Chapter 6
Brief and Design

ABSTRACT

This chapter is a general introduction to the connections between brief and design. First, the two basic forms of this connection are considered, feedforward (the transfer of brief information to design representations) and feedback (where the connection is used to evaluate brief satisfaction in the design and subsequently to improve either the design or the brief). Then the attention turns to the prerequisites for design representations: the structures that allow direct recognition of relevant entities and the enrichment of these entities with external information, in this case, from the brief. Finally, these prerequisites are examined in more detail in CAD software (AutoCAD) and BIM editors (Revit and AutoCAD Architecture).

FEEDBACK AND FEEDFORWARD

Linking a brief to designs made for it is an evident necessity for two related yet distinct purposes. The first is the evaluation of a design with respect to the satisfaction of brief requirements. Brief compliance is a primary measure of design suitability and acceptability. The second purpose is the analysis of the brief itself. What a design proposes and achieves can be powerful commentary on the brief’s intentions, structure, and content. It can help us improve and refine the brief in the same or subsequent projects. Consequently, a link between brief and design should support feedback both from the brief to the design and from the design to the brief. It is therefore important that feedback concerns all possible levels of abstraction: from general aspects of a design and brief goals to the accommodation of specific activities and their particular requirements.

The basis for most levels is the fundamental connection between activities and their requirements in the brief and the entities that accommodate or facilitate activities in the design. Preferably this should be a permanent, dynamic process that allows us to compare the two and improve either, if and when necessary. As we have already seen, on the brief side DBMSs offer sufficient support for flexible and efficient adaptation. It is widely known that CAD and BIM programs can do the
same for designs. Less familiar are ways of connecting the two sides indirectly (e.g., by exporting and importing data) or directly (e.g., via dynamic connections that propagate changes in both directions) – similarly to the connection between graphs and databases in Chapter 4. Chapter 8 describes how you can establish such connections and compare the two.

In addition to feedback we can also have feedforward from the brief to the design. The purpose of feedforward is to establish a design framework on the basis of brief requirements and constraints, an informative starting point for solutions to the problems posed in the brief. CAD and BIM programs allow extensive import of brief information. For example, you can predefine spatial objects on the basis of programmatic requirements and use these objects to develop a design in the same way you would use predefined building elements in BIM or CAD libraries. Alternatively you can develop a conceptual design and link its objects to records in the brief database in order to have access to the full spectrum of requirements and refine the design accordingly. Feedforward and the development of designs on the basis of brief requirements is the subject of Chapter 7.

Another example of feedforward is the use of constraints to define a framework that regulates the behaviour of design objects. Building regulations, for instance, can be used to define a permissible building envelope within which the overall volume of the design must lay. Such applications of constraints in briefing, in the relationship between briefs and designs, as well as the support offered by CAD and BIM software for the integration of constraints in design representations are described in Chapter 9. The reason for putting constraints separately is that in the context of the present book they combine feedback and feedforward: Constraints both predefined and regulate the behaviour of design objects.

**DESIGN REPRESENTATION**

Successful connection of a brief to a design depends on two prerequisites, the first being the structure of the brief. A brief that analyses goals, constraints, and requirements into specific and concrete items facilitates connection to design properties and aspects. Fire security, for instance, is a general goal but to make it operational we have to distinguish between goals and criteria for egress and compartmentation, constraints for door and corridor width, and so forth, and organize them analytically, as discussed in Chapter 2. The intention behind this is to develop a comprehensive, coherent, and consistent information system, but also ensure that each goal and especially each constraint and requirement can be explicitly linked to design objects, their properties, and relationships.

Consequently, the second prerequisite is a design representation that makes exactly these objects, properties, and relationships explicit and linkable to the brief. Developing a design representation in terms of such objects has recently become popular with BIM, but it has long been possible in CAD too. However, as the main purpose of digital architectural representations has been the production of conventional drawings on paper, the majority of CAD users just replicated analogue practices and merely drew lines in any fashion possible. The only thing that seemed to matter was that the result looked like and printed an analogue drawing. While production of analogue documents unfortunately still remains a major preoccupation, an increasing number of software developers as well as users appear to realize the importance of information processing in synthesis, analysis, and communication. In turn, this leads to the realization that information and processes that rely on it (e.g., integration) require meaningful, efficient containers. Despite the complexity of architectural thinking, the basic principle of such meaningful representations is rather simple: Anything we perceive and handle as a design entity should be represented by a single, discrete object (or group).