The generation of possible space layouts

Jesper Thøger Christensen
Research Assistant, Aalborg University, Denmark
jtc@civil.aau.dk

The general scope of this paper is to present the development of an approach for the generation of possible space layouts in the early stages of design. The approach sets out to bridge the gap between the free form brainstorming of the sketching designer and the deterministic parametric model of the computational designer. A new responsive design tool is developed; applying physical based modelling techniques to a computational environment. Springs and particles automates the initial placement and sizing of the spaces, and allows continuous user and software interaction. Keeping the design process alive, through springs that connects everything to everything and allows topology to change, opposed to the strict hierarchy and constraints of traditional parametric design. Thus instead of the topology being determined by initial constraints, the layout will be generated by probabilities of spaces to connect. Letting the storyline of potential users give answers to possible space layouts.

Keywords: Space layout, Digital aids to design creativity, Design tool development, Spatial quality, Storyline

MOTIVATION - SPACE AS A GRADIENT

The goal is not to generate an optimized geometric layout, but a range of (m)any possible topological layouts - a narrative landscape - to challenge the architect's imagination and inspire to investigate the possibilities further.

Compared to other methods of spatial layout, the spaces represented in this method will not be limited to represent functional requirements (kitchen, living room, bathroom, bedroom etc.) but also spatial qualities in general (light, dark, wide, deep, long, lifted, heavy, open, closed, private, public etc.)

This rejection of a pure functional division of space originates from my idea of space as a gradient. Here functional requirements are not the only driver of space, but also spatial qualities and the movement through space.

The field of view of the individual and the history of his movement will determine his next movement. This continuous revelation of space, Gordon Cullen named serial vision, and splits the optical viewpoint into two elements: the existing view and the emerging view. In separating the existing from the emerging view, Cullen captures the feelings related with the experience of the current and the anticipation of the emerging (Cullen, 1961). Recognizing the role of serial vision one can use it to build anticipation, create drama, and invoke feeling. Therefore, space is a discontinuous landscape without clear boundaries, where the storylines of the individuals acting in the space, is just as important as the actual space itself.

For simplicity, this initial paper will only gener-
The question at hand in this initial paper is topology being more important than the geometry. Thus, generating solutions for spatial layouts, which can inspire the topology of how spaces connect, without necessarily giving final answers to their geometry. (figure 1)

**THE TOOL**

The digital tool is not supposed to replace pen and paper in the early design process, but give the designer another resistance, than the unconscious mind acting through the hand and pen onto a piece of paper. The resistance being the definition of a set of parameters, as input to the generation of a storyline, and the tool’s subsequent interpretation of this.

The intention of the tool is to render the spatial layout as a storyline of spaces, where only the probability of two spaces to follow each other are predefined. Thus before using the tool, one do not need a conception of the resulting organization, but only the elements and their probability of connecting. Resulting in a spatial layout that allows perception, interpretation and behavior. Not following, but allowing it to happen.

For these spatial relations to have an effect on the layout, the tool is set up as a physical system of particles connected to every other particle with various types of spring forces. The particles represents spaces and the spring forces various spatial relations, one of them being the storyline, another the possibility of similar spaces to merge and create loops in the storyline.

**How the tool works**

The tool is written in Java in and organized in an object oriented way.

It extends the open source programming environment of Processing, to make use of its rendering mechanism and basic classes and methods. ControlP5, provides a GUI (Graphical User Interface), used to generate the user interface of the tool. NooLab, is the library that provides access to generation of the...

---

**Table: Qualities and Additional properties for a narrow quality**

<table>
<thead>
<tr>
<th>Qualities</th>
<th>Additional properties for a narrow quality</th>
<th>Transition matrix</th>
</tr>
</thead>
<tbody>
<tr>
<td>narrow - A</td>
<td>width 1 -- 2 [m]</td>
<td>A</td>
</tr>
<tr>
<td>dark - B</td>
<td>openings 0 -- 1 [#]</td>
<td>B</td>
</tr>
<tr>
<td>low - C</td>
<td>height 3 -- 4 [m]</td>
<td>C</td>
</tr>
<tr>
<td>high - D</td>
<td>length 4 -- 7 [m]</td>
<td>D</td>
</tr>
</tbody>
</table>
storyline and assigning additional geometric properties to each state of the storyline.

**Input for the tool**
The input for the generation of a storyline (figure 2):

- A set of named spaces. Ex. narrow, dark etc.

- A range of return values that specify additional properties of that space. Ex. a range for the width of a narrow space.

- A transition matrix of probabilities for the various spaces to be proceeded by any of the other spaces in the creation of a storyline. Ex. from a narrow space (A) there is 10% possibility to proceed into a low space (C), but from a low space there is 0% possibility to proceed into a narrow space.

In terms of programming the storyline is conceived as a Markov Chain. A Markov Chain is a mathematical system that undergoes transitions from one state to another, between a number of possible states. It is a memoryless probabilistic process, where the next state depends only on the current states possibility to connect to another state, and not the sequence of events that preceded it [1]. These probabilities are set by the designer and defined in the transition matrix.

**The tool generates a storyline**
The tool will start out with one storyline, with the possibility to add an infinite number of storylines.

A random identifying letter defines the first space of each storyline (figure 3). The transition matrix of the Markov chain generates every proceeding space. However, before generating the next space in the storyline, the current space translates to a particle with a certain size and some additional properties. Further, it connects to all other particles in the system through various springs.

With a sufficient long storyline, the output will represent a string of some of the various configurations that can emerge from the predefined matrix. The spaces at the two ends of the storyline will only have one connecting room, thus acting like a dead end, with only one possible option to proceed movement. In the other spaces with two connections, movement is predictable, and to proceed one can only go forward or backward.

**The influence of the physical spring system**
For a space not to become static and movement predictable, some of the spaces needs at least three connections to other spaces. One being the previously occupied space and the other two, the ones to choose between for the next step. Doing so, by allowing similar qualities within the same storyline or across storylines to merge (figure 4).

Separating spaces that are conflicting and con-
Figure 4
Not just one sequence of spaces, but several sequences overlapping. The letters represent various spaces.

Figure 5
Springs used to connect particles.

Figure 6
Springs used to align particles.
necting spaces that are compatible and reinforcing, ensures that spaces within and across storylines can interact and cross each other. Practically this is possible through a set of different springs between the spaces.

**Springs used to connect particles.** MarkovSprings are used to construct a spring between to particles (space) being created one after the other to make up the storyline. The length of the springs is set for the two particles to exactly touch each other (figure 5).

MinDistanceSprings are used between spaces with different qualities, which are not directly connected with a MarkovSpring, and therefore should be kept separate with a spring of a minimum distance (figure 5).

WithinDistanceSprings are created between spaces that have the same quality, but are not directly connected with a MarkovSpring. This allows the particles to merge into one, when they are within a certain threshold (figure 5).

**Springs used to align particles.** ChildParticles around the border of the parent Particle is controlled by the SoftBody class, which adds a polygon of ChildParticles to each Particle. The number of polygon vertices and ChildParticles per side is dependent on the additional geometric properties of the quality (figure 6).

CollisionSprings are used to align the SoftBody polygon of ChildParticles to neighbouring SoftBody polygons (figure 6).

**User interface**

The user can interact with the physical system by dragging spaces around or selecting spaces to prune. Further the user can turn the display of various springs and particles on or off.

At any point in the process, the markov chain can be paused, allowing the user to capture an image of a promising layout, prune parts of the layout or interact with it, without new spaces popping up.

**INVESTIGATING THE OUTPUT OF THE TOOL**

Discussing the effect on spatial layout by changing the transition matrix.

**A deterministic model**

One can completely control the output from the tool, when setting up the transition matrix so that each space can only connect to the next space in the list (figure 7).

This transition matrix will always produce the same spatial sequence and movement will quickly become predictable. Not making use of the tools

![Transition matrix](image)

**Phenotyping - for quality A**

<table>
<thead>
<tr>
<th>size</th>
<th>25</th>
<th>35 [m]</th>
</tr>
</thead>
<tbody>
<tr>
<td>sides</td>
<td>10</td>
<td>10 [#]</td>
</tr>
</tbody>
</table>
computational power. Furthermore, circular rings appear when similar spaces starts to merge.

**A corridor emerges**

As the transition table is configured and the resulting layout shows, quality A is dominant, as it links to every other quality (figure 8). Further, as the sequence A-A has a probability of 0.7, A will be followed by A quite often. This could lead to the perception that A symbolises a corridor space, whereas B, C and D could be considered as adjoining rooms.

Again the setup of the transition matrix generates a quite familiar result, which one could have predicted beforehand. Thus not taking full advantage of the tools opportunities to create differentiated answers to the same problem.

However if playing more with the input values and letting the program run for a longer time, one would be able to investigate the possibilities of this setup even further and possibly arrive at new spatial configurations for such a topology.

**A probabilistic model**

Spaces with more than one space to connect to, create various arrangements each time the program is run (figure 9).

Thus, the resulting configuration is changing from time to time, but one should be able to find identical sequences from model to model.

**Two storylines**

Two storylines, which each have a unique transition matrix but some similar qualities can merge to form more complex storylines (figure 10). This crossover creates completely new sequences that are different from what initially predicted from the two separate transition matrices. Thus with an infinite number of stories and qualities the result will be an infinite narrative landscape of spatial qualities (a qualitative landscape) where you, by dissecting the landscape into smaller parts, can find any organization/building - built and unbuilt - within the landscape.

This allows one to take full advantage of the tools opportunities to create differentiated answers to the same problem. Answers that one could not have thought of beforehand.

This in turn, also demands that the user of the tool takes time to interpret the findings, and to create their own story from the output presented to them. Just as intended for the tool.
Figure 9
A probabilistic model

Transition matrix

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>E</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>0.2</td>
<td>0.5</td>
<td>0.2</td>
<td>0.1</td>
<td>-</td>
</tr>
<tr>
<td>B</td>
<td>0.4</td>
<td>0.2</td>
<td>-</td>
<td>0.3</td>
<td>0.1</td>
</tr>
<tr>
<td>C</td>
<td>0.1</td>
<td>0.1</td>
<td>0.4</td>
<td>-</td>
<td>0.4</td>
</tr>
<tr>
<td>D</td>
<td>0.2</td>
<td>0.1</td>
<td>0.2</td>
<td>-</td>
<td>0.5</td>
</tr>
<tr>
<td>E</td>
<td>0.3</td>
<td>0.3</td>
<td>0.3</td>
<td>0.1</td>
<td>-</td>
</tr>
</tbody>
</table>

Phenotyping - same for all qualities

<table>
<thead>
<tr>
<th></th>
<th>size</th>
<th>50</th>
<th>50 [m]</th>
<th>sides</th>
<th>4</th>
<th>4 [#]</th>
</tr>
</thead>
</table>

Figure 10
Two storylines starts to merge

Figure 11
Possible layouts generated from the tool
DISCUSSION
Only setting up relations between functional requirements for a single storyline will generate a traditional spatial layout or bubble diagram (White, 1986), without being influenced by spatial qualities and the storyline of other individuals. If one on the other hand would only set up relations between spatial qualities, it becomes a more abstract layout of spatial qualities without any functional relation. In both cases limiting the space of possible solutions.

However, combining the functional requirements and the spatial qualities, both will inform each space of the spatial layout, and result in a true combination. This will be important for the future work, make the method more useful, and further distinguish and improve it compared to other methods of space layout in architecture.

CONCLUSION
The use of several spring systems, for various types of relationships between the spaces, allows the sketch to develop beyond the relations of each storyline while maintaining a balanced composition.

As it is a probabilistic process, the output will never be the same. Thus, it is neither a random nor a deterministic process, neither unpredictable nor predictable, but somewhere in between. Not optimizing, but using the power of computation to give differentiated answers to the same problem. Answers that one could not have thought of beforehand. This in turn, also demands that the user of the tool takes time to interpret the findings, and to create their own story from the output presented to them. Ultimately using the tool as a true partner in design.

Developed further this method of generating differentiated answers can implement other design intentions, such as daylight, view etc. Especially in the phase of determining an initial layout for further exploration in conventional parametric software.

REFERENCES
Lobos, D and Donath, D 2010, 'The problem of space layout in architecture: A survey and reflections', Arquitetura Revista, 6(2), pp. 136-161
White, E 1986, Space Adjacency Analysis, Architectural Media LTD, Tucson