Computer tools that provide support in the early explorative phase of architectural design are rare. In order to explore the solution space of a design problem, architects in general rely on the use of paper and pencil. We are developing a system which will assist the designer in the exploration of a particular design problem, focusing initially on 2D floorplans. The format chosen for this system is one in which the computer, in interaction with the designer, “evolves” designs. A major obstacle associated with the use of an evolutionary approach is the adequate representation of a floorplan in a genome. We propose a tree-structure in which the nodes represent organising-principles that dictate how the leaves attached to it are organised. Figure 1 shows an example of a tree with four levels; the terminal nodes represent the elements to be placed. The first level no only represents an organising-principle, but also a contour; this contour is either fixed (e.g. dictated by the environment) or free to evolve. On either side of the tree-structure two interpretations of this tree are shown.
They differ only in the organising-principle at the top level. The interpretation on the left applies a “fishbone” or slicing organisation. In the interpretation on the right, the elements on the first level are organised around a central space. Defining and implementing more of these organising principles, makes this representation very extensible and flexible.

Figure 1.

The evolutionary process is guided by the evaluations made by the computer and the designer. The computer evaluates the designs on objective- and fuzzy criteria. Fuzzy criteria could be evaluated by means of a neural network. The subjective criteria, or criteria that can not be otherwise evaluated, are evaluated by the designer. The general structure of the system is shown in figure 2.

Figure 2.

During the evolutionary process, there are interaction points at which the user is presented a number of alternative floorplans which have a high score on the objective and fuzzy criteria. The user can then alter (augment or decrease) these fitness values, thereby steering the process in the desired direction. Given the same input, the system is able to generate a different solution every time it is run; the random starting conditions make it very likely that the process will converge on another solution with approximate equal fitness.

The working method associated with the system is by no means restricted to the generation of floorplans. An obvious extension would be the development of 3D organising principles in the tree representation. By extending the system to a 3D environment, it could also be applied in, for example, an industrial design setting.
Up to now, we have developed the computational architecture and the algorithms. The next step is the actual implementation in C for Windows 98. Afterwards the efficiency and effectiveness of the tool should be measured in the laboratory as well as in the practice of the building and urban design office. The proposed software does not impose a working style to the designer, who is free to use it in any specific designing strategy, based on the emphasis given to the Concept, the Programme and the Site (see also Van Bakel, 1995). Because the software does not claim expert sketching, drawing or modelling skills, it is especially apt for Computer Supported Cooperative Designing (CSCD) with non-designers like clients and users. Moreover the program allows dynamic planning making use of a strategy of broadening the solutions field repeatedly. This strategy is aimed at creating a series of alternatives instead of the usual exploration in depth of only one alternative. But because the design approach proposed here is build up from the content to the shell (from lay-out to volume) it still needs a complement with an appearance oriented software package (like Xfrog) working from the shell to the content.

Reference