Communication between humans can be split into two types, formal communication and informal communication. Formal communication happens when people meet at a specific place and time, with the topic and the intentions mostly known to all participating. Informal communication on the other side is spontaneous. People, known to each other or not, accidentally meet and start talking to each other. Since the conversation wasn't planned upfront, the information gathered through this conversation is most likely surprising and therefore its information content is larger. Especially in information driven working environments, like universities, this form of communication is important.

Informal communication as a function cannot be planned. Instead the process of meeting has to be planned. To enable this, two criteria are crucial: movement through a space and the visibility of that space. These two criteria were the driving force behind the design. In order to make those criteria generative, not only analytical, some of the measurements proposed by “Space Syntax” were implemented into Grasshopper with the “SpiderWeb” plugin (GlahnNetz; Richard Schaffranek; http://www.gbl.tuwien.ac.at/archiv/digital.html [visited 08.10.2011]).

This design proposal is an investigation into the possibilities such analytical tools can have in combination with a generative environment, to help organise a building that would encourage especially informal communication. Therefore another form of building organisation, beyond corridors, open-plan,…. had to be found.
MATRJOSCHKA

set building footprint and entrance
define roomheight and maximum number of floors
place vertical circulation and atria
close to entrance fare from entrance
compute average distance to entrance → distance map
interpret generate geometries by hand
1st autogenerative step:
order function descending according to their "choiceBetweenPoints" value;
compare this list with the imported list;
the fitness of the solution is the difference between those lists (less difference = better fitness);
reposition variables and WMA;
→ minimal circulation
randomly add different circulation to the minimal circulation
to ensure a working circulation, compute the minimal and maximal spanning tree
→ minimal circulation

allocation heuristic:
allocate functions based on the values of "choiceBetweenPoints":
first function → smallest value; erase all points occupied by the first function; if a room is larger than i.e. 100 m², it occupies space on two floors;
second function → smallest remaining value; erase all points occupied by the second function;
... repeat until all functions are allocated;

2nd autogenerative step:
order function descending according to their "choiceBetweenPoints" value; compare this list with the imported list;
the fitness of the solution is the difference between those lists (less difference = better fitness);
reposition variables and WMA;
generate circulation and room-geometry;
the amount of solid / transparent wall is influenced by the value of "choice between points";

all 102 family suites have a very low "desired choice" value; a room with a high "desired choice" value should (i.e. faculty club) be part of the shortest path between all functions within the building more often than a room with a low value (i.e. family suite);
"desired choice" is a relative value;

allocate functions based on the values of "choiceBetweenPoints":
first function → smallest value; erase all points occupied by the first function; if a room is larger than i.e. 100 m², it occupies space on two floors;
second function → smallest remaining value; erase all points occupied by the second function;
... repeat until all functions are allocated;

import list and sort it descending according to 
"choiceBetweenPoints" value.
shed (the function 'faculty club') be part of the shortest path between two points? early often
compute all possible circulation between the placed functions → circulation graph

draw 1:200

richard schaffranek - designing form the inside out 1144