4

Configuration Systems
Architecture

When the product configuration tasks increase in complexity, traditional approaches show serious deficiencies. This chapter illustrates how modern configuration systems overcome the difficulties of product configuration. In particular the chapter:

- describes modern configuration software architecture, pointing out its fundamental components and functions;
- illustrates the concept of a configuration system, considered as a set of technical and human resources;
- discusses different alternatives for the definition of a configuration system;
- delimits the domain where product configurators are applicable, distinguishing them from selectors and meta-configurators.

4.1 Product configurators architecture

A company that offers customization via configurable products is able dynamically to adapt its products to a wide range of customer requirements, without carrying out design activities. This approach differs from pure customization, where the product variants are designed ‘to order’, and from variety without customization, where operational processes are not tied to order acquisition processes. The possibility of ‘customizing without design’ is connected to the concept of ‘product configurability’, i.e. the possibility of offering a complete description of product family architecture. This means that:
the company has defined ‘a priori’ all the possible variants of all the possible components used in a product family, and/or all the ‘rules’ according to which the components variants are determined;

product functions and purposes – and therefore their characteristics – have been associated with single component variants or with their combinations.

The first principle establishes that the elaboration of any product variant does not require new components – or if it actually needs some, the description of such components is obtained by using algorithm instructions. The second principle establishes that it is possible to associate, without uncertainties, the functions and purposes required by the customer with a given set of components. The final result, when defining the architecture of a product family, consists in the fact that the company has virtually eliminated any kind of uncertainty about the product characteristics. But, in the configuration process, this elimination is not enough to guarantee a fast and correct output. Configuration tasks, in fact, are characterized by little or no uncertainties and they may also be characterized by high complexity. This complexity in the configuration tasks derives from the fact that a great deal of information must be processed: checks, translation of customer needs into product commercial characteristics and vice versa, verifications of completeness and congruence of the specified commercial characteristics, etc. On the one hand, the definition of product architecture makes the configuration process logically possible, without involving the technical office. On the other hand, it is also necessary to define instruments to support configuration that can reduce computing complexity, giving the salesman – and even the customer – the conditions to operate autonomously in a fast and correct configuration process. From this point of view, let us consider again the case of a suit. The form ‘reminds’ the shop assistant of all the parameters that are specified in the customization of the product. Diagrams and double entry tables give some simple information to the salesman, who has to specify the values of the parameters written on tables and diagrams and point out the information associated with their intersections.

The structured approaches described in the previous chapter can be considered as attempts to reach a formal, even if partial, description of product architecture. For example, the form used for personalizing a suit implicitly describes all the parameters necessary to adapt the garment to a specific customer. This form supplies a commercial description of the suit. Diagrams and double entry tables, in contrast,