Conceptual Modeling and Application Integration in CAD: The Essential Elements

James Daniel Snyder II

Submitted to the School of Architecture
of Carnegie Mellon University in partial fulfillment of
the requirements for the degree of Doctor of Philosophy
School of Architecture
Carnegie Mellon University
May 1998

Advisory Committee:

Professor Ulrich Flemming, Department of Architecture (Chair)

Professor Steven Fenves, Department of Civil and Environmental Engineering

Doctor Duvvuru Sriram, National Institute of
Standards and Technology

Abstract

A research focus in design research has been the exchange of information between different participants in the design process. While information system automation has occurred in various areas, known as *islands of information*, significant software integration has yet to emerge.

A current belief among researchers in this area is that support for information sharing will require shared resources, and more specifically, shared descriptions of the information to be exchanged. If buildings are viewed as a product, the notion of a product and process modeling system ought to support the electronic exchange of information between various design process participants.

While significant research has been done, no consensus has emerged as to a satisfactory solution to design information exchange. Many important contributions have been discovered, however, no overall strategy has emerged that embraces both the research issues as well as the practical issues surrounding information exchange.

To address the above issues in a specific context, a series of experiments were conducted utilizing a prototype modeling framework that supports product modeling via the Object Model Language (OML). The results of these experiments along with a literature survey allowed for a comprehensive set of product/process modeling requirements.

The resulting requirements were then formalized into a product/process modeling environment that includes a modeling language called SPROUT (supported by a compiler) and an associated software architecture that can be targeted toward many different hardware and software platforms.

A particularly unique capability supported in this environment is formal support for integrating existing software systems. Given a schematic description in SPROUT, a formal specification can be used to generate computer programs that provably map data to and from the application program.