithm based guide for the apprenticeship of a given CAD platform. The guide exposes the structure of the CAD system through a series of algorithms of increasing depth, thus providing a global picture of the software and a generic roadmap for users/students to navigate through.

Furthermore, the use of algorithms allows the natural embedding of elements of methodology for the user. This is particularly useful for modules such as that of finite elements, where particular specification sequences must be followed in the construction of a typical analysis case.

Also algorithms allow the expression of various types of relations between CAD software functions. For instance, the use of a function may exclude or not the simultaneous use of another one. Finally, a graphical approach was adopted for the representation of the algorithms, thus making the guide very easy and intuitive to consults. The algorithms are completed by additional explanations at the end of the guide when necessary.

## Structure of the guide

The guide is comprised of four levels of diagram, each with a specific purpose and referring to the other levels. The four levels are imbedded and bring the user from a general overview of the software, down to the level of the specific CAD functions. These levels and their particular features are presented in the following sections.

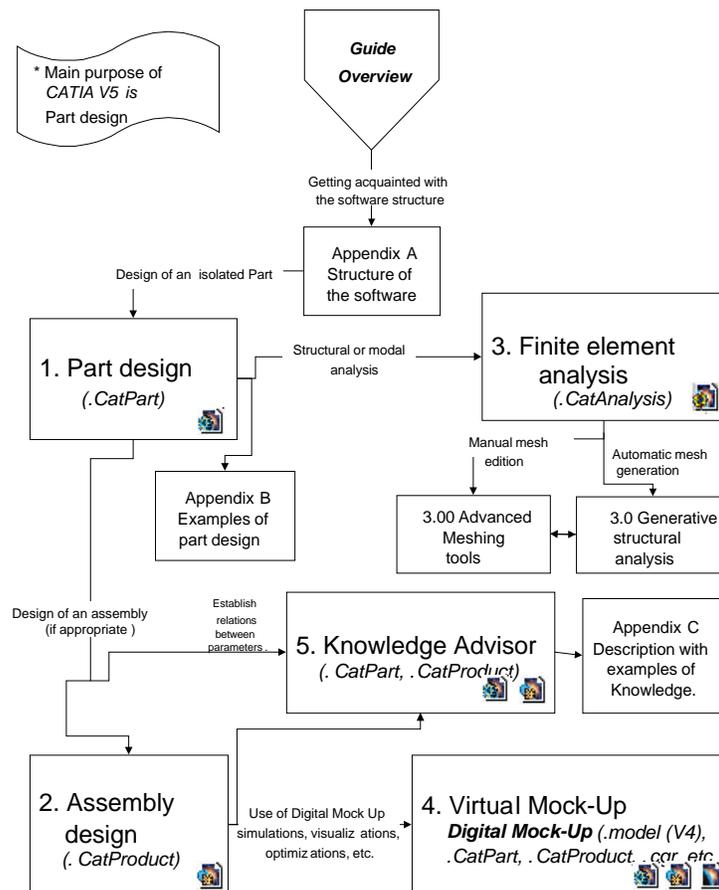


Figure 1: Algorithm at product and file structure level

Level 1 - Product and file structure level

This “entry” level provides details about the various type of information being processed and modeled and the corresponding files being handled. For instance, in Catia V5, for which the guide was constructed, an assembly is referred to as a product for which there is a corresponding \*.CatProduct file. A part, on the other hand, will be stored into a \*.CatPart file.

These two types of files can further be associated to various “container objects” such as a “component”, which may contain one or several parts, products, components and sub-components. A sub-component, in turn, is a container for a part, that includes links to given product files. Also, a part may be constituted of bodies for solid elements, and open bodies for surfaces. These definitions and relations are presented in appendix A of the guide.

However, the real basis for this level lies in the fact that the various design and application modules are grouped according to their common purpose or the file type(s) that they use to store their results. This is presented in figure 1 where five groups or applications are being formed :

- > Group 1 – Part design (files \*.CatPart)
- > Group 2 – Assembly design (files \*.CatProduct)
- > Group 3 – Part analysis (files \*.CatAnalysis)
- > Group 4 – Digital Mock-up (\*.model (V4), \*.CatPart, \*.CatProduct, \*.cgr, etc.)
- > Group 5 – Knowledge Advisor (\*.CatPart, \*.CatProduct)

1. Part design

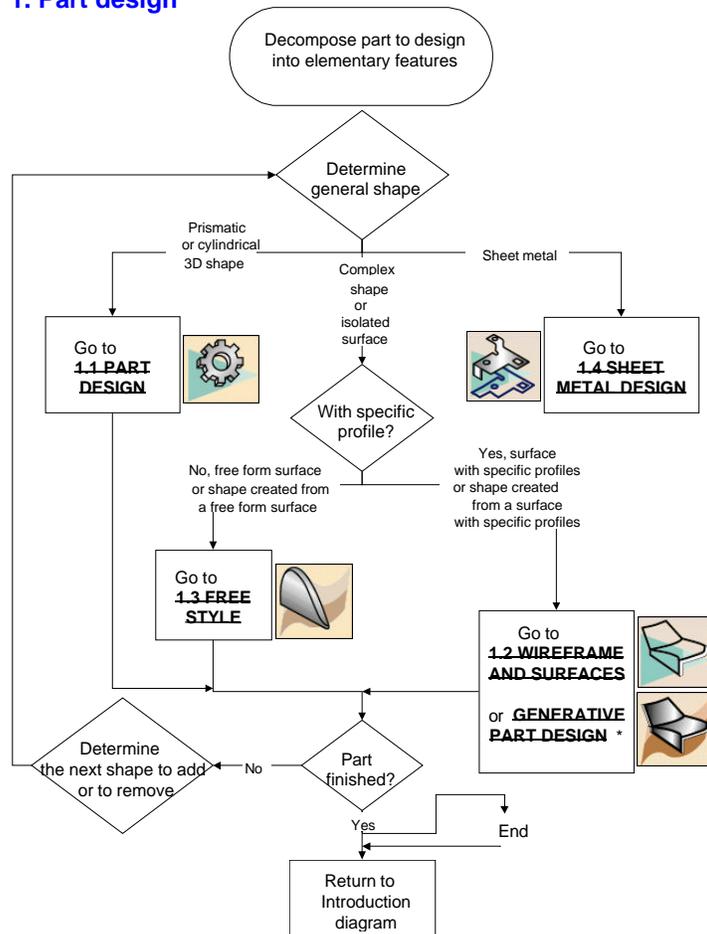


Figure 2: Part creation algorithm at application module level

## Level 2 - Application module level

The goal of this level is to discriminate between modules belonging to the same group, or in other words, associated to the same type of information. More precisely, the algorithm at this level will be associating a specific modeling or analysis duty to the various modules belonging to the same group at the product and file structure level.

For example, in Group 1 – Part Design, at the product and file structure level, there are again five modules that are being characterized in four classes according to their own specific modeling purpose (figure 2) :

- > Class 1.1 – Part design, for 3-D prismatic or cylindrical parts
- > Class 1.2 – Wireframe and Surfaces or Generative Shape Design, for curves and surfaces created from curves.
- > Class 1.3 – Freestyle, for freeform surfaces created from arrays of control points.
- > Class 1.4 – Sheet Metal Design, for part created from sheet metal with straight folds.

There are, however, other more specialized modules that have not been included in the guide. Indeed, the guide is already 150 pages long and these modules are too specialized for our purpose. Among those are the piping module, structure module, etc.

### 1.1 Prismatic shape

#### 1.1.1 Creation of prismatic shape

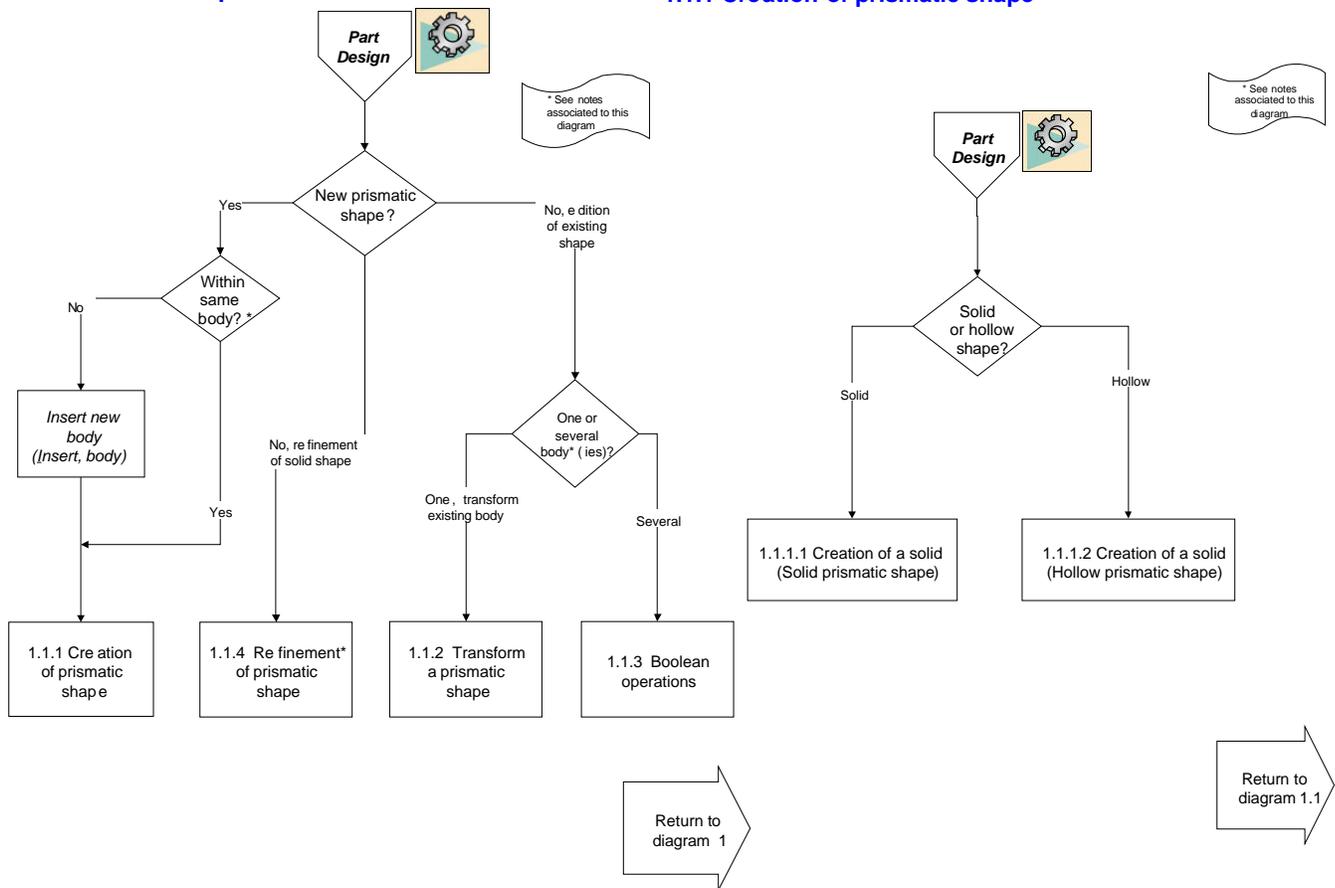


Figure 3: Prismatic part creation algorithms at task level

### Level 3 - Task level

Once in a module, users are guided towards the proper group of functions according to the type of task they are undertaking. In that respect, task classification reflects, in part, the evolution of the model (creation, refinement, transformation, ..). Task classification was performed using a reverse engineering approach and is therefore not necessarily unique, entailing that various interpretation would be equally valid at this level. For instance, the chamfer function has been associated to shape refinement rather than creation and the split function to shape transformation rather than creation.

The task level can also be seen as an intermediate level between modules and functions which are both entities precisely defined in Catia and each having their own graphic icon. In clear, the purpose of this level is to associate a certain order and methodology to the various functions by regrouping them accordingly.

Following our example, with Class 1.1 – Part design module, as defined in the preceding level, six algorithms have been developed with various sub-levels of details (figure 3) :

- > Algorithm 1.1.1 – Creation of a prismatic shape
- > Algorithm 1.1.1.1 – Creation of a solid prismatic shape
- > Algorithm 1.1.1.2 – Creation of a hollow prismatic shape
- > Algorithm 1.1.2 – Transformation of a prismatic shape
- > Algorithm 1.1.3 – Boolean operations
- > Algorithm 1.1.4 – Refinement of a prismatic shape

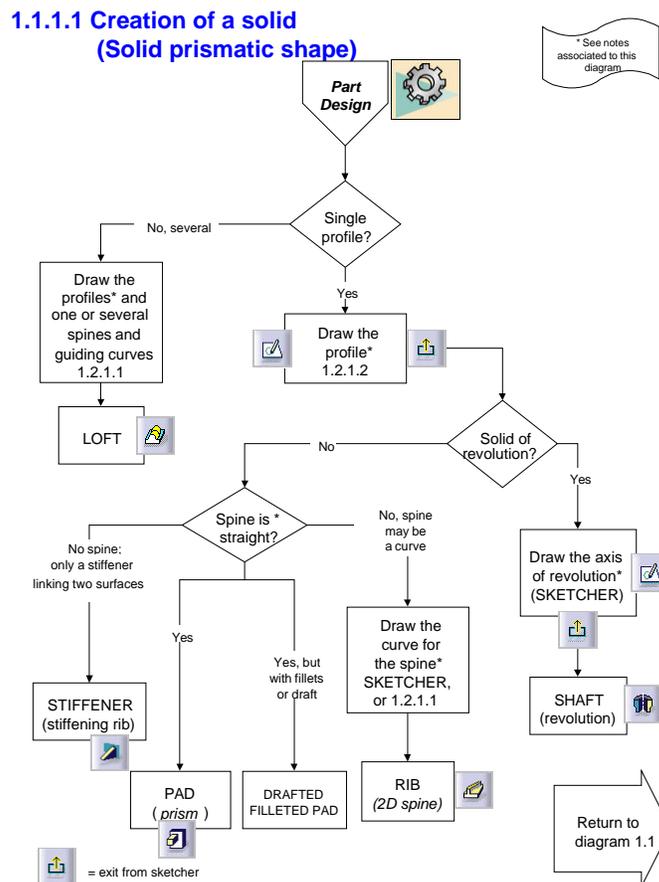


Figure 4: Solid prismatic part creation algorithm at function level

## Level 4 - Function level

This is the last and most detailed level of the guide. It allows the enumeration and differentiation of the various functions belonging to a common task group. As such, it covers all the functions of the different application modules referred to in the guide. However, this level does not cover the details pertaining to the use of each function, for instance how to fill out the dialog boxes, what are the available options, etc.

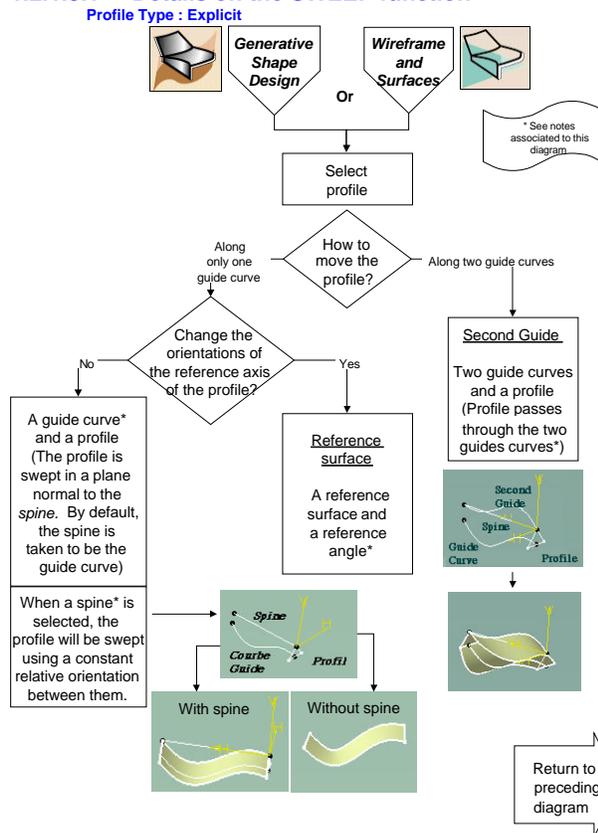
Again, taking an example from the preceding level, the algorithm 1.1.1.1 is found to be referring to the following module and functions : the Sketcher module and the Pad, Drafted Filleted Pad, Stiffener, Rib, Shaft and Loft functions of the Part Design module (figure 4).

An interesting and important feature of our approach lies in the methodological elements it includes. For instance, in the algorithm of figure 4, one will understand that it has to refer to algorithm 1.2.1.1 to generate the spine or guiding curves needed before using the Loft function or call the Sketcher module and use it prior to being able to use the other functions of the Part Design module.

### Relation to other training material

A complete guide would have included a fifth level called “Option level” since the various functions can be specified with a varying number of options. These options and choices are most often quite intuitive given proper CAD background, normally provided in formal courses. These options are moreover normally well documented in the on-line help or within the various training material provided with the software.

#### 1.2.1.3.1 Details on the SWEEP function



#### 1.2.1.3.2 Details on the SWEEP function

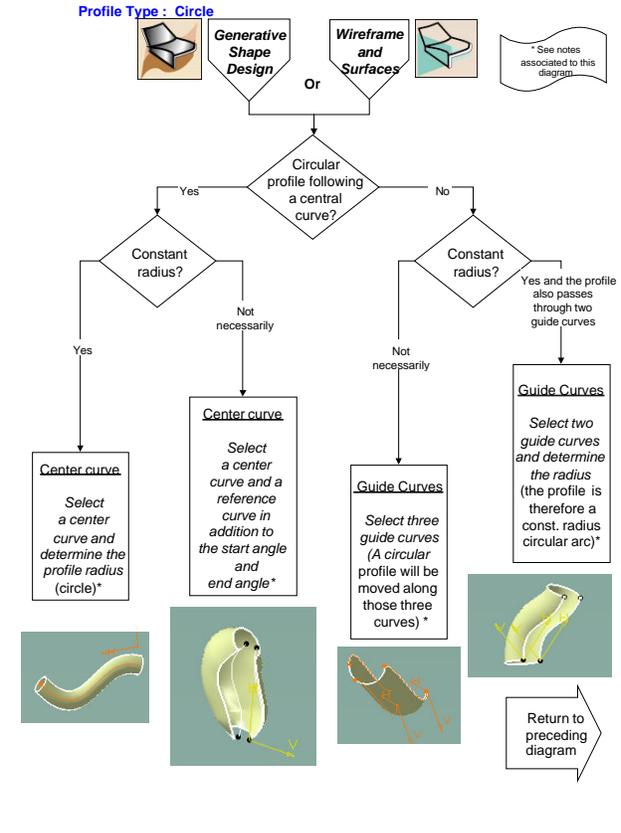


Figure 5: Algorithms (2 of 3) at option level for the SWEEP function

Integrating this fifth level in the guide was therefore beyond its intended scope and also a colossal work to undertake. However, an exception was made for the Sweep function of the Generative Shape Design module (figure 5). This function is indeed very powerful and versatile and the algorithm approach proved to be valuable to help grasp its full potential.

## Use of the guide

As will be shown in the following sections, the guide can be used in many different ways to assist in the apprenticeship of CAD software at the various levels.

### 1 – Introduction to the CAD system structure

As it is currently structured, the guide provides a complete overview of the software prior to using it. In fact, it proves to be especially useful to address the first two levels that have been presented above (product and file structure level and application module level) which are usually not covered in traditional training material. It therefore ensures that students have a comprehensive vision of the CAD system structure right from the start. This is indeed, the first intended use of the guide and this should be done in class, before students are left in front of a computer, with the CAD software, for the first time.

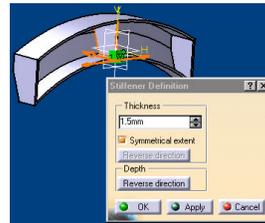
### 2 – A methodology roadmap

As was shown in the preceding sections, the guide obeys to a numbering system which allows the various diagrams to refer to one another. Moreover, algorithms can generally be used recursively until all the necessary functions for a given modeling tasks have been performed. Consequently, the guide allow the formulation of specific or preferred design methodology by providing the corresponding sequence of algorithm numbers and functions used. In that perspective, several modeling examples have been created by providing their own specific sequence of algorithms and functions as shown partly in figure 6.

#### Modify the part,

(i.e. Split the part using the surface Sweep.1)

- 1.1 Prismatic shape (Part Design module)
  - 1.1.1 Transformation of a prismatic shape
  - 1.1.1 Apply the SPLIT function using Sweep.1 as the cutting element



#### Create a shell from the solid part

- 1.1.2 Refinement of a prismatic shape
  - 1.1.2 Apply the SHELL function
- Return to diagram 1.1

#### Creating the internal walls

- 1.1.1 Creation of a prismatic shape
  - 1.1.1.1 Creation of a solid prismatic shape
  - 1.1.1.1 and 1.2.1.2 Create the profile for the wall (SKETCH in the *zx* plane)
  - 1.1.1.1 Apply the STIFFENER function using a thickness of 1.5mm (using the *symmetrical extent* option)

Replicate the stiffener

- 1.1.2 Transformation of a solid prismatic shape
  - Apply the RECTANGULAR PATTERN function, (Select *stiffener.1* (Object to Pattern) , the plane *xy* as the reference element, check the "keep specifications" option, specify two instances and a spacing of 50 mm)

Repeat the ARRAY function in the other direction since it cannot be used symmetrically.

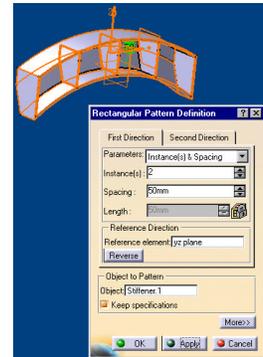


Figure 6: Algorithm based tutorial for aircraft part (excerpt)

Globally, the guide provides a bridge between a conceptual design methodology and the tools provided in the CAD system to achieve it. It is therefore most useful at the last two levels; the task and function levels. The general principle of nested algorithms of increasing depth is also practical and efficient by allowing a user to focus on those specific tasks that appear to be more difficult to her or him.

Finally, custom algorithms could be developed in an industrial context to guide designers by providing them with graphical representations of standard part creation methodologies. Such an approach was viewed favorably by the CAD specialists at Bombardier Aerospace, to whom the guide has been presented and who are also responsible for providing the designers with aircraft standard part creation procedures.

### 3 – Quick visual aid

Even though it has not initially been constructed for that purpose, the guide was found to be useful as a quick visual aid to provide information on specific functions or to highlight the differences between them. One way of exploiting the guide is therefore to use its algorithms backwards, to find the difference between two similar functions. Algorithms would then be followed from the functions up to their common intersection which should then indicate the difference between them. Figure 7 illustrate a case where such an approach is employed to seek the distinction between the “symmetry” and “mirror” functions, both belonging to the part design module. One can see, by moving up the diagram to their common node, that the mirror function replicates the elements whereas the symmetry function only transforms or displaces them.

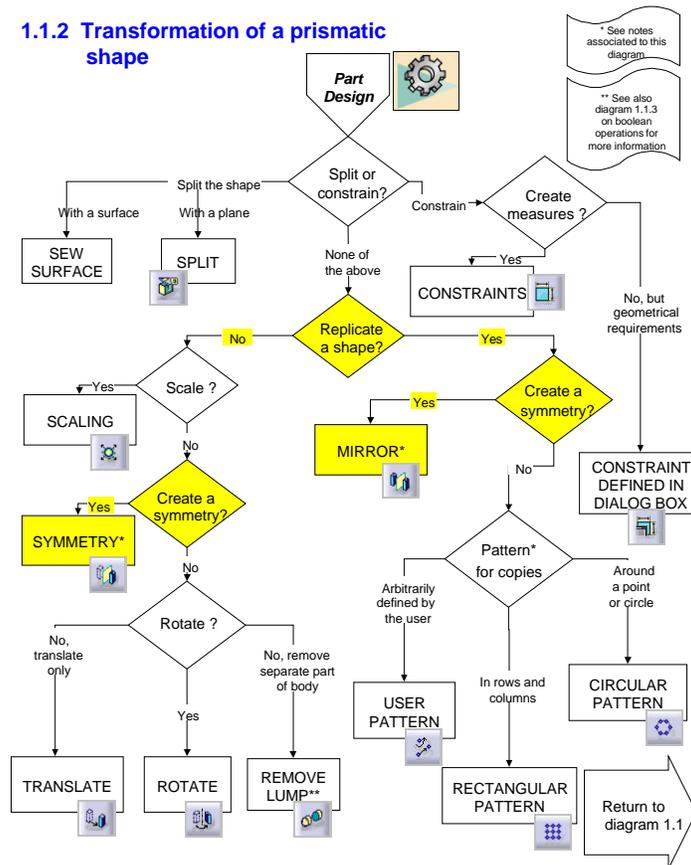


Figure 7: Use of algorithm for function discrimination purpose

Obviously, this example emphasizes the fact that the whole approach is based on the distinction between application modules and functions rather than simply on what they do. This, incidentally, proved to be an important challenge to undertake when constructing the guide. Of course, this last approach for using the guide is targeted at the fourth level, that is the function level.

Finally, the use of the guide as a quick visual aid was found to be extremely useful for specialists or professors who have a good global understanding of the software but who are not everyday users of the product or who might not be so familiar with the more specialized application modules. This fact was confirmed by the author as well as by the Bombardier CAD specialists to whom copies of the guide were distributed.

### Conclusion

The paper attempted to present a novel approach from global to detail for the apprenticeship of CAD software in an undergraduate mechanical engineering curricula. In addition to the description provided in the preceding sections, the approach features interesting characteristics but also some drawbacks which will be outlined as concluding remarks.

### Additional features of the method

On a practical basis the guide features the following additional characteristics which, in our view, makes it even more interesting :

- The proposed learning tool is less prone to update lead modifications than regular tutorial based training material. This especially true for the first three levels of the guide which vary very little with the various releases of the Catia Version 5 CAD software for which the guide was constructed;
- The guide contains no, or very few, long sentences but instead, short descriptive texts in the various algorithms boxes. It is therefore easily translatable into other languages which is rather practical since the only currently available version of guide is in French;
- Since the guide was constructed as a series of algorithms with questions and choices, it can consequently readily be programmed as an on-line companion (such as the “paper clip” in Microsoft Application Software).

### Drawbacks of the approach

However good, the approach nevertheless features several drawbacks which have to be weighted against its benefits:

- The guide requires increasing efforts to maintain, trying to keep up with the endless new releases of the software. Moreover, in addition to requiring continuous work, the type of financing usually available for professors is not targeted towards the maintenance of such guides;
- Effective use of the guide requires prior basic knowledge of CAD systems, especially when creating new parts. Indeed, one must first decompose a given part into a succession of basic element creation and operations. Once this has been done, the guide can efficiently be utilized to convert the generic construction sequence into the proper series of function calls specific to the CAD software;
- However useful in class to provide a good overview of the CAD software and its various modules, the students tend not to use the guide so much in laboratory, relying more on the assistant help or the documents provided to them to accomplish their work.

### Further work

The nature of the guide make it such that ongoing efforts should be deployed to keep it updated. However, as the core of the software stabilizes with time, the author expects that less efforts should be deployed in the future to maintain the guide as it is, that is without increasing the number of modules it is currently covering. The modules already included in the guide are those that are general purpose and closely related to the construction of parts.

Additional work could however be done to better integrate the use of the guide with the laboratories. Quite simply, the students should be asked to prepare their laboratory in advance, supplying the sequence of algorithms that would allow them to perform the required task. That way, the students would be more productive in front of the computer and would also make better use of the laboratory assistant, asking him more relevant questions.

### Acknowledgments

The work presented lead to the edition of an internal generic user guide for the CAD software Catia V5 release 3, later updated to release 6. It has been realized by two sophomore students, Édith Talbot and Simon Héroux during summer work terms in 2000 and 2001 respectively, using a reverse engineering approach.

The work was supported by two NSERC undergraduate research fellowships completed by an NSERC general grant.

### Biographical information

ALAIN DESROCHERS obtained his M.E. degree from École Polytechnique de Montréal (Canada) in 1984, his M.Sc. from the University of California, Los Angeles in Control, Systems and Dynamics in 1985 and his Doctoral degree from École Centrale Paris (France) in 1991 in Computer Aided Tolerancing. He is currently holder of the Bombardier Research Chair in Design and Modeling of Mechanical Systems and Complex Structures.