

Adjacency in hospital planning

Wolfgang E. Lorenz*. Martin Bicher**
Gabriel X. Wurzer*

* *Digital Architecture and Planning, Institute of Architectural Sciences, Vienna University of Technology, Treitlstraße 3, 1040 Vienna, Austria (e-mail: <firstname.surname>@tuwien.ac.at).*

** *dwh Simulation Services GmbH, Neustiftgasse 57-59, 1070 Vienna, Austria (e-mail: martin.bicher@dwh.at), Institute of Analysis and Scientific Computing, Vienna University of Technology, Wiedner Hauptstraße 8-10, 1040 Vienna, Austria (e-mail: <firstname.surname>@tuwien.ac.at)*

Abstract: Adjacencies stand at the beginning of a multitude of planning tasks. Especially in hospital planning they are essential for describing relationships between different organizational units – e.g. ‘close’, ‘distant’ or ‘neutral’. Mathematically, these terms map to relative weights between each pair of units in the range $[-1, 1]$ which are put into a (symmetric) adjacency matrix. This matrix subsequently determines relative locations of individual spaces (preliminary space layout). The paper deals with the effective definition of this adjacency matrix in the context of early-stage architectural planning. In contrast to current planning practice, which looks at each adjacency relation in isolation, our approach uses a Newtonian gravitation model to propagate changes to a single relationship immediately to the whole space layout. As a result, we are able to supply architects with a design tool that accelerates the definition of adjacencies and lets them preview the preliminary space layout at the same time.

Keywords: Adjacencies, gravitational model, planning tool, hospital planning, space layout planning.

1. INTRODUCTION

In the preparatory phase of planning, the definition of relationships between organizational units plays a central role. Information for this process usually has to be provided by client, the staff and/or architectural planners, and is typically recorded in the form of a spreadsheet. The authors have often advocated for tools that use these relationships in the context of early-stage conception rather than for late-stage verification and optimization (Wurzer, Lorenz 2012). In line with that strategy, the proposed model is a representation of adjacencies in nodes (corresponding to functional units) and edges (their relations) which is also called *bubble diagram* in architectural terms:

- Each node is attributed with the minimal, average and/or maximum size of its corresponding functional unit (*space requirements*),
- an edge represents the weight between two nodes which is independent of size.

Many notions of relationships can be modeled in that fashion, e.g. collaboration between two units or spatial vicinity in the final layout. This paper points out the importance of first concentrating on this specific aspect during the design process (while other aspects are faded out) and presents a new contribution to visualize the relationships between entities. The goal is to get a preliminary space layout once the adjacency matrix is defined by the client or the planner (This can happen in form of a spreadsheet). To fulfil this requirement designers must be able to interact and influence

the model at any time. Outcome of the simulation is the computation of an arrangement of functional units (*schema*) which must not be confused with a floor plan: Basically spaces are laid out in a planar fashion. Subsequent planning might shift these spaces to different (three-dimensional) floors of a building while still keeping the adjacency relationship (e.g. by introduction of a connecting lift).

2. BACKGROUND AND RELATED WORK

Spatial relationships remain so far either purely abstract (adjacency matrices and preliminary sketches) with no computational method employed, see e.g. in Neufert (2000), or are considered as spatial constraint during automated floor-plan generation for the final design (e.g. Elezkurtaj 2002). Most of the work focuses on the latter category (Lobos and Donath 2010), also taking into account parameters other than spatial relationships – e.g. orientation in relation to the sun, views from site and to site, noise, size and proportion (compare White 2004). In effect, a multi-criteria optimization problem results with a large solution space which can be reduced to a certain extent by means of genetic algorithms (Dutta and Sarthak 2011). However, key point is that there is no “unique optimal solution for over-constrained problems” (Balachandran and Gero 1987). Thus the commonly used solution of this issue is to finally ask architects for their subjective opinion, since automated solution of spatial layout problems started in the 1960ies (compare limitations of commercial space allocation products in Liggett 2000). We argue that this is in fact a procrastination of the problem, where choosing a result now requires an experienced architect who has to consider all variables that have driven

the design generation, which is highly doubtful to say the least.

In our work, we are not seeking to tackle the aforementioned problems coming from the explosion of the solution space in late stages of design. We are rather going back to the early stages of design where the spatial adjacencies can be handled by themselves, coming up with a single solution that can drive the design in later stages.

As already described by Lohfert (1973), we seek to establish a planning process where each task is handled only once, in the correct phase of a project. We argue that the focus on the adjacency relationship between units is a pre-step for designing, situated in between the schematic phase and the final layout phase. It arranges entities spatially without defining form or dealing with physical sizes. Only in a further step physical quantities are taken into account, still not as definite space but always in a schematic representation (as bubbles of different size). Only two of 13 listed rational criteria for architectural floor-plan layout as described by Lobos (2010) are focused on in this paper: the fourth criteria “related functions” and the ninth criteria “geometric composition”, which poses for the overall container (overall shape defined by the planner).

Our approaches are inspired of the physically based modeling technique by Arvin and House (1999), in which spaces are modeled as point masses and relationships between them as spring exerts forces. Especially the advanced model that behaves according to Newton’s laws of motion, presented in the last chapters of this paper, applies the physics of motion to entities in a space plan. Similar to the solar system the location of entities or individual spaces is affected by forces between them, which lead to a stable phase (or if it does not, indicating that there is no stable phase at all). In order to be useful within the actual workflow, we allow architects to interact with the model, making it possible to shift spaces until the preset adjacencies are fulfilled.

3. FROM ADJACENCIES TO SPATIAL DISTRIBUTION

The present work is a complement to our research in the field of simulation in the early stage of planning (Wurzer, Lorenz 2013). Thus, simulation is based before the actual planning or in parallel. As consequence of this strategy, the requirements are incorporated directly into the design, which results in an immediate verification.

The model of nodes and edges is another contribution to early stage planning and combines adjacency requirement – a topological property – with spatial allocation without defining or even touching the final form. This is advantageous for the planner as the output represents a preliminary spatial concept without narrowing the form-finding – the initially presented model not necessarily finds a final distribution, but may permanently change because of irresolvable conflicts. Although the size and location of spaces are not yet fully formulated, both their mutual relationship and in a later version their basic extent are taken into account (Fig. 1).

One benefit of this model is that changes, such as additions of functions, relationship changes and/or size changes, remain transparent at all times, for the planner as well as the client.

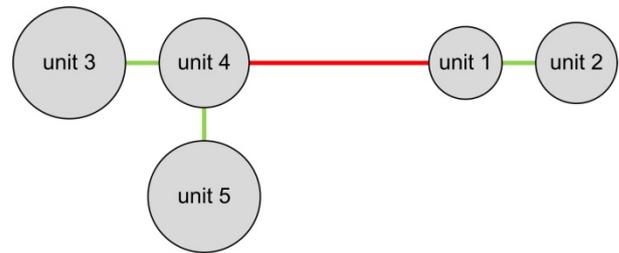


Fig. 1. Nodes (with size) and edges (green color: close relation; red color: distant relation)

3.1. Analyzing the final design

The same schema can be used for analyzing the final design (floor plan). At that time, both the size and the position of rooms – as containers of functions – are fixed. The evaluation is done in two ways:

- If the adjacency requirement (between functions) is not fulfilled, the edge between two nodes is coloured red (link between unit 1 and unit 4 in Fig. 2). In the other case, i.e. if the required relationship is fulfilled – the edge is colored green.
- If the specified size is not fulfilled, the node located in the middle of the room is coloured red (unit 1 in Fig. 2), otherwise it is coloured green.
- If the relationship between two units is defined as neutral, they are simply not connected by an edge (unit 5 in Fig. 5).

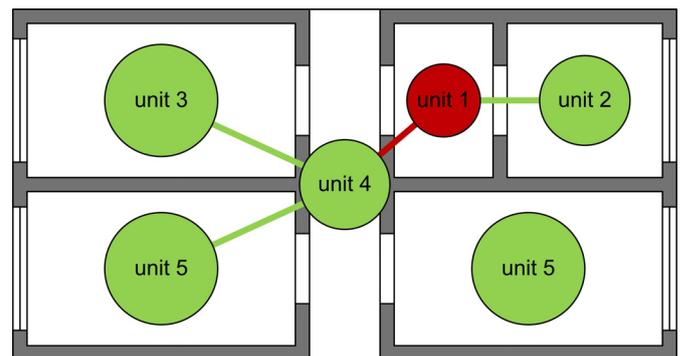


Fig. 2. Analyzing the final design (green color: fulfills the requirements; red color: does not fulfill the requirements)

3.2 A model for teaching

As the approach is very intuitive and the simulation is very user-friendly the model can also be used for teaching. For example, the sensitivity for relations between individual departments of a hospital can be trained to the staff. Working

with our model the user immediately receives feedback on the edges (links between nodes) when changing the settings. This way the following questions are answered visually: What happens to the whole system if...

- ... information about the adjacency of a single node is changed?
- ... the distance between two departments is changed by moving nodes?

Processes and relationships in a hospital are more comprehensible when entities are visualized as interacting nodes. Furthermore, a comparison of the adjacency from the patient's point of view and of the staff's point of view visualizes different sides of a coin.

4. THE INITIAL MODEL

In an adjacency matrix, each (functional) unit is connected to each other. A close relationship is identified by a green colour, a distant one by a red colour (Fig. 3). In a small graph, these relationships are relatively easy to handle. But, when dealing with adjacencies, complexity increases with the number of units and their relations. If we persist in the simple example of Figure 3 (adjacency matrix), then two different groups are obvious. The first group includes those units which attract each other – these are the pairs 3-4, 3-5 and 5-6. The other group repels each other and includes the pairs 1-2, 1-4 and 2-6.

What is a possible spatial solution of this adjacency matrix? First, adjacencies are translated into forces that act on nodes. This enables to visualize different degrees of interrelations by different thickness of the connecting lines (which is not implemented in the first step). A possible solution of the initial adjacency matrix is given in Figure 3 (right), in which the group of attracting units emerges significantly at the bottom. This simple model does not contain a value for the degree of relationship, which is the degree of closeness or distance, nor have the units a certain size. Nevertheless, it visualizes one possible spatial distribution. From this first observation, the main points for the development of a computer-program can be derived:

- restricted area should be given by the outline of the building,
- the initial distribution may not influence the final result (a random distribution is preferably),
- the relationship between nodes should be specified in degrees (0 to 100%) or between [0,1], and
- the size of the units should be given by diameters (circles may overlap each other to a certain extend).

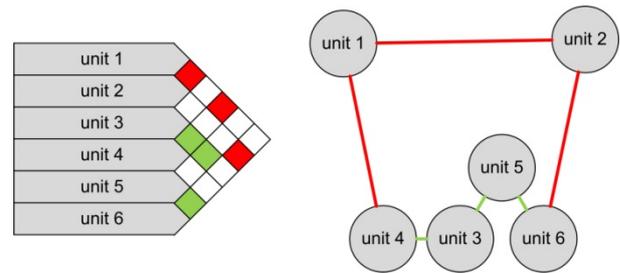


Fig. 3. Simple adjacency matrix (left); Possible spatial solution (right)

4.1 The implementation in NetLogo

Our starting model is implemented in NetLogo (a simulator basically designed to perform agent-based simulations. In order to achieve a spatial relationship the position of every single node is changed: Every single node is analyzed in relation to every other node, which includes three possible states: move towards, move away to a certain extent and stay. The latter is applied if the position between two nodes is neutral.

Here we consider a way to convert adjacency matrices into special relations by a model of nodes and edges. The data, coming from diagrams as commonly used in design work (White 1986), are first translated in a text file. This text file contains all relationships with the respective name of the unit followed by one list enclosed in square brackets and one enclosed in curly braces (Fig. 4). The first list contains close relationships and the latter distant ones, which allows considering one-sided relationships as well. This information is then stored in each node, again as two lists of names:

- firstly with names that have a close connection, and
- secondly with names that have a distant connection.

```
unit00 [unit04 unit01]{unit05}
unit01 [unit04 unit02]{unit05}
unit02 []{}
unit03 [unit05]{unit00}
unit04 [unit01]{unit00}
unit05 [unit03]{}
unit06 [unit03]{unit01}
unit07 [unit03]{unit00}
```

Fig. 4. Text file containing all relationships – attracting units in square brackets, rejecting units in curly brackets

Our first model starts with a random distribution of the nodes within a limited given space – e.g. the outline of the building (yellow space in Fig. 5). Initially, all nodes that have either close or distant relation are connected by an edge. In the next step, each node and its relation to all other nodes is analyzed step by step. That means the examined node moves one unit step towards a node which should be close. Similar it moves one step away from those who should be distant. This is done step by step. One result is presented in Figure 5. Here, two groups emerge with the following configuration:

- the repulsion of node number 1 of the right upper group with node number 0 (visualized by a red dot at unit 1 and a red line to unit 0), and
- the attraction from node number 1 to node number 2 and 3 (visualized by green lines with green arrows starting from unit 1).

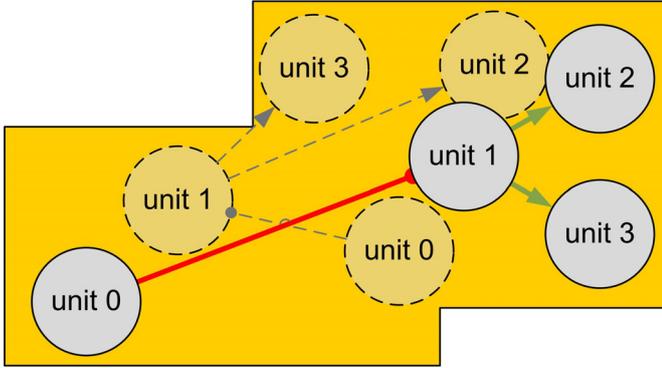


Fig. 5. Initial configuration of the experimental arrangement (dotted circles) and final contribution (continuous lines)

In an advanced version, the size of the spaces is taken into account by the radius of the units (Fig. 6). Even this simple model gives an impression of how adjacencies can be translated into a spatial distribution. With just a few nodes the system comes to a state where it slows down or even cannot be solved. These problems can occur as nodes may be captured so that they cannot pass each other. The user itself can solve these issues by interacting with the model manually moving some nodes.

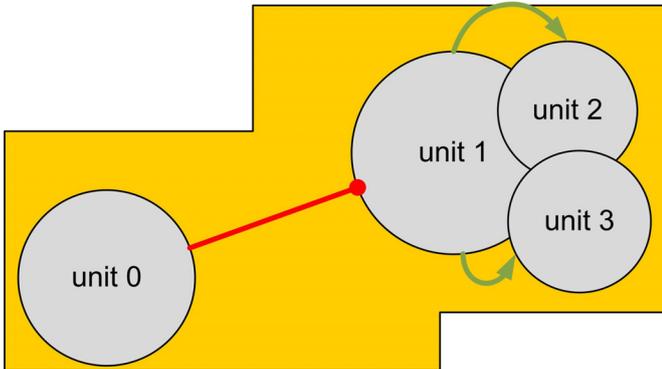


Fig. 6. Final contribution including size

5. NEWTON'S DIFFERENTIAL EQUATION

5.1 Introduction

In contrast to the previous described more simple model, which examines one node after the other the expanded model examines the behaviour of all nodes at the same time. This permits simultaneous viewing. Again it is possible that the model stops before a valid result is reached. The system comes to a standstill as soon as other nodes are in the way,

which means that the constellation prevents further movement. Therefore, a Monte Carlo simulation (rerunning) is proposed, which leads to several results. Finally, the verification of all results is performed by other (design) criteria.

5.2 The expanded model

Given the adjacency matrix $A \in \{-1,0,1\}^{n \times n}$ of $n \in N$ different units an optimal configuration of them within a certain space (e.g. the outline of the building) is needed to be found. Hereby the definition of an optimal configuration is a rather intuitive one as it only includes the following two aspects:

- If $A_{i,j} = -1$ unit j tends to be as distant as possible from unit i .
- If $A_{i,j} = 1$ unit j tends to be as close as possible to unit i .

However these two guidelines are not a strict definition and leave space for different solutions.

One possible interpretation is inspired by the gravitational attraction of two bodies by their mass. Referring to Newton's laws of motion (Newton 1726) movement of a body located at point x in space is defined by the sum of acting forces F_i and its own mass m via its acceleration:

$$\ddot{\vec{x}} = \vec{a} = \frac{\sum_i \vec{F}_i}{m}. \quad (1)$$

Additionally, referring to Newton's law of gravitation (Newton 1726) a force \vec{F} called gravitation is always acting on a body evoked by the presence of a second body with mass m' and location \vec{x}' :

$$\vec{F} = Gm'm \frac{\vec{x}' - \vec{x}}{|\vec{x}' - \vec{x}|^3}. \quad (2)$$

Hereby G denotes the so called gravitational constant. Connecting these two equations for n bodies located at a two dimensional surface a highly nonlinear system of $4n$ differential equations emerges simultaneously describing the motion of all bodies evoked by their gravitational attraction. Each body hereby fulfils the principle of stationary action (Hamilton's principle see Hamilton 1834) tending to reduce the distance to all other bodies.

A trivial observation shows that the attraction of each pair of bodies is defined by the direction of the force \vec{F} . Changing the sign of the term $Gm'm$ – which is (of course) physically nonsense – suddenly leads to detraction of the two bodies. Intuitively setting $Gm'm := A_{i,j}$ in formula (2) and $m=1$ in formula (1) a dynamic model fulfilling the two aforementioned aspects is achieved. To finally keep the bodies in the predefined domain (which is the outline of the building) – denoted by $\Omega \subseteq R^2$ and bounded by a (at least partially) smooth curve $\partial\Omega$ – the acting forces need to be

projected on the direction vector $\vec{d}(\bar{x})$ of the margin, if a body tempts to leave the area. Summarising, these ideas lead to the following system of differential equations:

$$\ddot{\bar{x}}_i = \begin{cases} \sum_{j=1}^n A_{i,j} \frac{\bar{x}' - \bar{x}}{|\bar{x}' - \bar{x}|^3}, \bar{x} \in \Omega^\circ \\ \left(\sum_{j=1}^n A_{i,j} \frac{\bar{x}' - \bar{x}}{|\bar{x}' - \bar{x}|^3} \cdot \vec{d}(\bar{x}) \right) \vec{d}(\bar{x}), \bar{x} \in \partial\Omega \end{cases} \quad (3)$$

Finally, possible steady states of these equations lead to the desired optimal configuration. As Ω may be a very complex shape and the existence of steady states cannot be guaranteed it is hardly possible to find them analytically. Numerical methods can be used to either detect possible steady states via zero-point algorithms or to iteratively solve the dynamic differential equation system waiting for it to converge towards its steady state. Hereby Monte Carlo analysis is applied, which varies the initial positions. So far the implemented system is only suitable for point-shaped bodies (units treated as nodes). In order to be suitable for rooms with a specified geometry, some additional boundary conditions have to be added.

A big benefit of the presented method is that it is not limited to adjacency matrices $A \in \{-1,0,1\}^{n \times n}$ and can be extended to $A \in R^{n \times n}$ in order to model strong and weak attractions.

6. IMPLEMENTATION

In order to receive an executable program the open-source physics engine Bullet-Physics (Boeing, Bräunl 2007) was chosen. The engine is mainly known for precise and fast collision detection in multi-body systems. As it is integrated in the open-source 3D visualization software Panda3D (Lang 2011), usually used by game designers for the development of PC-games, we decided to use the framework for the representation of simulation results. Therefore extensions of the principle regarding a 3-dimensional version of the planning tool are possible and can be a target for future work. The underlying multifaceted programming language Python (Sanner 1999) provides enough flexibility to guarantee that simulation results can be reproduced, documented and reworked with integrated statistical post-processing routines.

Figure 7 shows a simulation snapshot for the adjacency matrix defined in Figure 3. Initially all nodes were placed randomly and after dynamic physical simulation the shown steady state was achieved. As the bounding area is not convex and has lot of edges the imaged configuration might not be the only local energy-minimum. Therefore we are performing several simulation runs varying the initial positions of the units in combination with a weighting-function rating the steady-state configurations. After all finally an optimal solution can be obtained (optimal with respect to the weighting function).

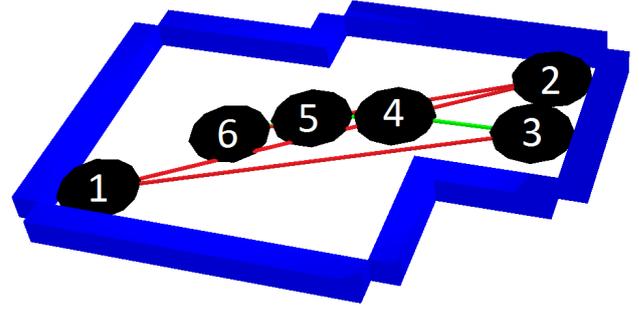


Fig. 7. Snapshot of an achieved steady-state for the adjacency matrix described in Figure 3.

Experiments showed that the forces defined in (2) on the one hand decrease too fast when units move apart and on the other hand get too strong, when units are close together. Therefore the formula was slightly adjusted leading to equation (4) using parameter $\alpha = 2$ instead of 3.

$$\vec{F} = Gm'm \frac{\bar{x}' - \bar{x}}{|\bar{x}' - \bar{x}|^\alpha} \quad (4)$$

So, instead of a quadratic decrease, forces only drop linearly when bodies move apart.

6.1 Performance

As most so called many-body simulations, the computation time and efforts of the presented algorithm increases squarely with the number of objects respectively (in our case functional units). Hereby the main efforts result as a sum of two different sources:

- Treatment of forces between entities
- Detection and treatment of collisions between entities and borders

Both lead to quadratic computational efforts. Solely for very sparse adjacency matrices and very efficient collision-detection algorithms, we might receive lower orders. It is furthermore very important to mention that a very complex geometry for the floor plan also leads to long computation times and high efforts.

7. CASE STUDY

As case study serves the example of an architect's office described in White (1986). The matrix consists of 20 different functional units, some of which involve multiple entities, e.g. 4 principals' offices or 2 staff conference rooms. The conventional way first analyses the matrix for potential design implications. Then a rough bubble diagram is drawn, cleaned up, refined and again read for potential design implications. Finally, building organization concepts are developed out of zoning diagrams which includes sorting qualities (e.g. operational groupings, open-closed spaces, or public versus private spaces).

The model of Newton's law of motion automatically detects and immediately visualizes functional clusters and relations which make the process of redrawing obsolete (Fig. 8). The detection of the outline may be included into this process by starting with a rectangular space and then optimize the plan step by step based on each result.

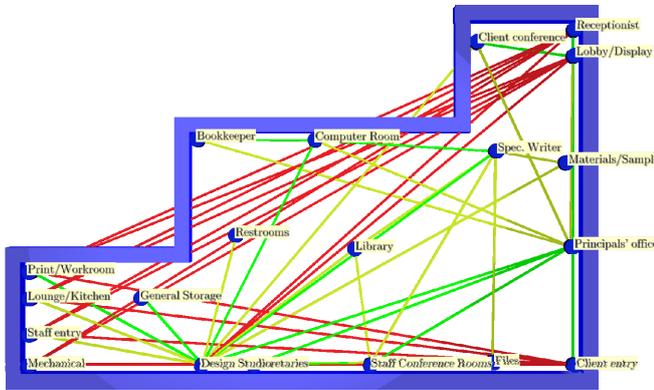


Fig. 8. One result of the case study.

8. FUTURE WORK

The here presented expanded model shows promising results. It indicates for example a stable adjacencies system if, and only if, all forces are balanced. This is visualized by a structure, where the whole may move, but inner parts do not change significantly. In turn, the system gives visual feedback for mismatching adjacencies in form of e.g. rapid changes. The next step of implementation concerns orientation which is also a topological property. This will lead to an even more meaningful structure (if the data allows a stable phase at all) on which the planner can work on.

9. CONCLUSIONS

We have presented a novel distribution to get a spatial relation out of a given list with adjacencies, targeted at the pre-tender phases e.g. of a hospital project. The authors have extended a simple model, based on a kind of rubber bands, by applying Newton's differential equation system. The output of this new system is suitable as input for the final design, since it gives the relation as positions and in the extended model their size, without defining form.

ACKNOWLEDGEMENTS

This paper was financed by the ZIT MODYPLAN project, under the call Smart Vienna 2012.

REFERENCES

Arvin, S., House, D. (1999). Making Designs Come Alive: Using Physically Based Modeling Technique in Space Layout Planning. In *Proceedings of the International*

conference on computer aided architectural design futures, 8, Atlanta, pp. 245-262.

Balachandran, M. and Gero, J.S. (1987). Dimensioning of architectural floor plans under conflicting objectives. In *Environment and Planning B* (14), pp. 29-37.

Boeing, A., Bräunl T. (2007). Evaluation of real-time physics simulation systems. In *Proceedings of the 5th international conference on Computer graphics and interactive techniques in Australia and Southeast Asia ACM*, pp. 281-288.

Dutta, K. and Sarthak, S. (2011). Architectural space planning using evolutionary computing approaches: a review. In *Artificial Intelligence Review* 36, pp. 311-321.

Elezkurtaj, T, Franck, G. (2002). Algorithmic Support of Creative Architectural Design. In *Umbau*, 19, Vienna, pp. 1-16.

Hamilton W.R. (1834). On a General Method in Dynamics. In *Philosophical Transaction of the Royal Society Part II*, pp. 247-308.

Lang, C. (2011). *Panda3D 1.7 Game Developer's Cookbook*, Packt Publishing Ltd.

Liggett, R.S. (2000). Automated facilities layout: past, present and future. In *Automation in Construction* 9, pp. 197-215.

Lobos, D, Donath, D. (2010). The problem of space layout in architecture: A survey and reflections. In *arquitekturarevista*, Vol. 6, n° 2, pp. 136-161.

Lohfert, P. (1973). *Zur Methodik der Krankenhausplanung*, Werner Verlag, Düsseldorf.

Neufert, P. (2000). *Bauentwurfslehre*, 36th edition, Friedr. Vieweg & Sohn, Braunschweig/Wiesbaden.

Newton, I. (1726). *Philosophiae naturalis principia mathematica. Bd. 1: Tomus Primus*, London.

Sanner, M. F. (1999). Python: a programming language for software integration and development. In *J Mol Graph Model* 17.1, pp. 57-61.

White, E.T. (1986). *Space adjacency analysis: Diagramming information for Architectural Design*, Architectural Media Tallahassee, USA.

White, E.T. (2004). *Site Analysis: Diagramming information for Architectural Design*, Architectural Media Tallahassee, USA.

Wilensky, U. (1999). NetLogo. Software. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Available from: <http://ccl.northwestern.edu/netlogo> [accessed 7 March 2013].

Wurzer, G., Lorenz, W. E. (2012). Pre-Tender Hospital Simulation Using Naive Diagrams As Models. In *Proceedings of the International Workshop on Innovative Simulation for Health Care 2012, Dime Università Di Genova*, pp. 157-162.

Wurzer, G., Lorenz, W. E. (2013). From Quantities to Qualities in Early-Stage Hospital Simulation. In *Proceedings of the 2nd International Workshop on Innovative Simulation for Health Care 2013, Athens*, pp. 22-27.