distinct levels, without the excessive creation of subclasses. We are also taking the first steps towards the definition of a pattern language, in order to make it easier the comprehension of the architecture.

5. Conclusions and Future Work

This paper discussed how design patterns have helped to evaluate the OO Architecture for DSS framework-based development presented [BEC93][BEC96]. Though the initial analysis encompassed a specific set of frameworks designed for Capital Budgeting domain, we were able to generalize the problems found for the whole architecture. Basically, the main problems found were: a) frameworks represented analysis (real world) concepts, and not a design targeted at reuse, b) emphasis on white-box frameworks, c) lack of standard interfaces to connect frameworks of different parts, and d) lack of adequate documentation. The paper has illustrated how analysis concepts and white-box frameworks were redesigned as a single design black-box framework, and discussed the flexibility found in this transformation.

We are currently redesigning all frameworks involved in the Capital Budgeting case study, with the use of design patterns, and very interesting result were obtained so far. We are also studying the definition of standard interfaces for redesigning the DSS architecture, as well as its documentation in the form of pattern languages.

So far, the resolutions of models have not been considered in the architecture, and future work includes the investigation of this aspect using commercially available mathematical packages, like Microsoft Excel spreadsheet, and middleware integration frameworks, like OLE or CORBA.

6. Bibliography


