Revitalisation of Existing Buildings through Sustainable Non-Destructive Floor Space Relocation

Thorsten M. Lömker[1]

ABSTRACT
The revitalisation of existing buildings is getting more and more important. We are facing a situation where in many cases there is no need to design new buildings because an increasing number of existing buildings is not used anymore. The most ecological procedure to revitalise these buildings would be through a continuous usage and by making few or no alterations to the stock. Thus, the modus operandi could be named a "non-destructive" approach. From the architects' point of view, non-destructive redesign of existing buildings is time-consuming and complex. The methodology we developed to aid architects in solving such tasks is based on exchanging or swapping utilisation of specific rooms to converge in a design solution. With the aid of mathematical rules, which will be executed by the use of a computer, solutions to floor space relocation problems will be generated. Provided that "design" is in principle a combinatorial problem, i.e., a constraint-based search for an overall optimal solution of a problem, an exemplary method will be described to solve such problems.

KEYWORDS
Revitalisation, Optimisation, Floor Space Relocation, Constraint Programming

INTRODUCTION
The renewal, reuse, and revitalisation of existing building stock is a process that could be observed throughout the entire history of private and public buildings. Reuse has taken place in all historical periods and all sectors of society and culture. The reasons leading to reuse have been of great variety. Obsolete structures, changed usage requirements, or even the transformation of aesthetic demands and fundamental positions caused owners of buildings to undertake conversions. These factors leading to reuse have hardly changed to the present day. However, new factors have appeared that necessitate the use of existing building stock under different conditions. More than ever before, economic and ecological parameters play a weighty role in the area of renovation and revitalisation of buildings. Sustainable use of our built environment has become an important subject in society and architecture. The architect is increasingly challenged to think and act in an interdisciplinary manner and to effect a complete solution in cooperation with specialists. Spanning several fields of study, such efforts require strategies and abilities that quite often cause uncertainty in the planning process. In contrast to planning of new building projects, the basic conditions for revitalisation are regularly vague and difficult to calculate. However, successful planning depends upon exact knowledge and precise definition of goals and procedures.

There are a number of factors that justify planning but, at the same time, do not necessarily justify building activities in existing stock. An essential starting point is the discussion of sustainability of building stock. Reuse of existing assets is undoubtedly the most ecological and often the most economical way of building. It can be understood as a strategy for conserving the value of the building, considering the building's life cycle an integral part of planning. Its goal is to make organisational changes to unused building stock in such a way as to be reusable, while performing no or very little structural modifications. Thus, the reuse of a building can be termed "Floor Space Relocation Strategy." In order to present a successful alternative to new buildings, this strategy strongly depend on the architect, when beginning to plan, to come to a conclusion as to whether a building can be used applying this strategy. This decision is made by comparing the desired state (room program) of the building with its current state.

[1] Associate Professor, Department of Computer Application in Architecture and Landscape Architecture, Technical University of Dresden, 01069 Dresden, Germany. Phone: +49 351 463 39653, Fax: +49 351 463 36120, thorsten.loemker@tu-dresden.de

METHODOLOGY

The comparison of the room program with the floor plans of a building essentially is a combinatorial problem. In this research project, it was examined whether solutions for reuse tasks can be produced automatically by the use of optimisation processes in floor plan design. These solutions shall be produced by swapping of existing areas. The objective is to obtain feasible planning solutions by means of these computer-based processes, which will serve the architect as a basis for the further editing of the plans. The methodical basis for this procedure is formed by models from Operations Research (Liggett, 2000). The design of the model developed relates to problems in logistics, for example, the loading in trans-shipment centres. It also has analogies to board games like Chess or Go. These games are based on a specific number of fields or crosses of grid lines which are occupied by various tokens. Occupation is subject to a variety of conditions or rules. Compliance to conditions and objectives is clearly defined by the use of these rules. The analogy to our model is the fixed grid, the limited possibility to occupy fields and the fulfilment of an overall goal, i.e., to win the game. Therefore the model does not alter geometric proportions or locations of rooms but changes their occupancy such that a new usage could be applied to the building. Spaces are defined by means of graphs, where nodes correspond to the rooms of the building and edges represent their neighbour relation.

SYSTEM CONCEPT

The Non-Destructive Model is based on the inherent structures of the existent building. These are extracted solely from the organisational decomposition of the building into individual rooms, which are represented in the model as areas. It is possible, albeit not desirable, to apply the model to even smaller units, for example, areas decomposed by a grid. In this way, the potential solution space of a problem, i.e. the amount of structural actions to be undertaken, can be precisely controlled. This model aims at finding areas in the existent floor plans that satisfy the requirements of the room plan. This is achieved by comparing the properties of the existing areas with the properties of the areas in the room plan. No structural alterations are performed on the existing building. This is possible because the Non-Destructive Model does not change the geometric shape of a room but merely its use profile.

Input Model

Data on the properties of individual areas within the entire model form the foundation for the optimisation model. Among these properties are details on the location of a unit, its size, or its neighbouring units. Since room programs usually do not contain data on relations, i.e., location and neighbours, assigning these properties is a task of the architect and an essential part of the design process. The Non-Destructive Model utilises both data on units taken from the room program as well as data on existing units from the current floor plan. The relation applied to units of the existing floor plan is

- *unit i - is adjacent to - unit j*

Using this relation, the existent floor plan constellation is described in the form of a graph.

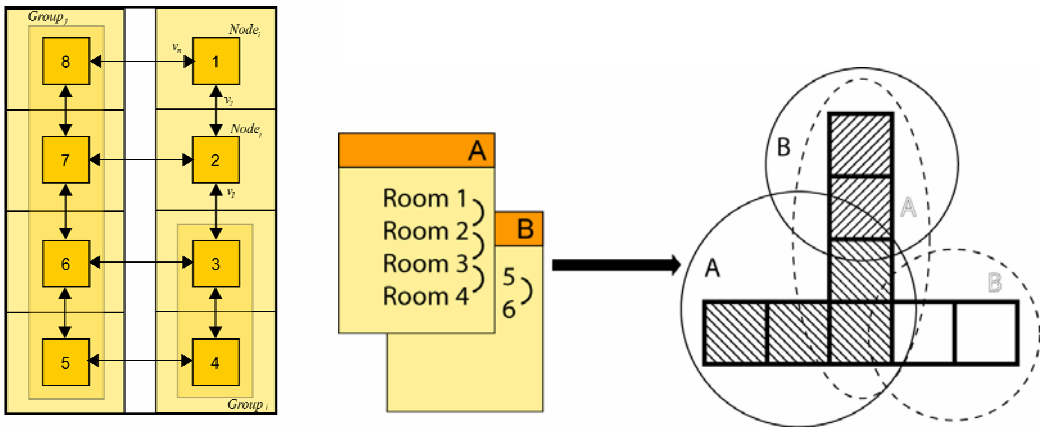

Figure 1: Graph representation of a floor plan; Methodology of the non-destructive model

Geometry Model

The model focuses on the geometric shapes of existing room limits or the orientation of system lines, such as those of the static system or the grid units of structural components. This abstraction of floor plans ensures non-invasive treatment of existing buildings. Units mapped by the Non-Destructive Model are not limited in their geometric shape. As nodes of a graph they can represent any shape, including non-rectangular shapes. The positions of its units are constant.

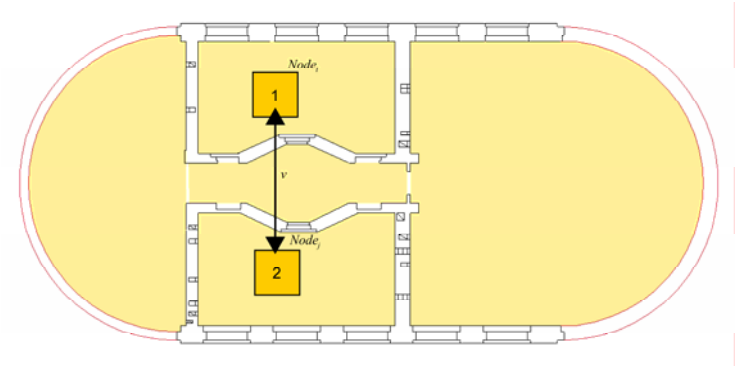

Figure 2: Definition of non-rectangular floor spaces

Variables

The most important variable in the optimisation process of this model is formed by an adjacency matrix. This matrix describes the adjacency between units of the existent floor plan. The manner in which this relation is defined may be modified and can be designated by the architect according to the specific project requirements. Adjacency is not required to be understood as immediate adjacency to another unit. Units that are opposite each other but touch a mutual transport space may be considered neighbours. Each unit is represented by a node in the graph. The graph's nodes are searched for all groups from the room program that are formed by several units. If no solution is found, auxiliary conditions represented by the following variables can be modified. This modification correlates with an adjustment of the room program. In terms of sustainability, this is the preferred procedure:

- *nbreOfGroups* - number of groups of connected units in room program to be found
- *sizeGroups* - size of groups of connected units in room program, in square meters
- *groups* - number of units within a group

Another, but less ecological, option for finding a solution is to modify the adjacency matrix. The matrix is enlarged by additional structural or organisational decomposition of the floor plan into smaller units, creating new room constellations. This procedure requires adjusting the following variables:

- *A* - adjacency of existent units in floor plan
- *nbreOfNodes* - number of existent units in floor plan
- *sizeNodes* - size of existent areas in floor plan, in square meters

An extension of the adjacency matrix does not inevitably result in a structural modification of the floor plan. Even so, this very ecological procedure may require larger distances between neighbouring nodes. If the number and therefore the size of units are altered, however, structural changes of the building are a practical consequence. In many cases, this is not as much of a problem in ecological terms as classic conversion since interventions are manageable if the matrix is incrementally modified. Removal or addition of walls alter the adjacency of units and lead to potentially greater flexibility for the process of swapping units. The optimisation process results in new values being assigned to the variables. These values

correspond to membership in a group whose name (depending on the number of groups) is assigned to the variable *label*:

- *label* - names of groups formed from room program

Constants
The model does not contain constant parameters that are used in the optimisation process. The number of units and their adjacency are not fixed constants as well, since the optimisation process of this model is distinguished by the very fact that the architect can adjust these parameters in order to better meet the objective. For example, several small units may be considered as a single larger unit if it is determined that their original size does not permit a feasible architectural solution.

Constraints
The model is based on adjacency relations between existent units and adjacency relations between the found groups. Adjacency is defined as a unit having at least one other unit as its neighbour. Furthermore, a feasible solution must meet the following auxiliary conditions:

- The number of units in a group matches the defined number of units for this group.
- The sum of the size of all units in a group (in square meters) is larger than or equal to the defined group size.
- The units in a group must be adjacent.

Upper and Lower Bounds
The model utilises upper and lower bounds when determining the deviation from the total group size required by the room program. These bounds are used to extend the potential solution space.

- *pso* - upper bound, i.e., percentage of required group size that must at most be met (usually ≥ 100%)
- *psu* - lower bound, i.e., percentage of required group size that must at least be met (usually ≤ 100%)

Objectives
In this model, target functions are in theory obsolete since the satisfaction of auxiliary conditions includes the satisfaction of the goal, i.e., finding the defined groups, group sizes, and numbers of units within groups (Constraint Satisfaction). It is, however, possible to define superior goals. Such superior goals weight the search for a solution. If, for example, all conditions except one are met within a floor plan (e.g., one group is five square meters too small or large), it would not make practical sense to not actually swap the units regardless. For this reason, such solutions that are made possible by upper and lower bounds are considered in the shape of weighted goals.

Search Strategies
The model does not utilise search strategies for the purpose of constraining the solution space of the optimisation problem. It merely uses a breadth-first search (Jungnickel, 2005) in order to ensure graph connectivity.

Sequence of Events
The Non-Destructive Model's sequence of events begins with the decomposition of the building's floor plan into units corresponding to the existing rooms. Given the satisfaction of the goal definition, this procedure permits reuse without structural modification. An important factor for successfully finding a solution is the definition of relationships between these units. The most significant relationship is adjacency, which ensures that units in a newly formed group are also adjacent. Since a complete integration of a room program into an existing building is rarely possible, the room program may optionally be decomposed into logical units. Equivalents of such units can be searched for within the floor plan. However, chances of finding a perfect fit to

the stock are still low for such decomposition. For this reason, matches between units can be defined as partial goals. Maximising a weighted sum of these partial goals can then be defined as a superior objective. Although this procedure has a negative effect on the quality of potential solutions, it does increase the size of the solution space. The solution can be further influenced by applying conditions to the units of the room program. A unit may be defined, for example, by individual unit size, the total size of units, and the number of units. By combining these definitions, the variance within a given floor plan constellation can be increased.

OPTIMISATION MODEL AND PROGRAMMING

A general optimisation model usually consists of a given number of variable and constant parameters, one or more objectives, as well as a fluctuating number of constraints. Each object that belongs to the model can be accessed and altered by the use of parameters. Within the model used in this research, a room exists as an object (node) with geometric parameters such as its surface area. Objects can also imply alphanumerical parameters such as their occupancy or neighbourhood. Parameters are defined in the form of variables (or constants), whereas variables can be used as inputs for the optimisation process. Responses result from the composition of other variables. If a variable is changed during the optimisation process, dependent variables will be changed as well. Inputs and Responses are named Optimisation Variables. These variables form the basis of constraints and objective functions. Both must be functions of one or more optimisation variables (Bhatti, 2000). Once the design problem is stated in form of design variables, constraints, and objectives, the parameters will be passed to the optimisation engine which tries to find a feasible solution to the problem. The optimisation model described above is a general model that can be applied to many problem domains beyond architecture. The relocation problem was set up by means of a programming paradigm that specifies a set of constraints and objectives that must be met without stating how to perform this task. A programming language that supports this paradigm and that was used in our research is OPL (Optimisation Programming Language) which was developed in 1995 (Van Hentenryck and Lustig, 1999). The optimisation model focuses on the geometric shapes of existing room limits or the orientation of system lines, such as those of the static system or the grid units of structural components. This abstraction of floor plans ensures non-invasive treatment of existing buildings.

Conventions:

$G = (V, E)$	graph with
V	set of nodes
$E \subseteq V \times V$	set of edges
$nbreOfNodes$	number of nodes $\left(= \lvert V \rvert\right)$
$Node_i$	node i of graph $\left(i \in \{1,...,\lvert V \rvert\}\right)$
$nbreOfGroups$	number of connected groups (sub graphs) to be found
$Group_j$	group j
$label_{ij}$	means that $label_i = j$

Variables:

$A = a_{ij}$	adjacency matrix of G with $a_{ij} = \begin{Bmatrix} 1, \text{falls } (i,j) \in E \\ 0, sonst \end{Bmatrix}$;
$sizeNodes_i$	size of node i in square meters
$groups_j$	number of required units in group j
$sizeGroups_j$	found total size of group j in square meters

$sizeRooms_{jk}$ size of room k in group j $\left(k \in \{1,...,groups_j\}\right)$

psu lower limit: percentage of required group size that must at least be met (usually $\leq$ 100%)

pso upper limit: percentage of required group size that must at most be met (usually $\geq$ 100%)

$label_i$ group to which node i belongs

Number:

$$\sum_{i=i}^{|V|} label_{ij} = groups_j \quad \forall j \in \{1,...,nbreOfGroups\}$$

Size:

$$\frac{pso}{100} * sizeGroups_j \geq \sum_{i=i}^{|V|} label_{ij} * sizeNodes_i \geq \frac{psu}{100} * sizeGroups_j$$

$$\forall j \in \{1,...,nbreOfGroups\}$$

EXAMPLES

The results of the model developed are encouraging. (Figure 3) shows 8 exemplary assemblies of 21 rooms. We were searching for 4 groups of rooms with 6, 6, 5, and 4 group members. All solutions fulfill the requirements pertaining to the size of the rooms, their adjacency, and the number of members in a unit (group).

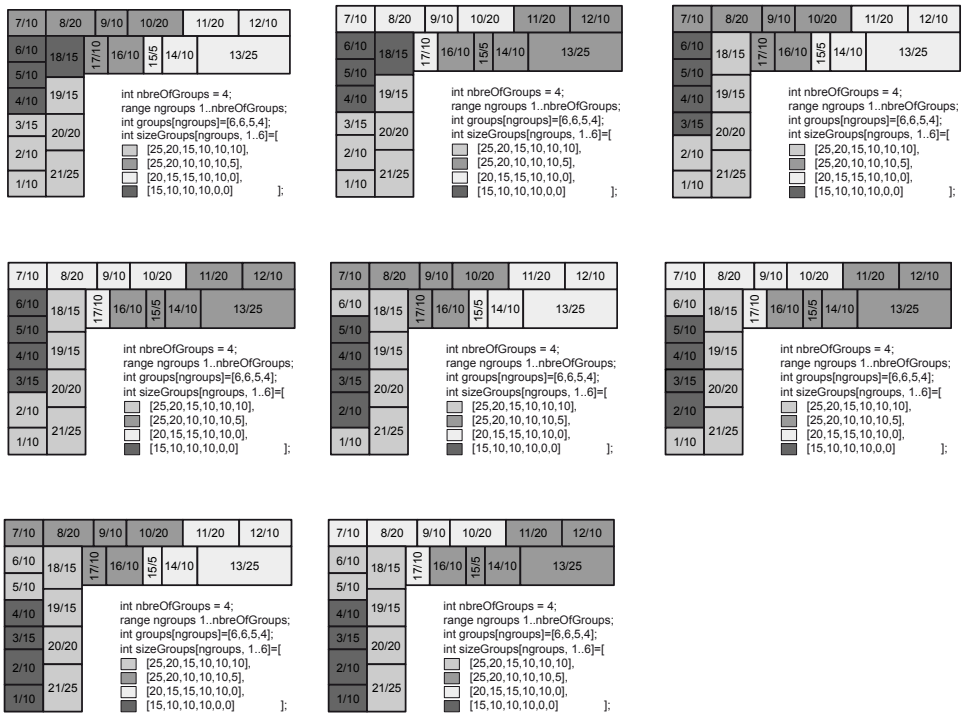

Figure 3. Non-destructive optimisation of existing floor plans

We began working with larger models consisting of 78 rooms, respectively 6,084 entries in the matrix (Figure 4). These optimisation runs can be solved within 2 hours time on ordinary machines with a reasonable amount of memory. Our latest attempts deal with more than 300

rooms (respectively 90.000 entries in the matrix) arranged on different stories. These tasks are difficult to solve from a computational point of view and might call for parallelisation of the programs developed. However, the matrix is capable of representing two- or three-dimensional arrangements (multiple stories) of rooms in equal measure. It can as well represent various buildings.

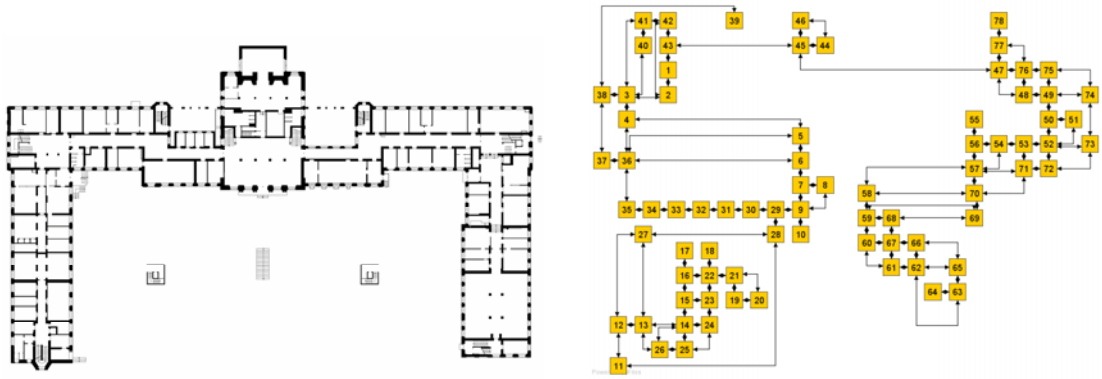

Figure 4: Plan and graph representation, TU Dresden, Fritz-Foerster Building

(Figure 5) demonstrates an exemplary calculation within which we were searching for 17 rooms. Each room had to fulfill specific requirements regarding its floor space and adjacency to other rooms. The figure represents 3 out of 4935 solutions found to the problem. All solutions range within a 3% tolerance of the specified objectives. To judge these solutions, we integrated quality ratings and performance measures as well as penalisations in the model. (Figure 6)

Figure 5. Three exemplary solutions

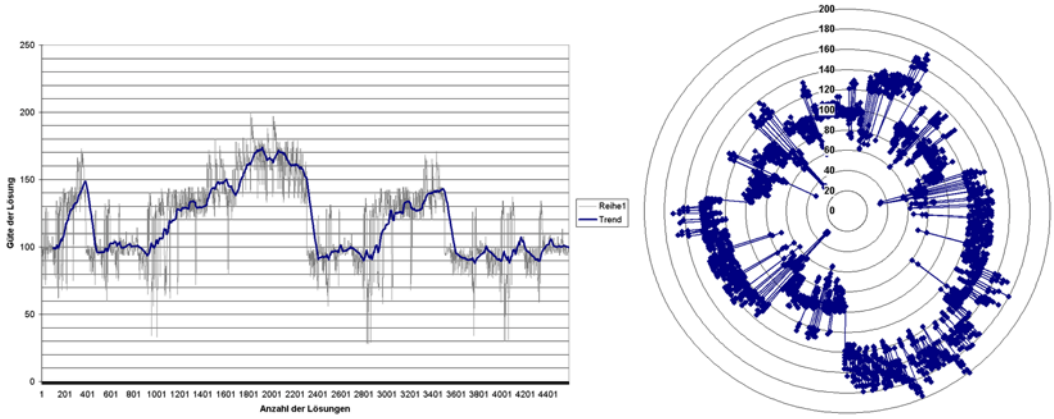

Figure 6. Calculation trend and performance measure

On the basis of these performance measures, a solution can be objectively evaluated. (Figure 7) shows the solution with the best performance. The numbers demonstrate a nominal / actual value comparison of the floor spaces we were searching for.

Figure 7. Best performance and nominal / actual value comparison

CONCLUSIONS

Non-destructive models can play an important role in the architectural design process. Our examinations demonstrate that these models, which are often NP-complete, dramatically benefit from search methods that the user is able to define in the programming language used (Van Hentenryck and Lustig, 1999). The examination of different models developed also demonstrates that many models could not be solved within an appropriate time range without the use of these search methods. However, the results of this model are promising not only from a computational point of view. Their greatest potential lies in the possible extension to multiple buildings. In the perpetual important domain of revitalisation, non-destructive optimisation models can aid the architect in finding quick answers to design problems.

REFERENCES

Bhatti, M. A. (2000) *Practical optimisation methods : with mathematica applications,* Springer, New York, NY.

Jungnickel, D. (2005) *Graphs, networks and algorithms,* Springer, Berlin.

Liggett, R. (2000) *Automated Facilities Layout: Past, Present and Future,* Automation in Construction 9, P. 197-215.

Van Hentenryck, P. and Lustig, I. (1999) *The OPL Optimisation Programming Language,* MIT Press, Cambridge, Mass.