POSITION PAPER FOR THE FOCUS GROUP "WHAT MAKES PATTERN LANGUAGES WORK WELL" EUROPLOP 2002

KRISTIAN ELOF SØRENSEN <ELOF@ELOF.DK> FINANSSYSTEM A/S
HTTP://WWW.FINANSSYSTEM.DK

From reading "A Pattern Language" and "Nature of Order" by Alexander, I got the idea that a software construction that is perceived as good, must have a pattern language as it's foundation. This can be with or more likely without the software manufactors knowledge.

For an existing good piece of software, an underlying pattern language must exist, that matches all the important design decisions of the piece of software. Areas of bad design will correspond to loose ends in the pattern language, such as unresolved forces and consequences that are not being dealt with by other patterns implemented in the software.

To test this theory, I started studying J2EE application servers and comparing them to earlier script based approaches to solving the problem of how best to create applications for the web. An early result of this work can be seen in the pattern language "Session Patterns" in the proceedings for EuroPLoP 2002.

So far I have reached the interim conclusion, that there do exist a pattern language for solving this problem, and that it manifests itself in the parts of the various software products that works the best in practice.

I feel, that for a certain problem, such as how to make applications for the web, progress is being made by the software manufactors up until the point where they have discovered the pattern language that lies underneath the problem they are trying to solve. After that point has been reached, the software can mature by means of bug fixing, resolving the remaining minor forces still left unresolved, code clean up etc. From this point the software can only evolve further, if it's creators attempts to solve new problems.

So at the point of software maturity, the complete pattern language beneath it, can be mined and written down.

It is possible to mine patterns earlier in the software life-cycle, than when maturity has been reached. The areas of the resulting pattern language that shows unresolved forces and other loose ends, signals areas where progress can still be made. If the software is far from mature, it will be impossible to mine a coherent pattern language out of it. As the software matures, the connections between it's patterns will gradually start to show.