TOWARDS MODEL BASED DESIGN – A CASE STUDY: THE MODULAR DESIGN SYSTEM

Towards model based design

E. D. Griffith, D.K. Hicks, K. D. McGraw, and M. P. Case
US Army Construction Engineering Research Laboratories, Champaign, IL, USA

Durability of Building Materials and Components 8. (1999) *Edited by M.A. Lacasse and D.J. Vanier*. Institute for Research in Construction, Ottawa ON, K1A 0R6, Canada, pp. 2579-2589.

Abstract

The U.S. Army Corps of Engineers has developed a tool called the Modular Design System (MDS) to assist design professionals in the processes of planning, design, and construction document preparation for repetitive facility types. The use of early versions of MDS has demonstrated a reduction in time by nearly two-thirds typically required to design and award a construction contract. Initially developed to support Army Reserve Training Centers, the USArmy Corps plans to expand its use over a wider range of repetitive facility types.

The current implementation is a hybrid document/model approach consisting of electronic drawings linked by an external database. Data consistency issues associated with this architecture limit its scalability. To meet expanded requirements, the USArmy Corps is developing a model based information approach utilizing emerging commercially available object based CAD systems. This redesigned information infrastructure marks a fundamental change from an implicit to an explicit model-based representation.

Three key capabilities make MDS a powerful tool. First, the ability to capture and reuse corporate design criteria at the architectural function level. Second, it provides an integration framework for engineering analysis. Third, it manages and integrates the contract document production.

The underlying MDS information infrastructure will move towards a model based approach. Future work will focus on collaborative processes such as conflict resolution and design review. Additionally, MDS offers the opportunity to transfer an information rich model downstream to operations and maintenance.

Keywords: model based design, facility design, object-oriented modeling, design criteria capture and reuse, integration framework, construction document production, computer aided design (CAD), industry foundation classes (IFC)

1 Introduction – MDS today

The Modular Design System (MDS) is a computer software add-on application to a commercial computer aided design (CAD) system that supports the planning and design of facilities (Bonham 1997). Working from pre-engineered modules (architectural functions), designers can quickly assemble design solutions. MDS supports a spectrum of facility types to varying degrees of completeness. When a facility type is decomposed to the reusable function level, many similarities between facility types become apparent. Functional overlap between facility types can range from 50 to 95 percent. Additional building types can easily be supported through the addition of new modules (functions) that support their unique requirements. Currently used by the Army Reserve and National Guard, MDS is being deployed for military construction and work for other government agencies.

MDS supports design activities from project requirements to construction document development. Each module contains information related to the design criteria, cost and specification requirements of its particular architectural function. This provides for a more accurate cost estimate and reduces the possibility for errors and omissions between drawings and specifications. Utilizing a charrette process, owners and users can more accurately identify facility requirements and transition into design production seamlessly. The result is a reduction of delivery time by as much as two-thirds, reducing the gap between identifying the users needs and satisfying them.

The current version of MDS (2) is implemented on top of MicroStation 95 using the MicroStation Development Language (MDL) (Bentley 1998). The technique utilized to capture information about design components is to link MDL primitives and graphical elements through resource files. The resulting building models are contained in a collection of drawing, relational database and ASCII "resource" files. Additionally, the model can be exported into Microcomputer Corps of Engineers Automated Cost Estimating System (MCACES), which associates cost data with MDS components.

While the current implementation has demonstrated success within the Army Reserve and National Guard construction program, expanding requirements warrant additional functionality. The main barrier to addressing these requirements is at the data level. The association between graphical elements and data in resource files has to be explicitly maintained. Each piece of data requires a unique software routine to maintain data integrity. This limitation exists for two reasons. First, the technology available at the time of development would not support a more robust implementation. Second, MDS was originally developed to support only two facility types, Army Reserve and National Guard training centers. A change in the underlying data infrastructure is required.

2 MDS 3.0 - Future framework implementation

Today's facility design process results in a number of application-specific facility models. Transferring information from one model to the next requires recreating the desired information by hand. Often the information does not exist in a computable form, rather, it can only be identified through human interpretation of lines and arcs. A robust engineering model is a sufficient informational model of the facility design and requirements supported in a computational environment. Such a model is important from a Corporate/Organization information management perspective. In addition, it opens the door for a design automation suite in which the appropriate level of information access and reuse can occur.

MDS 2 supports partial information reuse through a set of linked graphical elements, resource files and databases. While the current MDS 2 does not utilize a complete robust engineering model, the partial model information successfully allows reuse and adaptation. Users can actively manage design information at the functional level. This allows the information to be managed and reused corporately. It provides a means to maintain a consistent level of quality and reduces costs of design, acquisition and maintenance. But the current approach is ad hoc and fragile.

A change to an implementation framework is necessary within which applications develop the robust engineering model (Pree 1995). Frameworks are the top-level abstraction utilized by software engineers. At the framework level business rules are defined which can interact with expected behavioral patterns. The software abstraction in a framework is an object class definition, our robust engineering model.

To meet the expanding requirements cost effectively and prepare for future capabilities; a new information infrastructure framework (MDS 3) is being developed. The needs to satisfy organizational and end user requirements will be met with two fundamental changes to MDS 2: a more robust engineering model and a strong implementation framework. The MDS 3 framework will provide a means to manage corporate information about design criteria, design solutions and life cycle facility models. It will be an information-modeling infrastructure which supports cradle to grave facility delivery and operation.

3 Design criteria – Representing explicit requirements

MDS 2 uses the concept of programs and functions as the principle way to manage design requirements and criteria. A program is a collection of functions. Functions contain explicit design criteria for specific activities. By accessing and utilizing specialized design criteria at the architectural function or activity level, MDS 2 streamlines the documentation process and quickly produces an initial design solution ranging from 25 to 90 percent complete depending on discipline, thereby reducing delivery time. As the MDS implementation migrates to an information management rich environment, program and function information can be captured and managed explicitly.

3.1 Functions

The concept of *function* plays an important role within MDS 2 and it is necessary for MDS 3**.** A function specifies the requirements of activity areas within a facility. Requirements can be either prescriptive or performance based (Donia 1998). The characteristics and behavior of specific design criteria for a particular function may vary by organization.

In MDS 2, there is a one-to-one correspondence between a function and a room (IdeaGraphix 1996). The MDS 2.0 uses a static implicit description of rooms that represent the completed room as a graphical collection. Information is included or excluded through implicit graphical elements contained within graphical groups that are layered to create a specific function/requirements set. Related information such as cost and method of construction is supplied by the function description in tabular form contained within resource files. Resources files and graphical elements are independent of each other and consistency is maintained manually. Figure 1 illustrates the composite completion of a typical office function within MDS 2. The discipline specific design criteria is captured in the associated graphical companion on the right. Information such as cost models and specifications are relational table based.

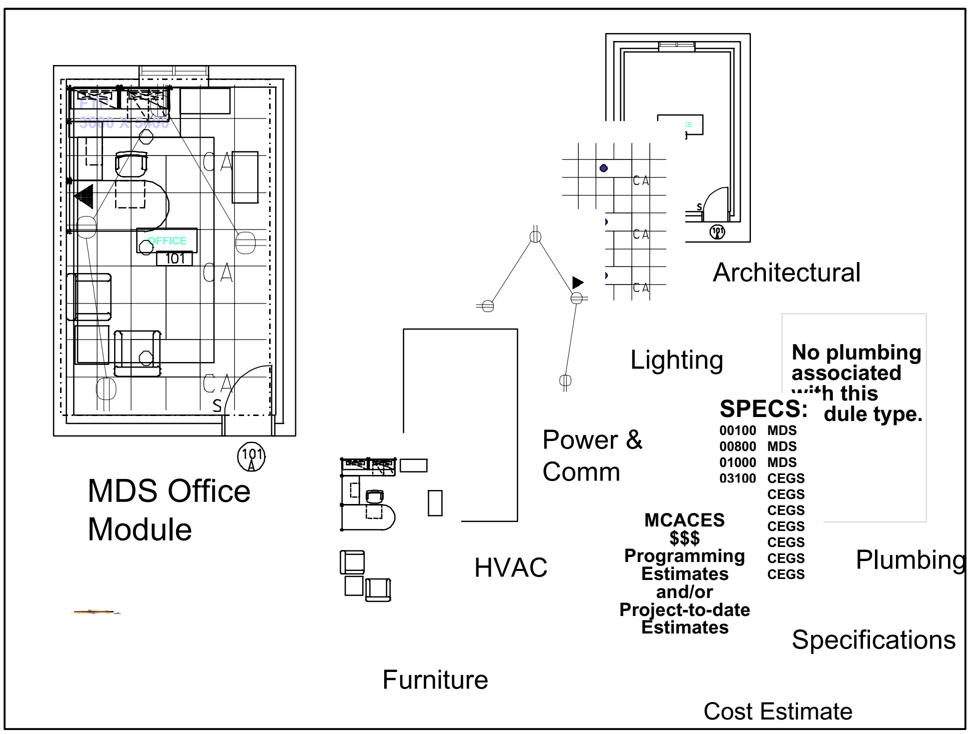

Fig. 1:Typical Office Function

3.2 Explicit representation

Moving towards a model based implementation; the collection of requirements for a function will be explicitly represented within MDS 3. With an explicit representation, the breadth of adaptability for a function to satisfy design criteria greatly improves across facility types and within localized solutions. The types of requirements that will be represented are:

- Occupancy – 24 hour schedule. An activity area may have established patterns of occupancy.
- Spatial – minimum/maximum area, width, length, and aspect ratio.
- Ventilation Heating – minimum required air changes per hour.
- Daylighting – minimum levels of daylighting that denotes windows or skylights.
- Lighting – task lighting levels at differing heights. Each function activity has specific lighting levels at a specific height.
- Physical Characteristics – specialty items. Examples of specialty physical characteristics could be handicap access, specialty equipment, or arrangements.
- Furniture – Layout and type. Each function can have specific furniture requirements both in arrangement and type.
- Finishes – colors, materials for each surface.
- Adjacency – A function may be required to be next to another function such as a lobby next to receptionist desk.
- Budget – The total cost budget allocated to this function.

As an example, the spatial requirements for MDS functions have been represented in SEED-Layout, a module of the Software Environment for Early Phases of Building Design (SEED) (Flemming and Woodbury 95). SEED-Layout uses the explicit representation to quickly generate design alternatives and solutions through a guided exploration metaphor. MDS 2 cannot represent the requirements or provide the resulting capability. Maintaining spatial relationships between functions is one of the many requirements driven design processes that needs to be managed explicitly within the framework.

4 The MDS 3.0 framework model

The framework model for design automation is comprised of three main components. The premise of the model is to capitalize on commercial applications as much as possible and focus on the information requirements that make the US Army Corps organization more effective. The most important component is the Facility Engineering Model Server. This layer provides the common model for all applications and the repository services that connect to a commercial database. The Discipline Interface Layer provides application program interfaces (API) that support domain specific activities such as structural analysis or HVAC design. The Support Services Layer is cross-discipline in nature. It supports corporate work-flow and information management requirements, including the ability to transform corporate design criteria into design solutions. It also manages construction document production.

4.1 Facility engineering model server

The Facility Engineering Model Server, as shown in Fig.2is the core component and provides the following capabilities:

- Database Connectivity
- Representation of Functional Requirements
- Artifacts
- Relationships
- Modules (specific configurations of Functional Requirements, Artifacts and Relationships)
- Version/Alternative Control
- Collaboration Mechanisms

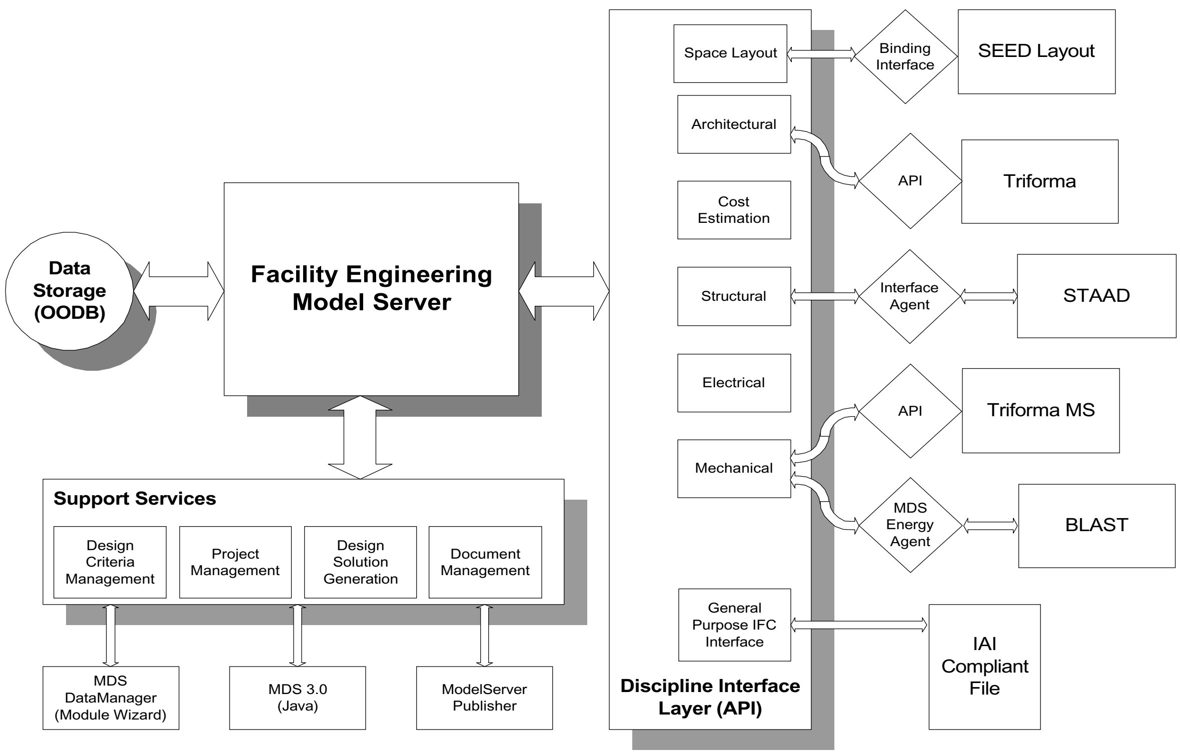

Fig. 2: Framework model

4.2 Support services

The set of services shown in the lower left of Fig. 2 facilitates the management of corporate information, processes and expertise (behavior and requirements). Support services are typically agent-based applications that provide selective domain expertise to facilitate manipulation of the engineering model. Agent applications are similar in nature to wizards used in applications like Microsoft Word. In this case, they apply corporate knowledge to facilitate design tasks. It is at this level that corporate lessons learned are incorporated into the design process, completing the feedback loop.

4.2.1 Design criteria management

This service facilitates the capture of organizational corporate information including specifications, cost models, requirement models, function models, document configuration and assemblies of building system components. Specialization provides a mechanism to distinguish the data-driven differences between differing corporate entities. For example, the criterion for the function *GenericClassroom* might apply to all corporate entities. However, the Immigration and Naturalization Service (INS) might decide that they have unique criteria for private offices. The INS would use a specialization of the *PrivateOffice* with their unique criteria modification. Additionally, the INS will need to define the requirements for a new functions that don't exist such as a *DogWash*.

4.2.2 Project management

This service supports corporate processes. Particularly important for collaborative design, it provides mechanisms to introduce change management and enforce corporate process control requirements. For example, this service can restrict access to a particular set of functions during space layout activities. This is also where collaborative communication protocols are defined. One agency might allow information to be broadcast to all design participants while another agency might require all changes to be channeled through the project lead prior to distribution.

4.2.3 Design solution generation

This set of services facilitates the application of corporate expertise (behavior and requirements), which act as a collection of Generate Agents. Each discipline and design activity requires a set of expertise to analyze the facility model and create solutions to specific design problems. These agents utilize the domain expertise and performance requirements to generate initial design solutions. These solutions include pre-placed components such as lights, diffusers, furniture, finishes and door types for a given function or activity. As with design criteria, the *GenericLighting* agent might be acceptable for the Corridor function in all facility types for all agencies. However, it is anticipated that specialized agents will be adapted to handle unique conditions such as architectural and plumbing requirements of an INS *DogWash* function. Specialization of generate agents enables a function to be adapted and reused, handling cases where specialized engineering is required.

4.2.4 Document management

This service will manage the information to produce the documents necessary for bidding and construction activities. It also facilitates new ways of interacting with the engineering model to improve customer coordination, design review and the construction process. Innovative techniques are possible by using the model to improve visualization and comprehension. The most important product produced from the end user's perspective is the Construction Document Set. These include graphic and non-graphic information (drawings and specifications). This service determines information requirements (e.g. number of drawings, specifications), collects and cross references information (e.g. schedules, details, sheet and drawing number), and publishes them (e.g. plotting or electronic bid sets).

4.3 Discipline interface layer

This layer provides the application interfaces and computational mechanisms to connect commercial applications with the Facility Engineering Model Server. Once initial design solutions for each discipline have been developed by the generate process, detailed design and analysis will be required. The degree and character of this activity will vary by organization, facility type, discipline and end user. Ideally, these activities should be supported by the principle commercial applications from which the design professional can choose. In practice, the applications depend on many factors including its functionality, data exchange capability, market penetration and cost.

4.3.1 Design and engineering domains

The following are the key discipline specific activities supported by the Discipline Interface Layer.

- Preliminary Design – Facility function requirements and preliminary layout.
- Cost Estimation – Building assembly models, work breakdown structure, unit price book.
- Architectural - Layout, massing, building enclosure, interior construction and interior design
- Electrical – Lighting, power and communications.
- Structural - Framing (vertical loads), lateral loads (bracing) and foundation (bearing).
- Mechanical – Energy analysis, HVAC, plumbing, fire protection.
- Furniture – Furniture layout, color schemes, open office arrangements.

4.3.2 Application interface mechanisms

There are several ways that a domain application can access the framework. Each method has a particular method for "delegating" responsibility for data integrity. Each application should be evaluated to determine how to best ensure data integrity.

- General Purpose IFC Interface – This service provides a file-based information exchange mechanism as defined by the International Alliance for Interoperability (IAI 1998). It will provide access to an increasing number of applications as support for this standard (and the standard itself) grows. The framework, in this case, assumes all responsibility for data integrity.
- Interface Agent - These are "wrappers" that serve as a translator between the model server and the application supported. Typically, an interface agent adds value to the design process by filtering, enhancing or adding information beyond that provided by the application. In addition, the interface assumes all responsibility for data integrity. The MDS Energy Agent and Structural Interface Agent are examples of this method.

- Binding Interface - A mapping approach useful for linking applications that have similar concepts. The interface provides an Interface Definition Language (IDL) that supports information exchange between the framework and application. In this case responsibility for data integrity may be shared between the interface and the application.
- Application Program Interface (API) – This method offers the tightest level of integration between the application and the framework. It allows applications to communicate directly with the framework. Since there is no intermediate software, the application assumes full responsibility for data integrity.

5 Document production

The state of the construction industry still requires construction documents as the final product of the design phase. MDS 2.X organizes, coordinates, produces and manages the construction documents. The documents are "extracted" from the collection of CAD files that together is the facility model. The proposed MDS 3.0 framework will still extract the information to produce the files, but the extraction will be a query against a computatable facility model. In addition, as the designer creates the engineered solution the facility model is developed without any additional effort on the designer's part and is therefore a by-product of the design process. Most architects and engineers consider their job to be the production of the documents, not the model. With this approach to document production, the facility model and construction documents go hand in hand.

In addition to the US Army Corps generated designs, a significant amount of the design work is contracted to private architectural and engineering firms. In the process of monitoring the work of both the in-house and contracted designs, the US Army Corps spends considerable time reviewing the work. Through this approach, the review process is streamlined because the requirements are explicitly represented and provide the preliminary solutions as the point of departure. Also, the facility model changes the nature of the review and understanding of the solution. Construction contractors will be able to view the facility model in non-traditional methods thus communicating the design intent more fully. This same capability will also facilitate the bidding process. The traditional process of poring over contract documents is replaced by reviewing the facility model itself. This better understanding of the facility should reduce the cost of the work by more accurate bidding and fewer errors to resolve. By changing the process of producing, reviewing, and estimating construction documents, the total quality will be greatly improved thereby minimizing costly change orders.

6 Benefits

Today's business environment is characterized by rapidly changing technology, heightened cost awareness and the aggressive measures to boost internal productivity. Designers want practical technology that enables them to get the job done. Smart use of automation technology will significantly improve design productivity, help designers apply and improve specialized engineering knowledge, and improve the product quality while building an information infrastructure that can be used to collect as-built information.

The benefits of using MDS go far beyond basic CADD tools because intelligent design "modules" allows owners to explicitly capture design criterion, insuring project quality. By utilizing COTS, owners can focus their efforts on improving the design modules, specifications and details so designers can spend more time evaluating alternatives while consistently applying owner and industry standards. People involved in construction, operations and management, maintenance, etc., can also provide lessons learned feedback to improve the design modules and avoid repetitive mistakes. Other immediate benefits include improved multi-disciplinary communication and coordination, robust modeling and engineering visualization capabilities, shortened project delivery time, capture of corporate knowledge, and a better end product. This benefit supports the US Army Corps' Vision in the areas of Revolutionizing Effectiveness, Improving Performance, and Increasing Customer Satisfaction (USACE 1998).

7 Conclusion

The current MDS implementation must be modified to meet future requirements as interests expand beyond the original scope of the MDS project. The MDS 3.0 Framework will provide an information infrastructure foundation to transition MDS into the next century. The framework will exploit standard efforts in the software industry and domain relevant efforts such as IAI and the National CADD Standard. Working with industry partners, the framework will utilize plug and play capabilities for sophisticated engineering analysis activities while providing minimum domain specific capabilities. The MDS 3.0 Framework will focus on the information modeling requirements and manage the model independent of specific applications. The model will become a life cycle corporate asset that is reusable and adaptable.

8 References

Bentley Systems, Inc. (1998) *MicroStation and MicroStation Development Language*, Exton, Pennsylvania.

Bonham, L (1997) *Reserves, Corps, Guard Pioneer New Technology*, Public Works Digest, Vol. IX, No. 2, Alexandria, VA pp. 6-7.

Donia, M.A. (1998) *Computational Modeling of Design Requirements for Buildings*, PhD. Dissertation, Carnegie Mellon University, Pittsburgh.

Flemming, U. and Woodbury, R. (1995) *Software Environment to Support Early Phases of Building Design (SEED): Overview*, Journal of Architectural Engineering, Vol. 1 No. 4, ASCE, pp. 147-152.

IdeaGraphix, (1996) *Modular Design System DBA Specifications*, IdeaGraphix, Atlanta.

International Alliance for Interoperability (IAI), (1998) *IFC Object Model Reference*, IAI, Washington, D.C.

Pree, W. (1995) *Design Patterns for Object-Oriented Software Development*, ACM Press, New York.

US Army Corps of Engineers (USACE) (1997), *Strategic Vision,* US Army Corps of Engineers, Washington, D.C.