A Representation Scheme for Integrated Building Performance Analysis

Ardeshir Mahdavi[1], Georg Suter[1], Robert Ries[2]
[1]Vienna University of Technology
Department of Building Physics and Human Ecology
Vienna, Austria
[2]University of Pittsburgh
University of Pittsburgh School of Engineering
Civil and Environmental Engineering Department
Pittsburgh, USA

ABSTRACT

This paper presents a representational scheme for integrated building performance analysis. The underlying research work was motivated by the need for seamless exchange of structured design information. A comprehensive and widely accepted industry standard suitable for exchanging design information among the various AEC (Architecture/Engineering/Construction) applications has yet to emerge. As a contribution to this on-going discussion, we present a specific approach to the integration problem in building product modeling. This approach can be viewed as pragmatic or bottom-up in the sense that it was driven by the informational needs of related individual domains (particularly in the early stages of design) rather than by a quest for a universally applicable solution. In this paper, we describe a schema which emerged from the SEMPER effort, a multi-year project aimed at supporting detailed performance analysis for early design in the energy, life-cycle analysis, lighting, and thermal comfort domains. This schema relies on a representational division of labor between a shared building model, and various disciplinary (domain) models. Specifically, we present a documentation of the shared object model together with disciplinary models for the energy, light, acoustics, and life-cyle assessment domain.

Keywords: building product models, building performance, integration

1 INTRODUCTION

It has been stated frequently that there would be numerous advantages to integrating multiple domain applications for building design and analysis in a unified computational design support environment. Such integration would allow, amongst other things, to eliminate building model input redundancy, maintain the design consistency in a collaborative design process setting, and provide timely performance evaluation feedback to building designers and engineers. Yet, the corresponding research efforts have not resulted in satisfactorily functioning integrated system for simultaneous building design generation and behavioral (performance) analysis. Few integrated systems have actually left the research bench, let alone become a reality in the building delivery process. In previous publications we have discussed potential contributing factors to this state of affairs (Mahdavi 2000). To address some of the impediments in the path of integration, we initiated the SEMPER project, which resulted in a prototypical computational environment for integrated building performance modeling (Mahdavi 1999). The SEMPER system aimed to realize a unique synthesis of a number of features, including:

- A methodologically coherent (first-principles-based) performance modeling approach, possibly throughout the entire building design process.
- Seamless and dynamic communication between the simulation models and a general building representation. Applications should ideally obtain the necessary information for their domain analysis directly and without user intervention from a shared building scheme.
- Comprehensive (multi-disciplinary) performance modeling capability. The multi-disciplinary behavioral analysis requirement responds to the multi-actor nature of building design, construction, and operation activities.

In this paper we focus primarily on a documentation of the data models in SEMPER. This should provide other researchers in this area with information on the general and disciplinary representational schemes in the SEMPER system, and how they facilitate the integrative features of this environment.

The Unified Modeling Language (UML) diagrams are used to describe object models for SOM and three DOM's (UML 2002). These diagrams are complemented by brief explanations and definitions for the principal classes in each model. These definitions are neither formal nor complete. Rather, they are meant to conceptually summarize what a class represents.

2 SYSTEM ARCHITECTURE

The overall architecture of the SEMPER environment is schematically illustrated in Figure 3. It involves the user-interface, the shared object model (with an associated geometry reasoning kernel for real-time derivation of the topologic relations between the spatial and construction components of the model), various domain object models (with associated technical computational codes), and databases for project, building, materials, and context information.

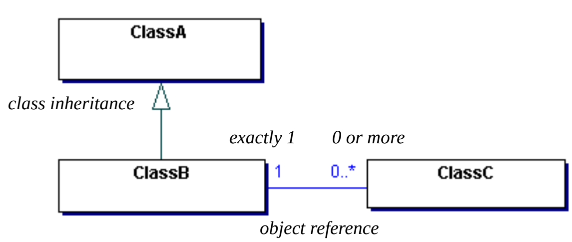

Figure 1: **UML notation**

SEMPER's representation scheme involves a shared object model (SOM) and multiple domain object models or DOMs (Figure 2). The underlying reasoning behind SOM was based on the supposition that, at some basic level (preferably the one at which design descriptions are first communicated to the medium), some shared notation of the constitutive building entities and their topological interrelations was sine qua non. However, we did not believe that this notation could be conceived as an ontological necessity to be accessed via meditations on the nature of the building as such. Rather, it had to evolve and had to be tested in the context of requirements of the "down-the-line" manipulators of the entities encapsulated by such a notation system. In the context of the SEMPER project, requirements pertaining to the integrity of the shared building model on one side, and the operational effectiveness of domain models on the other side, informed and inspired both, via an iterative process, resulting in a particular and empirically testable integration effort (Mahdavi 2000).

As to the domain models, SEMPER incorporates various building modeling and evaluation tools. In the present documentation we focus on the DOMs for energy, lighting, and life-cycle assessment. Note that:

- While this list of applications does not include all disciplinary views involved in the building delivery process (may be not even a representative sample of such views), it does represent a wide variety of informational interests, simulation technologies, and terminological idiosyncrasies.
- Almost all of the algorithms adapted for these applications are detailed and first-principles-based, i.e. they require detailed and rich descriptive information to be executed. The behavioral modeling tools, in particular, are sensitive to buildings' contextual dynamics and the topological relationships of their constituent elements.

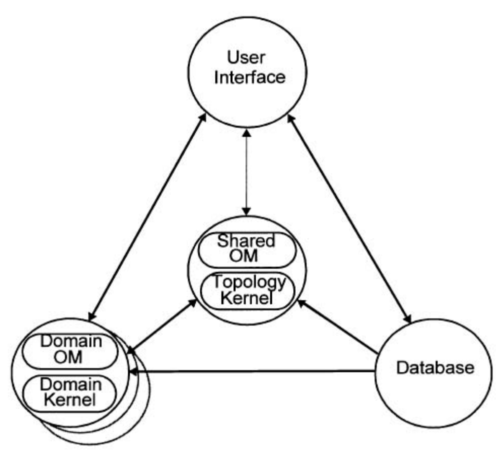

Figure 2: **The general architecture of the SEMPER environment**

3 SHARED BUILDING REPRESENTATION (SOM)

SEMPER is "space-based" in the sense that the notion of space is an important part of the way multiple applications in the SEMPER environment relate to a shared scheme of constitutive building components. Space seems to be such an intuitive entity. Nonetheless, for a long time, the space element has been absent in conventional CAAD representations. The first attempts to transcend the primitive geometry notions of points, lines, and polygons in CAAD representations focused on the introduction of "architectural" objects such as walls, windows, stairs, etc. Only recently, a rather limited understanding of the spatial entities is beginning to emerge in CAAD representations. While this lag seems striking, given the persistence of space-based references in architectural discourse, we should remember that spaces are of lesser importance in the construction vocabulary, where material objects dominate. Moreover, certain applications may need only an informationally sparse description of space (e.g. for facility management and organizational planning purposes), which may be contained in the rudimentary space information provided by some commercial CAAD tools.

Be that as it may, a close look at the informational requirements of a large class of behavioral analysis applications in architecture (e.g., energy, light, and sound simulations, thermal conditioning systems) clearly shows their need for a rich description of space, containing geometric/topologic and semantic information on bounding enclosure components (which may include openings and other sub-components with variable behavioral attributes), orientation in the overall building and site context, and adjacency relationships to other spaces (both external and internal). First-principles based behavioral prediction tools need this information. A large group of algorithmic strategies in behavioral analysis of buildings involves performing operations on discrete and well-defined (non-ambiguously bounded) spatial compartments or finite control volumes. In particular, the class of balance computations (be it in thermodynamics and heat transfer domain, be it in mass-transfer phenomena involved in air flow analysis, be it acoustic energy distribution and decay in reverberant spaces, etc.) require such discrete spatial entities and their topological interrelationships. The SEMPER project included a space-based representation that provides the necessary condition to cater for the informational needs of a large class of analytical applications from a single shared building model (Mahdavi 2000).

This shared object model in SEMPER is a hierarchically structured template (a class hierarchy in object-oriented programming terms) to capture the essential elements of a building and their properties, to the extent required by the simulation applications in the SEMPER environment. In our experience, a shared building model can be arrived at for a number of technical analysis applications and for performance inquiries of a certain range of informational resolution. Moreover, this shared model in itself does not contain the entire building information. Rather, it contains a tightly structured "notation" of constitutive building elements, with pointers to (addresses for) the detailed information on such elements in the data repository for the persistent storage of such information. While the shared object model may allow retrieving all the necessary building geometry, material, and context information that the SEMPER applications require, it is not sufficient on its own for a building performance simulation application to function. For each disciplinary domain, the simulation application's representation, or the "Domain Object Model", must be generated upon filtration and modification of information in the shared model according to the specific view of the building in that domain. Furthermore, domain specific entities (e.g. finite control volumes in numeric heat and mass transfer computation) may have to be added to what is inherited from the shared model.

The mapping process from the shared object model to domain object models is treated in section 7 of this paper in more detail.

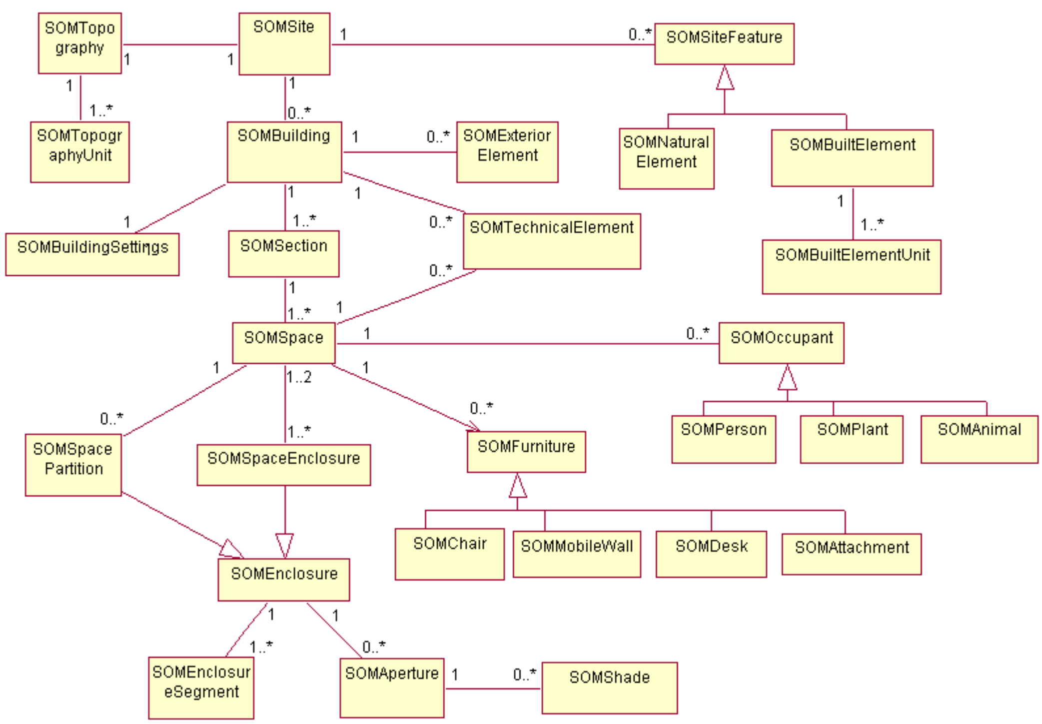

Figure 3: **Shared Object Model (SOM) class diagram**

Table 1: **Overview of Shared Object Model (SOM) classes**

Class	Definition
SOMSite	A site is the top-level physical entity consisting of a topography (terrain), and site features (natural or man-made elements placed on the terrain). One or more buildings are associated with a site.
SOMSiteFeature	Natural (e.g. trees) or man-made elements (e.g. walls) that are part of a site.
SOMTopography	The topography or terrain is attached to a site and decomposed into polygon-shaped units.
SOMTopographyUnit	An individual polygon which is part of a site geometry. Individual properties can be attached to each polygon (e.g. reflectance,).
SOMExteriorElement	Attached to the building exterior. An example are shades spanning across a facade.
SOMBuilding	A building is a logical unit which is part of a site and decomposed into sections. Settings attached to a building are used to assign, for instance, default wall constructions or windows
SOMSection	A section is an arbitrary grouping of spaces. Each space belongs to one and only one section. An example are spaces with identical elevation forming a floor.
SOMSpace	A space is the volume enclosed by a collection of surfaces that form a closed polyhedron. The polyhedron can be thought of as an virtual boundary without material properties. It contains furniture, occupants, and technical elements that are relevant for simulations, such as light fixtures and thermostats.
SOMSpaceEnclosure	Space enclosures are the physical boundaries of a space. Examples are ceiling, floor, or wall elements. An enclosure element has a polygon geometry and separates either 2 internal spaces, or an internal space and the exterior.
SOMSpacePartition	A space can further divided by space partitions. Dividing walls, raised floors, and suspended ceilings are modeled as space partitions.
SOMEnclosure	Enclosures constitute the physical boundary of a space. An enclosure may be decomposed into one or more enclosure segments or apertures. The distinction between apertures and enclosures is crucial for all SEMPER modules.
SOMAperture	An opening in a space enclosure, for instance, a window, door, or skylight. An aperture has polygon geometry.

Table 1: **Overview of Shared Object Model (SOM) classes**

Class	Definition
SOMEnclosureSegment	All non-aperture enclosures of a space. An enclosure segment has polygon geometry.
SOMShade	Shades are external or internal shading elements and can be either dynamic (blinds) or static (external light louvers). Shades are associated with apertures.
SOMFurniture	Furniture is contained within a space. Examples are chairs, desks, and storage. Furniture is modelled as a collection of surfaces with relevant properties.
SOMOccupant	Occupants are associated with a space. Occupant density and occupancy over time are considered.
SOMTechnicalElement	Technical elements are those entities that make up the space conditioning infrastructure. Examples are light fixtures, diffusers, chillers, and air ducts.

4 NODEM - ENERGY SIMULATION MODULE

NODEM is a heat-balance based thermal modeling tool that has been developed to address the need for detailed simulation even in the early stages of design (Mathew and Mahdavi 1998, Mahdavi and Mathew 1995). Accordingly, NODEM uses the same heat-balance techniques as detailed heat-balance simulation programs (Clarke 1985), but it is designed in a manner that allows for a ‘coarse’ resolution of the building. In NODEM, spaces are discretized utilizing a three-dimensional grid corresponding to the space boundary vertices. Each cubic grid unit is referred to as a cell, and a space is comprised of any number of contiguous cells. Each cell has a cell node associated with it. Walls are discretized into a number of layers and wall nodes are associated with the layer interfaces. Because of the grid structure, each cell node is coupled to exactly six other nodes, each of which is either a wall nodes, or a cell node.

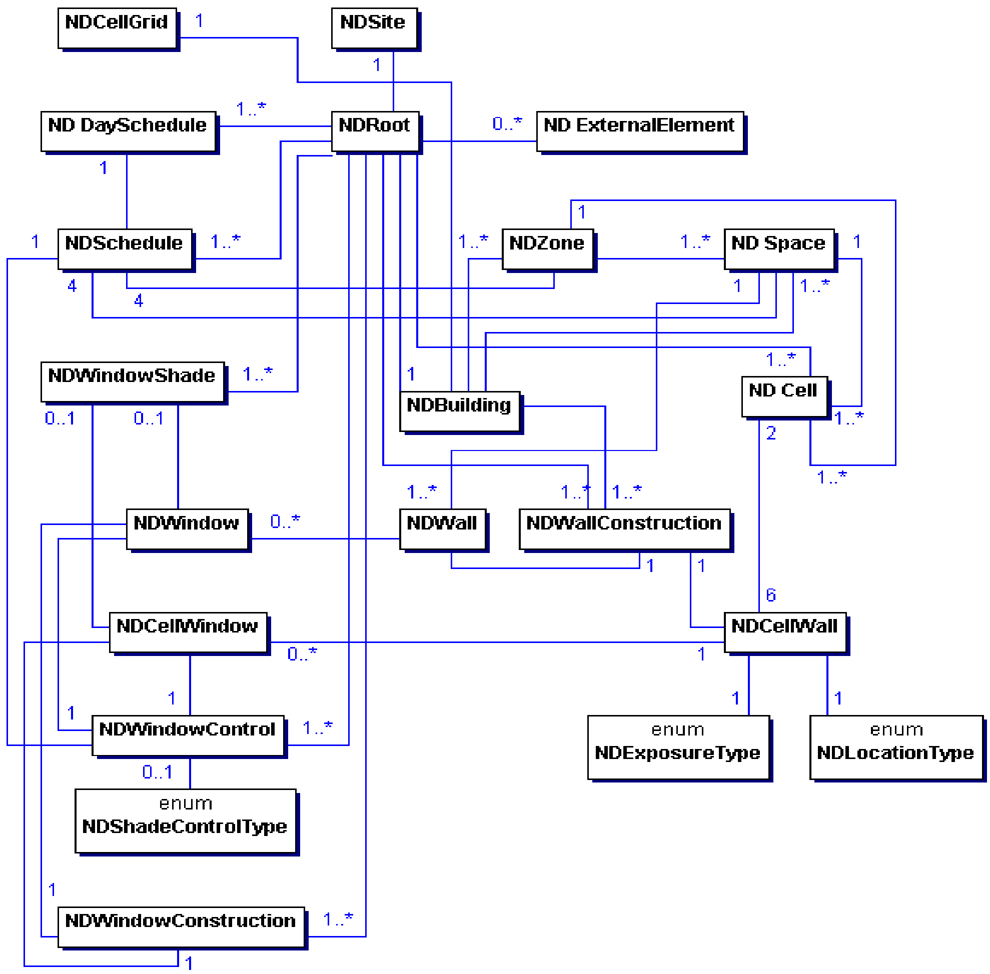

Figure 4: **NODEM class diagram**

Table 2: **Overview of selected NODEM classes**

Class	Definition
NDCellGrid	A cell grid is a three-dimensional arrangement of cells derived automatically from a set of space polyhedrons. The resolution is determined by the grid size set by a designer.
NDZone	A zone is an aggregation of spaces. The grouping of spaces can be arbitrary, that is, space arrangements do not necessarily need to form continuous clusters.
NDSpace	A space is decomposed into cells and walls. Relevant parameters are defined at the space level, for instance, electrical power, equipment power, number of people, air infiltration rate, etc.
NDWall	A wall includes a list of windows and a wall construction, but does not have a geometry. An boolean attribute indicates if a wall is adiabatic.
NDCell	A cell is a space fragment. Its 6 sides form a parallel, rectangular cube. The volume of a cell consists of air, and the sides can be made of a solid or air material.
NDCellWall	Cell walls constitute a cell's boundary. Each cell wall might include one or more windows. A cell wall has associations to the 2 cells it separates.
NDExposureType	A cell wall might be exposed to the outside, ground, air, or be adiabatic.
NDLocationType	Location type refers to the orientation of a cell wall. Location can be up, down, left, right, above, or below.
NDWindow	A window has a location with respect to a wall, a width and a height.
NDCellWindow	A cell window has a location with respect to a wall, a width and a height.
NDWindowShade	A window shade is described in terms of solar and visible transmittance values.
NDWindowControl	Shades can be deployed dynamically according to a shade type. The types defined are no shade, irradiance-based (shades are deployed when a target irradiance level is exceeded), or schedule-based.

5 LUMINA - LIGHTING SIMULATION MODULE

The program LUMINA (Pal and Mahdavi 1999) is used for the prediction of light levels in architectural spaces. LUMINA utilizes the three component procedure (i.e. the direct, the externally reflected, and the internally reflected component), to obtain the resultant illuminance distribution in buildings. The direct component is computed by numerical integration of the contributions from all of those discretized patches of the sky dome that are "visible" as viewed from reference receiver points in the space. Either computed or measured irradiance values (both global horizontal and diffuse horizontal irradiance) can be used to generate the sky model. External obstruction (i.e. light redirection louvers) are treated by the projection of their outline from each reference point onto the sky dome and the replacement of the relative luminance values of the occupied sky patches with those of the obstruction. A radiosity-based approach is adopted for computing the internally reflected component. The results generated by LUMINA have been compared with measurements in different rooms and display a generally good agreement.

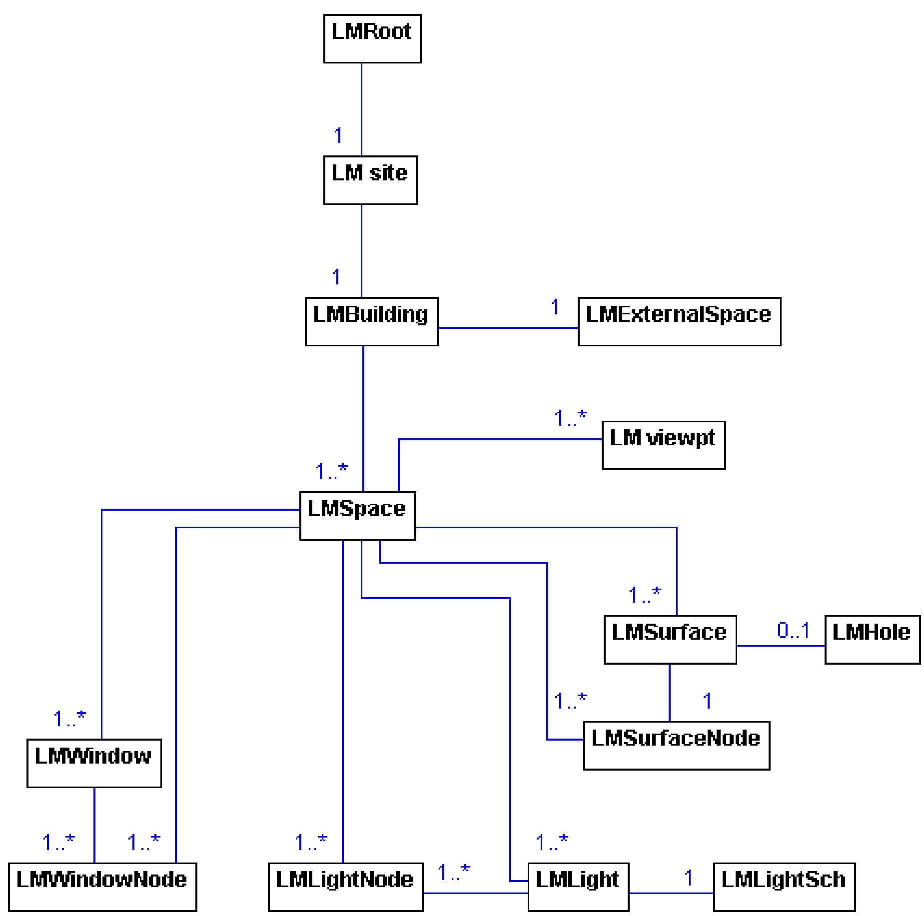

Figure 5: **LUMINA class diagram**

Table 3: **Overview of selected LUMINA classes**

Class	Definition
LMSpace	A space is composed of polygon surfaces, windows, lights, a viewpoint and corresponding nodes. The polygons make up a polyhedron.
LMViewPt	Point of observation in the space. Used for glare calculations.
LMSurface	A surface has polygon geometry and is decomposed into surface nodes. A surface might contain one or more holes. Surface properties are reflectance and transmission.
LMSurfaceNode	Radiosity and luminance properties are determined for each surface node, a patch that is part of a surface. Reflectance and transmission properties are derived from the corresponding surface.
LMHole	Holes are polygons that are subtracted from a surface. Each hole geometry typically corresponds to a window geometry.
LMLight	A light is a virtual source and contains the settings and properties (e.g. photometric properties) associated with a group of light nodes.
LMLightNode	Light nodes represent light sources in a space. A light node has a location. Settings and properties are derived from an associated light.
LMWindow	A window is defined as a polygon and comprised of window nodes.
LMWindowNode	Window nodes are discretized surface patches making up a window surface. lum

6 ECOLOGUE - LIFE-CYCLE ASSESSMENT MODULE

Ecologue is a life cycle environmental impact assessment analysis module (Ries and Mahdavi 2001, 1999). The Ecologue domain model combines a building model, an activity model, and an environmental model. The building model is automatically derived from the SOM and information from the SOT databases. The activity and environmental models are a combination of a representation of the processes and emissions occurring in the life cycle of buildings and an impact assessment model. The impact assessment model utilizes impact factors or a context model of the physical characteristics of a region and a sub-regional fate and transport model based on the fugacity concept (Mackay 1979, Mackay 1991, Mackay and Patterson 1981). The integrated simulation framework allows the environmental model to utilize other analysis modules to include operational effects such as heating, cooling, and lighting in the analysis.

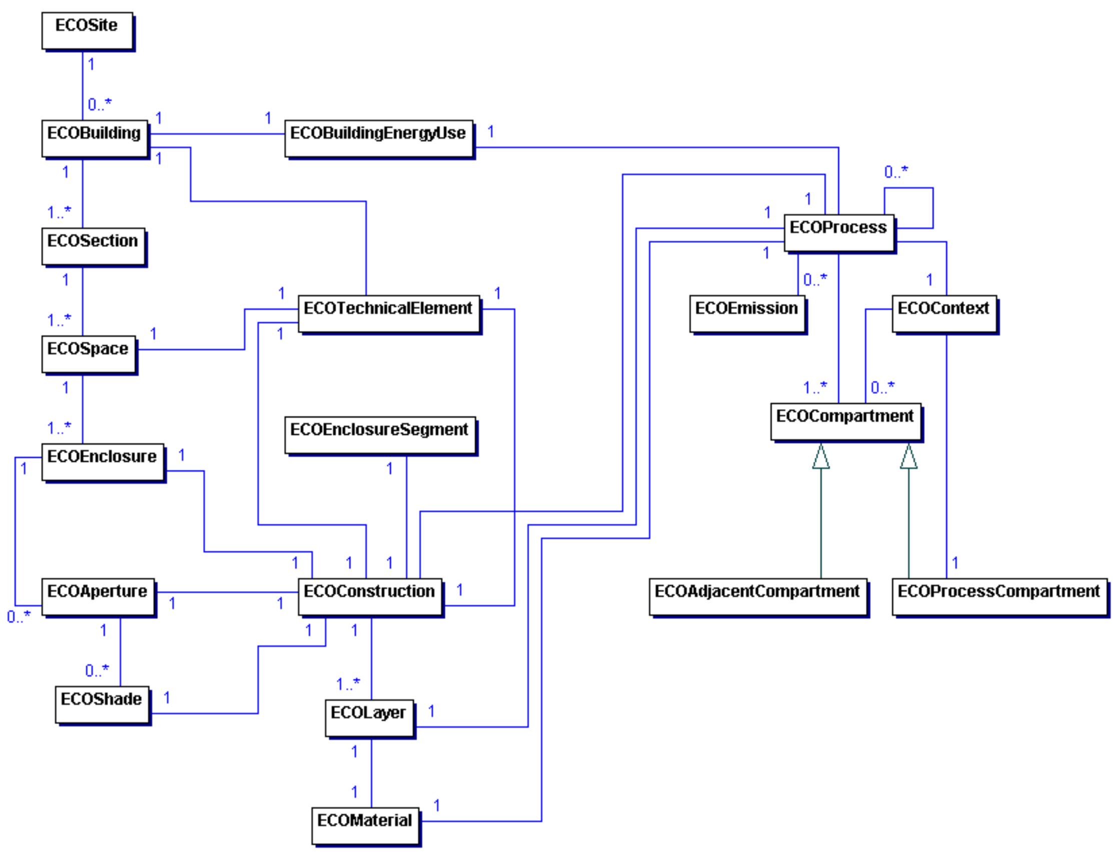

Figure 6: **ECOLOGUE class diagram**

Table 4: **Overview of selected ECOLOGUE classes**

Class	Definition
ECOSite, ECOBuilding, ECOSection, ECOSpace, ECOEnclosure, ECOAperture, ECOShade, ECOTechnicalElement	These classes are derived from the SOM and represent the building. They create an object hierarchy that is traversed for the analysis.
ECOConstruction, ECOLayer, ECOMaterial	These classes represent either a unit or layer construction and are associated with a building element. The classes are instantiated from SOT databases. They contain attributes and methods relevant for environmental analysis.
ECOProcess, ECOEmission	These classes represent an activity within the building life cycle and are associated with a building element. ECOProcess represents the characteristics of the activity and is related to an ECOContext and one or more ECOEmissions. ECOEmission contains attributes that describe an emission type.
ECOContext, ECOCompartment	These classes represent the environment or region where activities within the building life cycle occur.
ECOBuildingEnergyUse	This class represents the energy use in the building. It queries the SOM application for building energy use data generated by NODEM and has an associated ECOProcess.

7 A MAPPING EXAMPLE

There is a potential trade-off between the general intelligibility, coherence, and consistency of the shared representation, and the extent to which it can relieve individual applications from adaptive operations toward their idiosyncratic computational requirements. In the course of the SEMPER development, the dialogue between the shared representation's requirements and those of the domains revealed that while the domain representations might use different internal spatial representations for their computations (e.g. a thermal zone, an airflow control volume, or an acoustical space), they might be structurally homologous (configurationally isomorphic) to the shared building model. Such homology can be exploited for (in part) automated and non-ambiguous mapping operations from the shared building model to the domain models of the applications currently incorporated in the SEMPER environment (Mahdavi 1999, Mahdavi and Wong 1998, Mahdavi and Mathew 1995). The homology-based mapping uses the configurational isomorphism between the shared building model and various domain models to derive the latter from the former. We have not made a claim, however, that such mapping technology would work for all domains and independent of the informational resolution of the pertinent inquiries. Instead, we have argued that the question, if and to which extent such mappings can be automated, must ultimately be decided on an empirical basis. For a certain class of applications working with information of a certain level of resolution, the SEMPER project has demonstrated the feasibility of homology-based mapping.

As an example, figures 6 and 7 illustrate the mapping process for NODEM (cp. section 4). In this case, two types of mapping are of interest (Mathew and Mahdavi 1998):

- Object-Object mapping: An object in the shared model has a corresponding object in the domain model (e.g. the 'Space' object in SOM has a corresponding 'Space' object in NODEM). However, these objects do not necessarily have an identical set of attributes.
- Object-Attribute mapping: An object in the shared model is transformed into attributes of an object in the domain model (e.g. the 'Occupant' objects in SOM are transformed into attributes of the 'Space' object in NODEM).

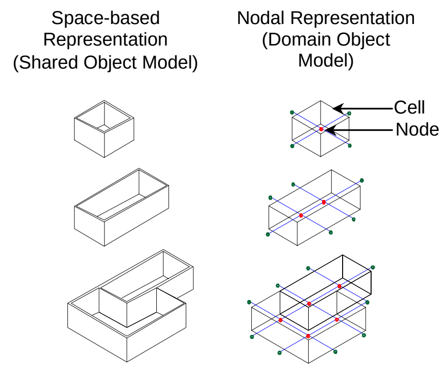

Figure 7: **Illustration of representations in SOM and NODEM**

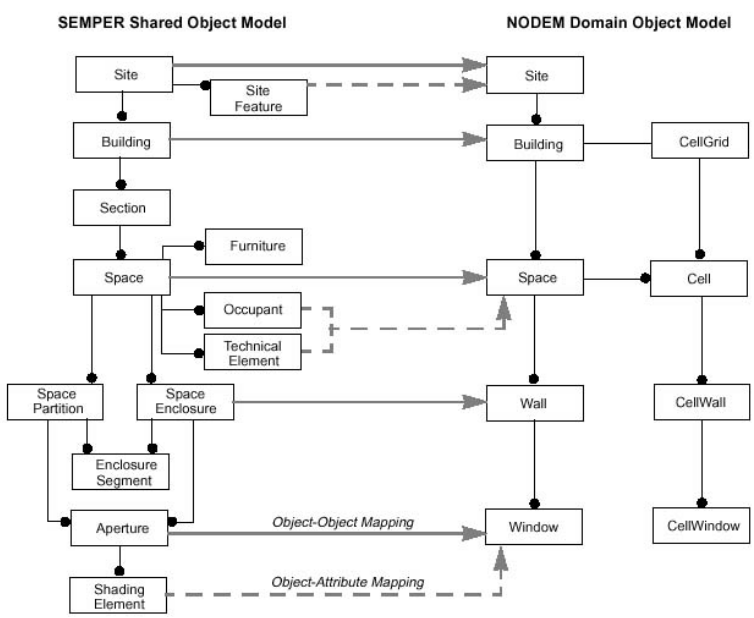

Figure 8: **Schematic illustration of mapping between primary objects in SOM and NODEM's domain object model**

8 DISCUSSION

The SEMPER project had its limitations, some organizational, some thematic. As a research project it could not deliver more than a prototype and a proof of concept for a limited set of design support functionalities. A subsequent research effort (the S2 project) demonstrated that the SEMPER environment could be realized in terms of a geographically distributed seamless computational environment (Mahdavi et al. 1999). But a commercially viable realization has not emerged. An important reason for this state of affairs may lie in the mismatch between design information representation in commercial CAAD and in SEMPER. As long as CAD design documents do not meet the criteria of non-ambiguity, integrity, completeness, and consistency, one cannot do without the corrective, interpretative, corrective, and complementary role of an (naturally or artificially) intelligent agent. As long as the artificial versions of such agents are not up to par, mapping operations cannot be fully automated (Mahdavi 2002).

However, despite its limitations, the SEMPER project did demonstrate that, for a certain applications and for queries of a certain level of detail, a well-balanced representational labor division between a reasonably detailed shared building model and a number of behavioral domain models is possible, and that the latter can autonomously infer their informational requirements from the former via mapping operations. A likely question is, of course, if and to which extent this integrative framework could be expanded to accommodate other applications and other levels of detail (in application queries). Given what we have learned about the problems and perils of integration, it would be best to approach this question empirically and on a case-by-case basis.

It is likely that a grand unified representational scheme for the consideration of all aspects in building design and construction domain is unattainable. But there is much room for improving the informational fidelity and communicative effectiveness of building models.

9 REFERENCES

Clarke, J. (1985) Energy simulation in building design, Bristol, Boston.

Mackay, D. (1991) Multimedia environmental models: the fugacity approach, Lewis Publishers, Chelsea, MI.

Mackay, D.(1979) Finding fugacity feasible, Environmental Science and Technology **13 (10)**, 1218-1223.

Mackay, D. and S. Patterson (1981) Calculating fugacity, Environmental Science and Technology **15 (9)**, 1006-1014.

Mahdavi, A. (2002) Notes on representation in computer-aided architectural design. (Unpublished manuscript).

Mahdavi, A. (1999) A comprehensive computational environment for performance based reasoning in building design and evaluation, Automation in Construction **8** (1999) pp. 427 - 435.

Mahdavi, A. (2000): Supporting collaborative design via integrated building performance computing, InterSymp-2000, 12th International Conference on Systems Research, Informatics and Cybernetics, Baden-Baden, Germany. Proceedings: Advances in computer-based and WEB-based collaborative systems. ISBN: 0-921836-88-0. pp. 91 - 102.

Mahdavi, A., and N.H. Wong (1998) From building design representations to simulation domain representations: an automated mapping solution for complex geometries, Comuting In Civil Engineering; Proceedings of the International Computing Congress, 1998 ASCE Annual Convention. PP. 1-10. October 18, 1998.

Mahdavi, A., M. E. Ilal, P. Mathew, R. Ries, G. Suter, and R. Brahme (1999) The architecture of S2, Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. III. ISBN 4-931416-03-9. pp. 1219 - 1226.

Mahdavi, A., and P. Mathew (1995) Synchronous generation of homologous representations in an active multi-aspect design environment, Proceedings, Fourth International Conference, International Building Performance Simulation Association, Madison, WI, pp. 522 - 528.

Pal, V., and A. Mahdavi (1999) A comprehensive approach to modeling and evaluating the visual environment in buildings, Proceedings of Building Simulation '99. Sixth International IBPSA Conference. Kyoto, Japan. Vol. II. ISBN 4-931416-02-0. pp. 579 - 586.

Ries R., and A. Mahdavi (2001) Integrated Computational Life Cycle Assessment of Buildings, Journal of Computing in Civil Engineering **15 (1)**, pp. 59-66.

Ries, R., and A. Mahdavi (1999) Environmental life cycle assessment in an integrated CAD environment: the ecologue approach, Computers in Building: Proceedings of the CAADfutures '99 Conference, Atlanta, Georgia, June 7-8, 1999.

UML (2002) http://www.omg.org/uml