31 ADDING FUNCTION AND BEHAVIOUR TO PROJECT MODEL OBJECTS TO INTEGRATE AEC INFORMATION

Dr. Rafael Sacks
Technion - Israel Institute of Technology
National Building Research Institute Technion Campus, Haifa 32000
Telephone: 972-4-8293190, Fax: 972-4-8324534
http://tx.technion.ac.il/~cvsacks
cvsacks@tx.technion.ac.il

Abstract

A number of trends in Information Technology (IT) indicate that the challenge of integrating information for construction projects may be served best by tight coupling of application methods with project model objects. A paradigm is proposed in which designers, contractors and suppliers process project information, which is stored on a central server, using application routines provided on-line as object-oriented methods, which reside in the central server. Development of such "intelligent" object sets will be financed by construction component suppliers; they will be based on foundation class standards, programmed by software houses, and provided on-line by project management web sites. The use of Intelligent Parametric Templates (IPT) in three sample applications suggests how Building Project Models (BPM) can support this vision of Computer Integrated Construction (CIC). The obstacles and research requirements relating to the proposed paradigm are discussed, in light of the experience gained in development and use of the IPTs.

Keywords: *Computer Integrated Construction Information, Intelligent Parametric Templates*

INTRODUCTION

The 'Year 2000 Building Product Model Coastline' (CIB W78-1994 proceedings) (Karstila et al 1994) expressed the hope and/or expectation that product modelling would mature toward 2000 and provide most of the solutions for integrating the information process in construction. However, while definite progress has been made, product model based integration remains a future goal. Meanwhile, the Internet, in the form of commercial project management web sites (PMWs), has had significant impact in industry. In addition, the way in which software is provided to end-users is also undergoing rapid change in the Internet environment, in the form of Application Service Providers (ASPs). Rather than replacing the need for building project model technology, the improvements in communication emphasize the need for it – with improved communication channels, the possibility of collaboration without co-location is enhanced, and thus the form and nature of the content that is communicated becomes of greater relevance.

In light of these developments and the shortcomings of BPM standards, it is proposed that the concept of a centralized project database, implemented as a building product model, be revised, particularly with regard to the ways in which it can be delivered and financed. The following discussion begins with a brief review of current technology, and describes a likely scenario of implementation of product models in the short term. Next, a new paradigm is proposed and described, in which application software methods are delivered together with building project model object instances, rather than as pre-sold packages. Three examples of the use of Intelligent Parametric Templates are presented to demonstrate the concept. Finally, the obstacles and future research requirements are discussed in light of the experience gained through implementing the examples.

Building Project Modelling

The process of building construction has been aptly described as consisting of two highly integrated sub-processes – an Information process and a Material process (Bjork 1999). A significant proportion of the problems that occur in physical construction activities can be traced back to failings in the information process. Information Technology (IT) is seen as holding the potential to radically improve the Information process, by enabling collaboration and integration in both design and construction. Galle (1995) defined ten desirable properties of computer building modelling systems, grouped as issues of *integration*, "*intelligence*" and *compliance*. While a wide range of technological approaches to product modelling for construction has been researched, no clear consensus has yet emerged on many basic issues. Amongst these are:

- The degree of control that should be maintained by the users, versus the level of automation that should be built into systems (Flemming 1994, Sacks and Warszawski 1997).
- How to resolve the need to provide versatility (the ability to add new building technologies, sometimes ad-hoc) with the need for industry-wide standardization of building project models (Galle 1995, Tolman 1999).
- The architecture of the databases. Centralized data models, distributed data models, hybrids of these, data transfer models (neutral file formats), and others have been investigated (Teicholz and Fischer 1994, Gielingh 1988, Yau 1992, Eastman 1992).

- Providing for concurrent design collaboration while at the same time ensuring semantic integrity of the data (Eastman and Kutay 1991, Galle 1995).
- The degree of coupling between software packages and the project model database (Bjork 1990, Fenves et al 1990)
- The possibility (and desirability) of incorporation of “intelligent” or “creative” systems with “learning” capabilities, both within application modules (Flemming 1994, Sacks and Warszawski 1997) and within the data model itself (for preserving integrity).

Two major efforts at standardization have been pursued: STEP (ISO 10303), starting with the AEC building systems model (Turner 1988) and continuing with building construction core model (BCCM – Tolman and Wix 1996); and more recently the Industry Foundation Classes (IFC) development within the Industry Alliance for Interoperability (IAI – Liebich and Wix 1999, Kiviniemi 1999). These assume no coupling with software applications and no “intelligence”, provides limited integrity constraints, and is likely to suffer from lack of versatility due to the need to update versions in incremental steps. Doubt has also been expressed as to the chances of success of standardization efforts such as STEP or IAI, or of any common model (Tolman 1999), due to the innate complexity of a unified model, the multinational nature of the construction industry, and the required level of concentrated financing. Tolman (1999) suggested that “the first practical solution to the integration problem will be realized (through providing a service) rather than the unified development of standards”.

Despite these criticisms and its limitations, the IAI effort has significant prospect for progress, as it enjoys broad industry support (Froese and Yu 1999), precisely because it adopts the simplest solution to each of the abovementioned issues. The CIMSTEEL project, based on STEP, (Watson and Crowley 1997) has been successful, and was recently adopted by the American Institute of Steel Construction (AISC).

Construction Project Management Web Sites and Application Service Providers

The use of commercial project management web sites (PMWs) has become common practice in AEC projects; such sites provide a common data repository for the CAD drawings, text documents and other data of the project (Hartung et al 2000). The site is paid for by the project owner and managed directly by the project manager. While such sites create a ‘virtual company’ for a construction project, and thus improve communication, to date they have not deployed any project data modelling technologies. Their prospects for long-term success are enhanced greatly by their potential to provide business-to-business e-commerce platforms for construction. According to one investment report “Through technology applications, the time it takes to complete a construction project declines, the efficiency within the construction time period increases and the cost of goods procurement declines, while the transparency and accountability of the process increases dramatically…We estimate that the e-construction space alone could command company values in excess of **$150 billion** by 2005” (Hartung et al 2000).

In the Application Service Provider (ASP) model (ASP Island 2000), software is installed only on the provider's server, and used on-line by the client through the Internet. Remuneration is for a service rather than for purchase of a package. This has the significant benefit of relieving the user from the need to maintain or upgrade software, or finance its purchase; the user is assured access to the most recent version of the software.

THE BASIC IFC APPROACH

A likely scenario of the use of the IFC classes for integrating information processing in construction, using project management web sites as facilitators, is shown in Figure 1. In this view, software vendors incorporate IFCs in their products, i.e. they are used as an Application Programming Interface (API) rather than as a model exchange standard (Tolman 1999). Using such programs, the AEC consultants (architects and engineers) each build and use their own local project model subsets; data objects are downloaded to and uploaded from the central repository as and when needed. Any project participant can view the project data at any time by using appropriate viewing software.

The straightforward drawbacks of this scenario are that:

- Ensuring the integrity of the project data is difficult (multiple copies of each instance are stored) and data must be 'mapped' or translated between each local Building Project Model and the global BPM wherever applications are not IFC compliant.
- Adding new building products to the system, with attendant data objects, requires updates to software, even if the new object classes inherit from the IFC classes,
- Updates of vendor software require upgrading at each different location.
- The cost of financing the integration is borne largely by the AEC consultants through the purchase of application software.

However, it suffers from a more fundamental problem: the vendor software packages, as 'intelligent' as each may be, have limited scope (Tolman et al, 2000). Consider a simple question an owner may pose: what will be the cost and schedule implications of a design change (changing exterior finishing technology, for example)? Such multi-disciplinary issues are common, and the limited scope of each application in the above scenario does nothing to aid the project participants resolve them.

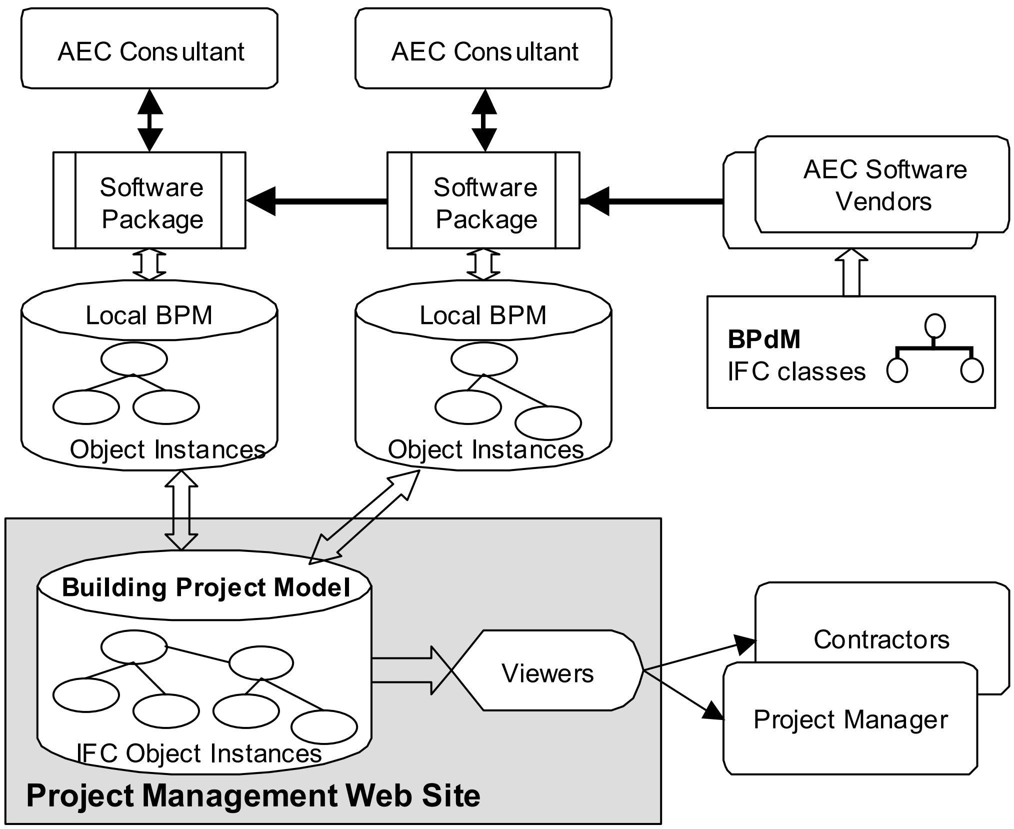

Figure 1: AEC Integration using Industry Foundation Classes and a Project Web Site

Two system architectures, which represent improvements to that described here, have been tested: the WISPER project (Alshawi 2000) and an Internet-based CAD system (Han et al 1999). In both of these, IFCs provide the basis for a common project model, and the Internet is used as the communication medium between users (through CORBA technology). However, only the integrity problem is ameliorated. The others remain, and in addition, the client software packages still require mapping or translation of the project data to their native data formats.

THE PROPOSED PARADIGM

It is proposed that information generation in construction can be re-engineered so as to overcome many of the problems perceived in current research. The proposed paradigm is presented in Figure 2. The key differences here are that

1. Application software is no longer delivered as user based packages, but as object-oriented methods, tightly coupled to the data objects, and delivered over the network, and
2. The Building Project Model is centralized – there are no local distributed models.

The project participants all work through a common interface, which provides the standard project communication functions and access controls. In addition, each user is given appropriate access to the building project model instances through a specific professional 'view' interface (these are programmed by software houses and provided directly by the PMWs). The nature of software applications and the role of the software vendors are changed. Applications are not built and sold as packages, but rather as object method sets encapsulated with the Building Project Data Model. All of the actions a user need perform (e.g. add and edit assemblies, produce drawings, bills of quantity, and other outputs, etc.) are done by launching object methods. It should be borne in mind that the project model includes not only product objects (elements and spaces), but process objects too (activities and resources). Producing a project schedule, for example, could be initiated by a method of the '*IfcProject*' class which would instantiate an '*IfcWorkSchedule'* and its *'IfcTasks'* (IAI 2000). Producing a bill of quantities for a particular building system could be launched by an '*estimate*' method of the *'IfcSystem'* class, producing an '*IfcCostSchedule'*.

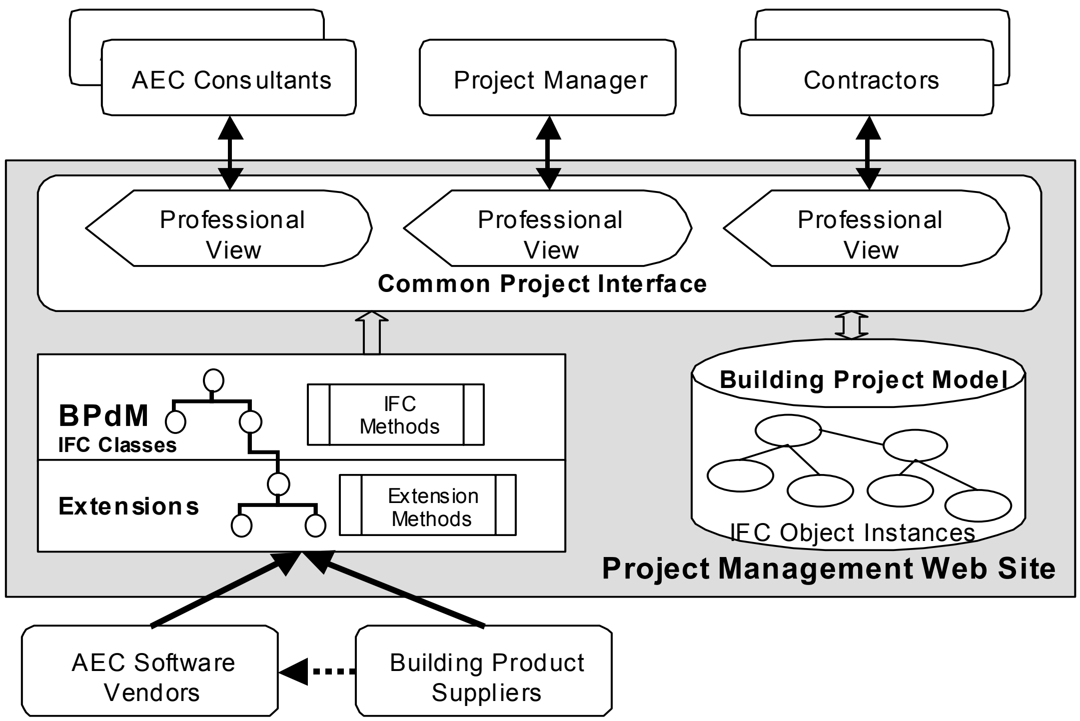

Figure 2. Proposed paradigm for AEC Integration

Two sets of object class definitions and their methods are used, as distinguished by their source: basic IFC classes and methods, and Extension classes and methods. The basic IFC methods provide simple functionality for the root IFC classes, such as *instance()*, *read()*, *write()*, etc. In many cases these will be 'place-holder' (template) methods, whose main purpose is to define the object level interface for the IFC classes. All of the extension classes must inherit from root IFC classes. The extension methods are the majority of the methods required. The extensions can be classified as belonging to one of two kinds:

1. Methods provided by AEC software companies, such as method groups for Preliminary Design, Structural Analysis, Output of Drawings, Bills of Quantities, etc. The vendors' revenues will be accrued by selling the methods 'per use' or by licensing the project management web sites to sell them.

2. Classes and Methods provided by building component suppliers. These will take the form of Intelligent Parametric Templates (IPTs) (Sacks 1998), which will enable a user to design, analyse and detail complete work assemblies directly in the model. It is envisaged that product supply companies will provide these free of charge, with the aim of promoting sales (at present, CAD file libraries are supplied free of charge by a wide range of building component suppliers – e.g. door and window manufacturers (e.g. Andersen 2000), precast concrete components (Spancrete 2000), etc.). Sophisticated software tools, which represent progress in this direction, ", have recently become available (Design Variations 2000).

In figure 2 both the IFC classes and the extension classes are shown as residing on the project management server. This is not the only (and not necessarily the optimal) possibility: in the ASP model, the physical location of software resources is transparent to the user. These may therefore just as well reside on the vendors' servers.

In many cases design offices represent the weak link in the economic chain of construction projects. Their resistance to integration is often associated with the capital investments required for purchasing and upgrading software for use in specific projects. In this paradigm, little or no capital investment is required on their part.

INTELLIGENT PARAMETRIC TEMPLATES

Intelligent Parametric Templates (IPTs) are a technique for populating building project models with sets of object instances (Sacks 1998). An IPT is a defined as "an intelligent mechanism for instantiation of a set of objects and of the relations between them". The 'parametric template' is not a single complex geometric object with many parameters and a predefined fixed form, but rather a set of parametric objects, instantiated dynamically, which can adapt to represent complex real-world assemblies. The term 'intelligent' refers to the collections of object methods and knowledge sets (production rules, neural networks, fuzzy logic, etc) which can be processed in order to instantiate the object instances and determine values for their parameters. As distinct from traditional expert systems, there is no central 'knowledge-base'; instead, packets of 'knowledge' are coupled to the objects representing the building elements and assemblies to which they pertain. Rule sets are inherited in a similar fashion to methods. This top-down approach allows new technical solutions to be added incrementally to the library of IPT's without any change to the system itself or to the previously defined solutions. While the IPT technique has been tested in design tasks, it is suggested that it can be used for scheduling and estimating, preparation of contract documents, and other construction applications.

Three examples are outlined in this section; they demonstrate how intelligent object sets, rooted (but not necessarily contained a priori) in a Building Project data Model (BPdM)[1], may be made available for and used by construction professionals. All were implemented using the AutoLISP++ object-oriented development tool, and based on the Building Project data Model (BPdM) developed for the ABS - Automated Building System (Sacks and Warszawski 1997). The BPdM defines three levels of detail for each of three main aspects of building information:

[1] The term 'Building Project data Model - BPdM' is used to distinguish the object-oriented data model definition (of classes, inheritances and links) from particular instances of building project data, which are termed 'Building Project Models - BPM'.

the building Spaces (Building, Primary Space or Floor, and Secondary Space or Room), the Assemblies (Building Assembly, Work Assembly, Element) and the construction process (Task, Activity and Basic Activity) (Sacks 1998).

Reinforced Concrete Column and Slab Design and Analysis

Two IPTs were implemented for structural design of reinforced concrete columns and slabs in rectangular shaped buildings (Sacks et al 2000) - for flat slabs spanning in two directions and for one directional ribbed (joist) slabs. The requirement was for technical solution templates which could adapt themselves to all of the possible geometric and topological configurations that might be required, such as different slab dimensions, openings in the slab of arbitrary size and location, variations in live load requirements over the floor areas, etc. In the case of the ribbed (joist) slab, the solution made use of the following classes and their methods and rule sets (class names are italicised, and those in bold type were added specifically for this IPT implementation - the others are predefined in the BPdM) (figure 3):

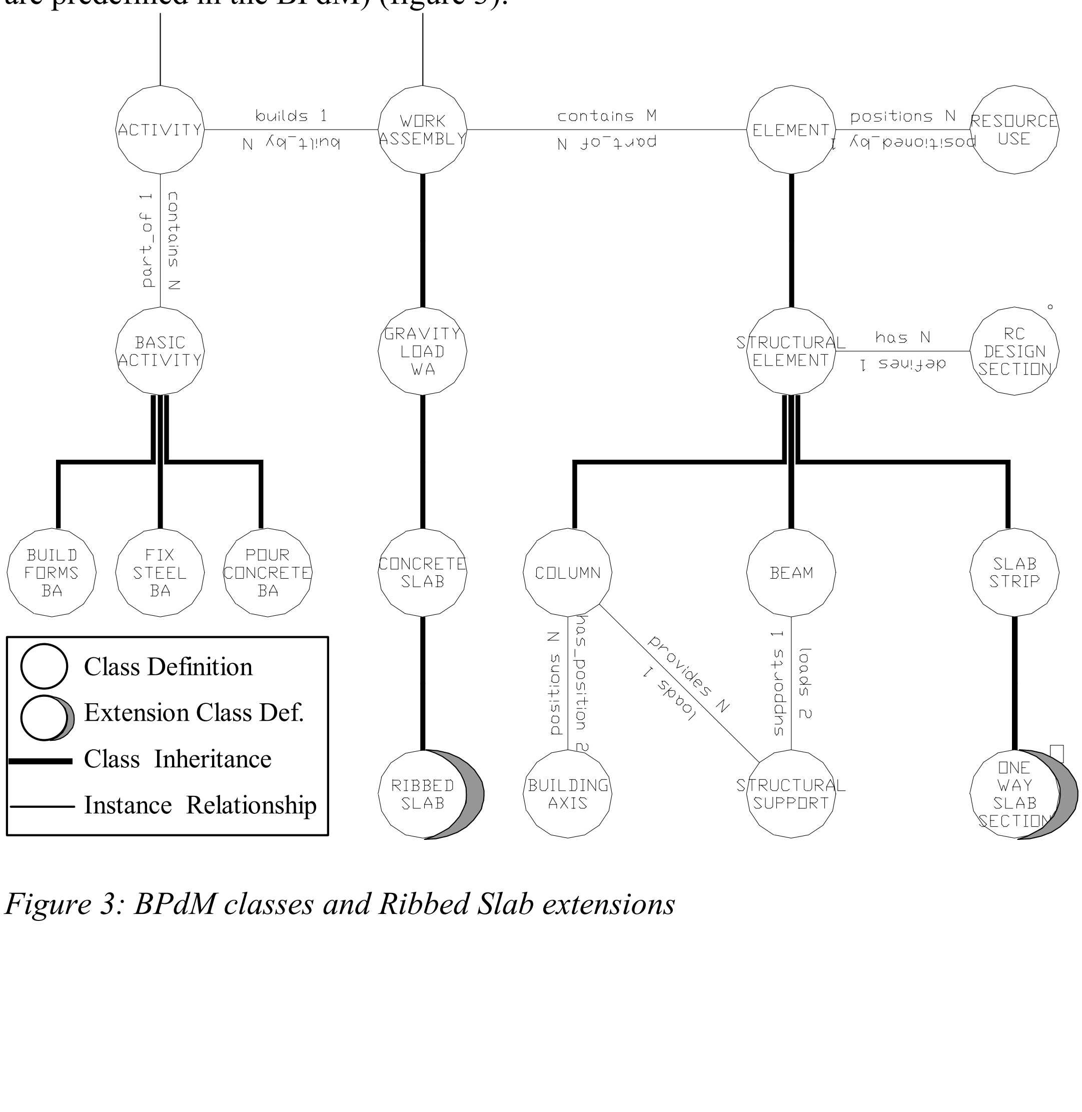

Figure 3: BPdM classes and Ribbed Slab extensions

1. ***Ribbed_Slab***, with rule sets for proposing preliminary feasible span lengths and evaluating the cost of each feasible alternative, an instantiation method for the work assembly, an algorithm and rule set for positioning and instantiating *Columns* in relation to *Building_Axes* and S*tructural_Walls* (building core elements), an algorithm for laying out ***One_Way_Slab_Sections*** of uniform continuous span configurations with respect to openings in the floor, an algorithm and rule set for positioning and instantiating *Beams*, which are supported by columns and core walls (using *Structural_Support* objects), and a method for performing preliminary sizing and structural analysis of the slab system.
2. *Column*, *Beam* and ***One_Way_Slab_Section***, each with methods and rules for fixing dimensions, instantiating *RC_Design_Sections* (reinforced concrete cross sections for storing analysis results and reinforcing details), instantiating basic activities (*Build_Forms, Fix_Steel, Pour_Concrete*), drawing different views, in both 2D and 3D, and measuring and reporting quantities.
3. Ancillary objects, such as *Structural_Support*, *RC_Design_Section*, *Rebar_Use*, *Rebar*, various basic activities, etc., each with the methods required for their operation.

Two features, which comply with the principles inherent in the proposed paradigm (figure 2), should be noted:

a) The two-tier nature of the object definitions - basic objects and interfaces are part of the common BPdM, while others are added to provide the specific functioning of a particular technical solution,

b) The tight coupling of the objects' 'form' (attributes), 'function' (defined by links within the model) and 'behaviour' (methods and rule sets) within the model definition. Specifically, the behaviour is independent of the user interface, or of any one application.

The data resulting from running an IPT to design a slab define the structural solution and are an integral part of the overall project data. Just as the layout method for the structural ***Ribbed_Slab*** object reads the architectural data describing the floor's geometry and openings stored in *Primary_Space* and *Secondary_Space* object instances, the structural *Element* data are available to the architect and other consultants, since the appropriate view methods are defined within the base BPdM.

Automated Rebar Constructability Checking

The second example demonstrates how additional methods and rules can be associated with the object classes in order to perform constructability checking of the reinforcement details (Navon et al 2000). The system consists of two modules - for Diagnosis and Correction. The function of the system is in this case implemented entirely as methods and rule sets of the BPdM classes (figure 4). Different structural elements have different tests associated with them. For example, the Diagnosis module includes specific tests for rebar congestion and collision in beams (*Beam::CheckCongestion*, *Beam::CheckCollision*), while reinforcement ratio and cover are checked for all structural elements. The values of required reinforcement/concrete ratios differ between different element types (beam, column, slab, etc.) - this is implemented simply by associating the rules, with appropriate values, with the specific Structural_Element classes. The system makes use of general drawing and display methods; all that was required to 'mark up' the diagnosis results on the 3D presentation were simple calls to notation methods (figure 5).

Physical interference between elements of different building systems (such as the conflict between drainage pipe DP-1 and beams B2-B5 in figure 5) is also checked. Integration at this level is only possible because the 3D space occupied by an *Element* is exposed by a base class method interface. This is the only way to achieve such integration, because the alternatives - either a) assuming prior knowledge on the part of the programmer of the piping layout system of all the potential building elements with which pipes may conflict, or b) programming a 'checking' expert system, which has a-priori meta-knowledge of all building elements – are not feasible.

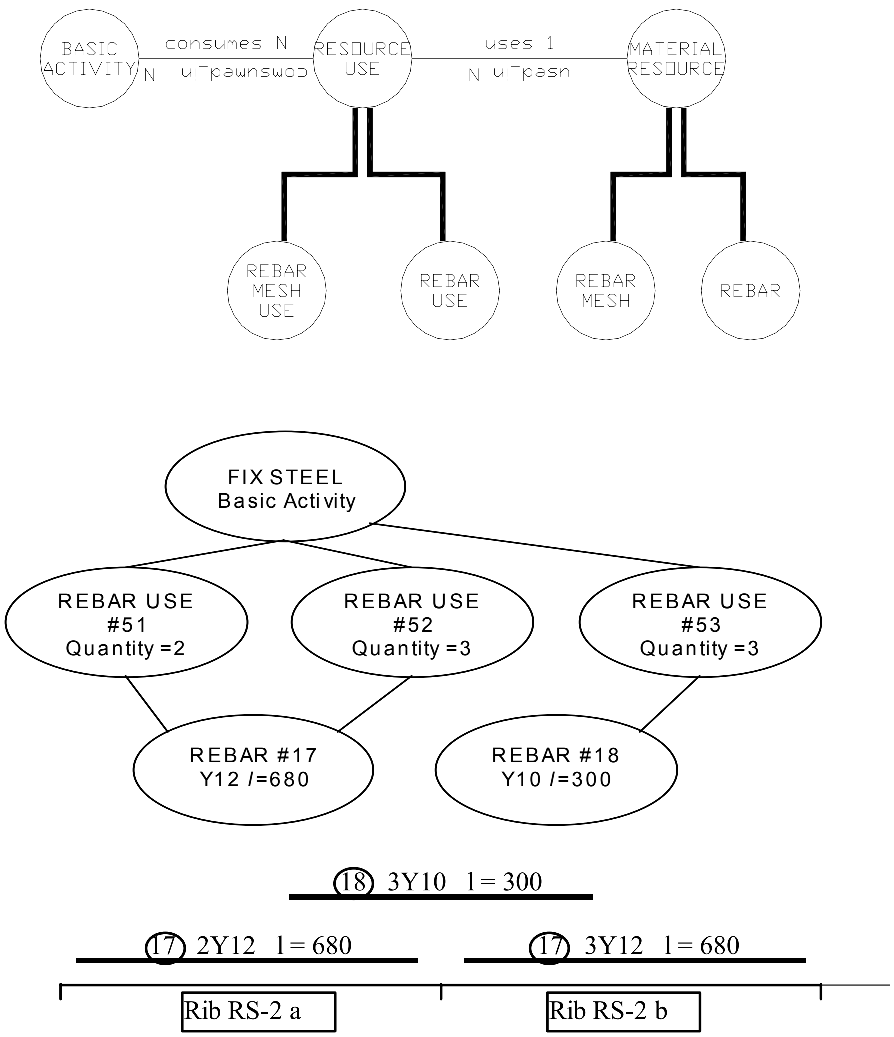

Figure 4: BPDM Rebar Classes, Instances and Detail Drawing

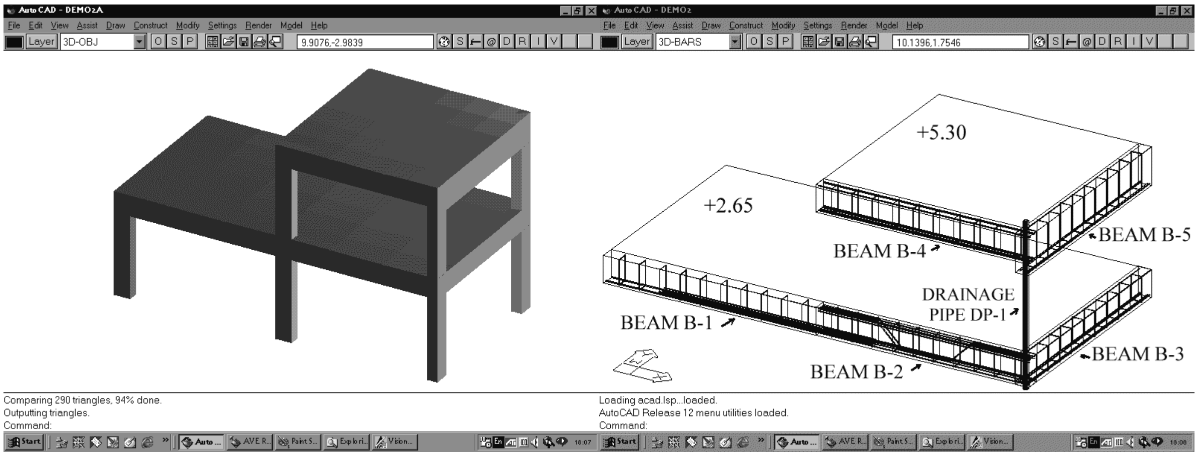

Figure 5: Mark up of Rebar Constructability Problems (Navon et al 2000).

Formwork Design with High Shoring Towers

A prototype design aid for laying out and sizing temporary shoring towers for slab formwork has also been implemented using the IPT concept (Shapira 1998). The knowledge (rules and objects) for laying out the towers, joists and stringers, and checking their spacing for strength, stability and safety requirements, is implemented as an IPT for each proprietary formwork system. Five classes were defined for the IPTs: *Temporary_Support_WA*, which inherits from the *Work_Assembly* class of the BPdM, *Temporary_Support_Element*, which inherits from the BPdM *Element* class, and *Tower*, *Joist* and *Stringer_Array* classes (figure 6). Figure 7 shows a typical tower layout for a flat slab on beam configuration.

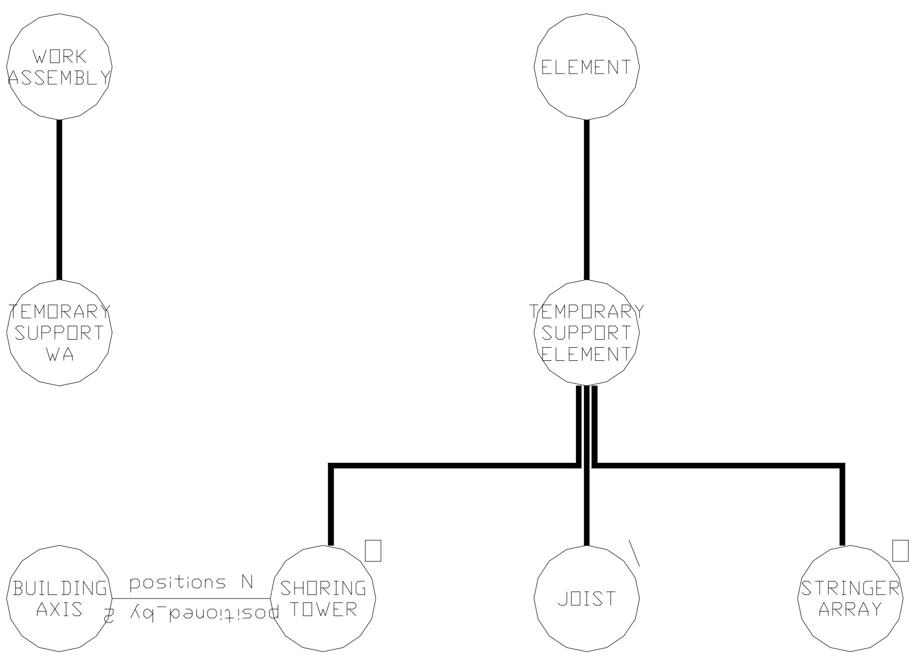

Figure 6: Extension Classes added for the Shoring Tower IPT.

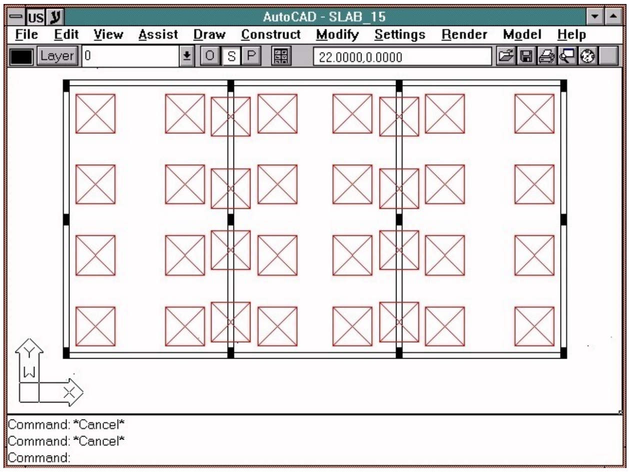

Figure 7: A typical Shoring Tower layout (Shapira 1998).

It is common practice for suppliers of specialized formwork systems such as these to assist the contractor in detailed design of specific shoring tower applications. The prototype system suggests how an IPT could be funded by an equipment supplier, programmed and maintained by a software house, provided as a service by a project management hosting web site, and used by any project team member, as outlined in Figure 2. The formwork IPT objects are fully integrated with the overall project model, and will exhibit all of the behaviours of their parent classes. For example, integration between structural design and construction planning is achieved in that all the structural slab information required by the formwork IPT – the geometry, load/unit area, curing time – can be obtained using the standard interface methods of the generic BPdM *Concrete_Slab* class. Thus a planner could request generation of a number of alternative designs for support for formwork of any particular slab, evaluate them, and make a well-informed decision.

DISCUSSION AND CONCLUSIONS

In order to integrate AEC information across the different design and construction disciplines, not just form, but also function and behaviour of the project elements must be made available. This requires that not only objects be stored in standardized form, but that common class methods must be available too. In addition, no standard can cover all known (and future) building elements. Rather, a common base of classes and methods is required. A paradigm is proposed which offers a solution to these requirements by providing AEC project participants with both a centralized project database and with application methods, delivered over the Internet. The Building Project data Model, and the methods embedded with the data objects, are two-tiered - first, the basic model, and second, the supplier/application level. The second tier is entirely rooted in the first through object-oriented inheritance. The examples demonstrate the extension of the base BPdM by adding new objects and methods (to serve as IPTs or analysis modules), while retaining the functionality of the base model and methods, such as the ability to produce outputs automatically.

The centralized database architecture reduces the problems of maintaining integrity, which are difficult where local databases are used. It allows a standardized basic project model to be combined with growing domain models. Thus it provides versatility in terms of the range of building products it can support. It represents a practical approach to the accumulation of capabilities, through “piecemeal development of modules” (Galle 1995).

While provision of such a service over the Internet has not been tested in this research, other projects have shown it to be technically feasible (Faraj et al 1999). In this way, the financing of development and deployment are transferred from design practices to Project Management Web site ‘dotcoms’ and construction component suppliers, who are better positioned to support capital investments. A window manufacturer, for example, will provide a set of object classes and methods in order to enable an architect to insert/specify such a window in a building design (i.e. in a growing project instance model). Using the IPT technique, the window will be able to “open” the (IFC based) wall object into which it is inserted, detail its’ own parts, and through its methods with IFC defined interfaces, exhibit thermal, lighting and other behaviour. The classes will have to inherit (in OO terms) from the IFC “window” class, so as to ensure that they will exhibit ‘behaviour’ compatible with the kinds of analyses, drawing production, or other information processing that might be required.

Control remains in the hands of the users. The degree of automation is dependent on the methods and modules available – the data structure implicitly supports incorporation of ‘intelligent’ (AI) methods for design or integrity maintenance, as demonstrated by the IPTs.

A number of issues remain unresolved, and require further investigation. The upper (IFC) tier must include not only classes and their attributes and links, but also basic methods (such as *read(), write(), draw(),* etc.) and their interfaces (i.e. method input and output definitions). The precise nature of these has not been defined, nor is it clear who will program them. The same is true of the professional view interfaces, which will need to be provided by the PMWs. The model presents no solution to the problem of maintaining integrity during concurrent editing of the BPM by more than one project participant. The traditional view of application software will no longer hold; it is not yet clear whether software vendors will impede or accelerate the adoption of the ASP strategy. Lastly, changes to the allocation of responsibility and liability for errors resulting from the use of software must be considered.

To date, PMWs provide repositories but not applications. It is likely that market forces will increasingly pressure them to offer enhanced integration services by providing applications as well. Product suppliers can vie for market share by providing ‘better’ methods for use by the specifiers (designers). Adoption of the proposed paradigm could enable them to provide significantly greater integration of the construction information process.

REFERENCES

Alshawi M. (2000) “Integrated Construction Environments: Technology and Implementation”, INCITE 2000: Implementing IT to obtain a competitive advantage in the 21st Century, Hong Kong, HK Poly University press, Jan, 2000, pp 36-53.

Andersen (2000), “Andersen® Window Symbols” http://commercial.andersenwindows.com
ASP Island (2000), http://www.aspisland.com/services/model.asp
Bjork B.C. (1999) “Information Technology in construction: domain definition and research issues”, International Journal of Computer Integrated Design and Construction, Vol. 1 No. 1 pp. 3-16.

Design Variations Inc. (2000), “Varia and the Internet”, White Paper, http://dvinc.com/about.html

Eastman C.M. (1992) “Modeling of buildings: evolution and concepts”, Automation in Construction Elsevier Science B.V. Vol. 1 pp. 99-109.

Eastman C.M. and Kutay A. (1991), “Transaction management in design databases”, in Sriram D., Logcher R. and Fukuda S. (Eds.), Concurrency in Engineering Data Management, Springer New York 1991.

Faraj I., Alshawi M., Aouad G., Child T. and Underwood J. (1999) “Distibuted Object Environment: Using International Standards for data exchange in the Construction Industry”, Computer-Aided Civil and Infrastructure Engineering Vol. 14 pp. 395-405.

Fenves S.J., Flemming U., Hendrickson C., Maher M.L. and Schmitt G. (1990) “Integrated software environment for building design and construction”, Computer-Aided Design Vol. 22 pp. 27-36.

Flemming U. (1994), “Case-based design in the SEED system”, Automation in Construction Elsevier Science B.V. Vol. 3 pp 123-134.

Froese T.M. and Yu K.Q. (1999), “Industry Foundation Class Modeling for Estimating and Scheduling”, Durability of Building Materials and Components 8 (Vol. 4) Lacasse M.A. and Vanier D.J. (Eds.) NRC Research Press, Ottawa, Canada, pp. 2825-2833.

Galle P. (1995), “Towards integrated, “intelligent”, and compliant computer modeling of buildings”, Automation in Construction Elsevier Science B.V. Vol. 4 pp. 189-211.

Gielingh W. (1988) "General AEC Reference Model (GARM)", in Christiansson P., Karlsson H. (editors) "Conceptual Modeling of Buildings", CIB Proceedings, Publication 126 W74+W78 Seminar, Lund, Sweden October 1988.

Han C.S., Kunz J.C. and Law K.H. (1999) “Building Design Services in a Distributed Architecture” ASCE Journal of Computing in Civil Engineering, Vol. 13 No. 1 pp. 12-22.

Hartung C.J., deLone A.C., Kwas R.M. and Austrian B. (2000) “e-Commercial Real Estate Industry Overview”, Banc of America Securities Equity Research Working Paper, March 2000, San Francisco California USA.

IAI (2000), Industry Alliance for Interoperability, IFC 2x, (http://cic.vtt.fi/niai/technical/IFC_2x/index.htm)

Karstila K., Karhu V., Kihlman V., Pellosniemi., Seren K-J.(1994) “Computer Integrated Building Design Based on a Generic Product Data Model”, Proceedings, CIB W78 Workshop on Computer Integrated Construction, Helsinki, Finland, August 1994, International Council for Building Research Studies and Documentation.

Kiviniemi A. (1999) “IAI and IFC - State of the Art”, Durability of Building Materials and Components 8 (Vol. 4) Lacasse M.A. and Vanier D.J. (Eds.) NRC Research Press, Ottawa, Canada pp. 2157-2165.

Navon, R. and Goldschmidt, E. (1999) “Automated Data Acquisition for On-Site Control” Proceedings of the 16th IAARC/IFAC/IEEE International Symposium on Automation and Robotics in Construction (ISARC'99), Madrid (Spain), September 1999, pp. 681-686.

Sacks R. (1998) “Issues in the Development and Implementation of a Building Project Model for an Automated Building System”, International Journal of Construction Information Technology, Salford University, Salford UK Vol 5 No 2 pp 75-101.

Sacks R. and Warszawski A. (1997), “A Project Model for an Automated Building System; Design and Planning Phases”, Automation in Construction, Elsevier Science B.V., December 1997 Vol 7. pp. 21-34.

Sacks R., Warszawski A., Kirsch U. (2000),”Structural Design in an Automated Building System”, Automation in Construction, Elsevier Science B.V., October 2000 Vol 10. No. 1 pp.181-197.

Shapira A. (1998) “Formwork Design for High Shoring Towers”, Concrete International, American Concrete Institute, October 1998.

Spancrete (2000), AutoCAD Based Product Catalog, SPANCRETE Prefabricated Concrete Products, Ashdot Yaakov, Jordan Valley, Israel.

Teicholz P. and Fischer M. (1994), “Strategy for Computer Integrated Construction Technology” ASCE Journal of Construction Engineering and Management, Vol. 120. No. 1, March 1994.

Tolman F.P. (1999) “Product modeling standards for the building and construction industry: past, present and future”, Automation in Construction Vol. 8 pp. 227-235.

Tolman F.P., Ozsariyildiz S. and Schevers H. (2000), “Computer Aided Inception of Large Scale Engineering Projects: Evaluating Value Versus Cost”, INCITE 2000: Implementing IT to obtain a competitive advantage in the 21st Century, Hong Kong, HK Poly University press, Jan, 2000, pp 744-755.

Tolman F.P. and Wix J. (1996) “Building Construction Core Model (BCCM) version 1.0”, ISO/TC/184/SC4/WG1, June 1996.

Turner J. (1988) “AEC Building Systems Model”, working paper ISO/TC/184/SC4/WG1, October 1988.

Liebich T. and Wix J. (1999) “Highlights of the Development Process of Industry Foundation Classes”, Durability of Building Materials and Components 8 (Vol. 4) Lacasse M.A. and Vanier D.J. (Eds.) NRC Research Press, Ottawa, Canada, pp. 2758-2775.

Watson A. and Crowley A., “Deploying the CIMsteel Integration Standards”, Competitive Deployment of Product Data Technology, European Conference Product Data Technology Days 1997 (Sophia Antipolis), QMS, Berkshire UK, 1997 pp. 39-46.

Yau N.-J. (1992) "An Object-Oriented Project Model for Integrating Building Design, Construction Scheduling, and Cost Estimating for Mid-Rise Construction" PhD Thesis, University of Illinois at Urbana-Champaign.