CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

An Object-Oriented Approach for Integrated Project Management Software

by
Thomas Froese and Boyd C. Paulson, Jr.

WORKING PAPER
Number 11
March 1991

Stanford University

CIFE Center for Integrated Facility Engineering • Stanford University

An Object-Oriented Approach for Integrated Project Management Software

If you would like to contact the authors please write to:

c/o CIFE, Civil Engineering,
Stanford University,
Terman Engineering Center
Mail Code: 4020
Stanford, CA 94305-4020

Table of Contents

Page

Note: This paper is based on a research proposal submitted to the National Science Foundation and, as such, describes a proposed program of research for investigating the use of object-oriented principals to support integrated project management software.

Summary

This paper outlines a new approach to developing software for project management that is expected to offer a wide range of improvements to both the software development process and the characteristics of the resulting applications. The overall objectives are to construct systems that offer greater flexibility, higher intelligence, and significantly better integration. The proposed approach does not replace either traditional project management applications or advanced applications such as expert systems; rather, it offers a more efficient and uniform methodology for implementing these programs and a new integrated environment in which these applications can reside.

The basic approach, which relies heavily on the concepts of object-oriented programming, is to create a set of standardized software objects that represent the fundamental elements of the project management process. For example, objects would be created to represent a building's physical components (e.g., beams and columns), the basic construction activities needed to produce these physical components, and the resources required to perform the activities. These basic "building block" objects could then be pieced together, according to general design and development guidelines, to form the core of various project management application programs.

This research will develop the overall design—or object model—for such an approach, will perform detailed design and implementation of the main objects involved, and will use these common objects to construct sample integrated applications.

The advantages of such an approach include the following:

- More efficient software development process because of the advantages of OOP.
- Reduced development effort through the use of a shared common framework.
- Increased flexibility and configurability because of the decoupling of software components, the high modularity of the software, and the overall system's increased functionality.
- Extensive system integration because applications are based on a common view or schema of the domain, and because they physically share common software objects.
- High potential for increased system intelligence since the basic representation is "model-based" and because of the breadth of accessible information in the integrated system.

The resulting object model for project management will not only form the basis of the proposed software development approach, but could also provide a common language or vocabulary for expressing issues related to computer applications for construction (e.g., documenting algorithms, structuring relational databases, or creating inter-application communications), contribute to the eventual establishment of industry data or object model standards, and offer insight into what the fundamental informational elements of the project management process are.

An Object-Oriented Approach for Integrated Project Management Software

A. Introduction and Background

This research proposal deals with the use of computers to support project management. The immediate focus is on the management of the construction phase of engineering projects by a general contracting or construction management company, but it is intended that the concepts could be used for the entire project life cycle—from initial feasibility analysis to facilities management—and by any project participants individually or, ideally, collaboratively.

Computers have long been used to support project management. For certain tasks they are generally indispensable, while in other areas their shortcomings often overshadow their advantages (for example, a recent survey of computer use by small contractors showed that 92% used computers for accounting purposes, while only 25% used either scheduling or estimating programs [Skibniewski 90]). A great deal of research is ongoing into new computer tools for project management. However, new tools that offer greater functionality alone will likely be received with no more enthusiasm than existing systems.

This research is aimed at promoting more fundamental changes to the way that project management computer systems are built and used. The overall objectives are to construct systems that offer greater flexibility, higher intelligence, and significantly better integration. The proposed approach does not replace either traditional project management applications or advanced applications such as expert systems; rather, it offers a more efficient and uniform methodology for implementing these programs and a new integrated environment in which these applications can reside.

The basic approach of the proposed research, which relies heavily on object-oriented programming (OOP) concepts, is to create a set of standardized software objects that represent the fundamental elements of the project management process. For example, objects would be created to represent a building's physical components (e.g., beams and columns), the basic construction activities needed to create these physical components, and the resources required to perform the activities. These basic "building block" objects could then be pieced together, according to general design and development guidelines, to form the core of various project management application programs. This approach is expected to greatly improve the software development process, the characteristics of the resulting software, and–perhaps most significantly–the connectivity or integration of the resulting applications.

This research's main focus is to provide an overall conceptual design for such an approach, to test the approach by performing detailed design and implementation of many of these common objects, and to use the objects to construct sample integrated applications.

Future Directions for Project Management

Project management software has undergone few significant changes over the past 10 to 15 years. Yet we have found that many researchers and industry experts envision some dramatic advances to be made in the near future [Ibbs 85], [Thisner et al 87], [Hansen 87], [Levitt 87], [Logcher 87], [Badger et al 87], [Ibbs et al 87], [Teicholz 87]. We have categorized these advances into three development goals: increasing flexibility, increasing intelligence, and increasing integration. Increasing flexibility involves enlarging the breadth of operations that future systems can perform so that the systems will provide fuller and more generally useful assistance to project management teams. It also refers to increasing the potential for customization or configurablility to suit the unique and dynamic nature of construction projects. Increasing systems' intelligence will allow programs to perform much more interpretation of project data. By imbuing systems with some symbolic representation of what the data mean, the potential is introduced for using knowledge-based techniques to infer portions of the system's data input requirements, to perform simple but voluminous checking of input data and calculated results, and to offer in-depth planning and decision support. Integrating systems will simplify computer use by bringing multiple functionality into a single cohesive system, will support knowledge-based and other "information-hungry" applications by allowing data sharing instead of data entry, will reduce the effort and errors associated with manual information sharing, and may ultimately change the way that project teams work.

Object-Oriented Programming

The preceding section has outlined a set of goals for future project management systems. This proposal suggests that an object-oriented approach will contribute to the attainment of these goals. While there are a variety of concepts and techniques that are generally associated with OOP, the term itself is not well defined. The most general characteristic is the use of objects, which can be thought of as a way of combining software code and data into a single "package." From a software engineering point of view, the underlying idea is that importance lies in *what* operations can be applied to data, not in *how* these operations are performed or how the data is represented. An abstract data type—the object—is therefore defined in which the description of its operations is public, but the representation and implementation details are kept private. This has the effect of localizing much of the programming detail and data control, thereby minimizing programming complexity by reducing the code's "surface area" [Cox 86], [Booch 91]. In contrast, OOP has become popular in artificial intelligence work because each object can be used to represent some object or concept in the real-world. Thus the object-oriented software constitutes a high-level functional model of reality which lends itself well to knowledge-based processing. Finally, objects are being embraced for some database efforts because the database manager software can treat each object in a uniform manner, even though each object may contain very different data types (e.g. character fields, graphics, sounds, etc.). While these are three very different uses of objects, they are all mutually compatible and the proposed research will take advantage of each. The research is also expected to determine the appropriate use of various other OOP characteristics (e.g., encapsulation, polymorphism, inheritance). In summary, the term "OOP" in this proposal refers not to one specific definition or language (such as C++ or Intellicorp's KEE environment), but rather to a general body of programming techniques. Included in the research will be the identification and clarification of those techniques that are most appropriate for the task at hand. Section C demonstrates how OOP will be used to help attain the development goals outlined for project management software.

B. Objectives of The Proposed Research

The broad objectives of this research are to improve software systems' ability to support the practice of project management, particularly through the attainment of increased flexibility, increased intelligence, and increased integration.

The specific objectives of this work are to develop a body of theory that describes a new, object-oriented approach to integrated project management software systems. This new approach is expected to offer improvements in accordance with the broad research objectives.

In addition, it is a research objective to design a set of specific software objects—or an object model—to support the development and integration of a wide range of project management applications. This is intended to help initiate the development of a generally accepted industry standard object model for project management that would enable widespread exchange of software modules and project information.

This proposal will also include the implementation of a prototype system for testing the above ideas.

C. Overview of the Proposed Approach and General Plan of Work

As mentioned above, the overall approach of the proposed research is to define a series of software objects that represent project management's basic physical and conceptual components. These objects would then be used as "building blocks" to construct computer applications for project management. Section E shows that this will improve the software development process, increase the software's intelligence potential, and greatly improve the level of integration among applications.

This section will describe the proposed approach in greater detail, discuss the role of object models and outline a conceptual object model for project management. The specific research methodology is then presented.

The Object Model

The object-oriented software design process involves first identifying what software objects need to exist, then defining what their basic characteristics must be, and finally implementing the objects. Of course, this is an over-simplification of the process. The identification of objects must be based on a study of the users' requirements; objects are usually designed to represent the actual objects (physical or conceptual components) of the problem domain. Implementation must be followed by validation and testing. The object-oriented design process also tends to be iterative, with rapid functional prototypes gradually evolving into final applications. However, the idea of design as a three-step process is useful since these steps can generally be sequentially independent. That is, the major objects can be mapped out with very little concern about the specific details of each object or the implementation techniques. Next, each object's detailed definition can be designed independently of implementation issues. Finally, these definitions can be used to implement each object independently of any others.

The identification of the objects required in the system and the definition of the requirements of each object constitute a design for the system (independent of most specific implementation details but complete in terms of detailed functional specifications). This form of system design plays a central role in the proposed research and is referred to as the "object model". The object model acts as the documentation of the system design from which programming efforts can proceed, much in the way that plans and specifications for a building form the design documentation from which construction can be performed. The object model also acts as a system road map to guide future programmers in maintaining and upgrading the software.

For integrated systems, each individual application (or module within the integrated system) would have its own object model. However, for integration to occur the system must be able to translate each model to every other, must be able to translate each model to some global framework, or must initially base each application's model on a single global model. This proposal strongly recommends the latter approach as the most effective for integrated systems, with the second approach used to interface with applications that are developed independently of the overall integrated system. In either case, the role of a generally applicable global object model for all project management applications becomes of paramount importance for providing a central unifying framework around which the software can be constructed. The design of such a global object model for construction is one of the research's main objectives. A preliminary conceptual model has already been developed and is described in the following section.

An Object Model for Construction

Figure 1 gives an overview of a preliminary conceptual model for general project management applications. The three main elements of this model are the libraries of general construction objects, libraries of specific project objects, and application objects.

First, common construction objects reside in the **general construction libraries**. These objects typically represent collections or categories of things. For example, they might include libraries of resource types (including labor, equipment and materials), component types (the facility's physical elements), action types (e.g., Install, Form, etc.), and method types (that identify the methods and resources needed to perform different types of actions on different component types). A construction activity type library is derived from the component, action, and method types. An activity type identifies a specific method type used to perform a specific action type on a specific component type. The typical level of granularity is approximately that at which companies might keep productivity and unit cost data. These libraries are analogous to the type of information found in an estimating data handbook or in master job breakdown structures that many companies use.

Second, the basic elements of each individual project are identified in **specific project libraries**. These objects are typically individual members of the collections represented by the common construction objects. These include the project's specific physical components, the resources used, and the specific activities involved. An activity is a specific activity type applied to a specific component or set of similar components. *Thus activities represent a basic unit of construction action or effort*. These objects would typically be organized into various breakdown hierarchies, the depth of which is based on the detail available from or required by applications. Therefore, while many of these objects would be of very low-level granularity and would be very numerous on typical projects, they would represent little more information than is necessary to perform typical project management functions (e.g. Primavera requires definitions of activities, the

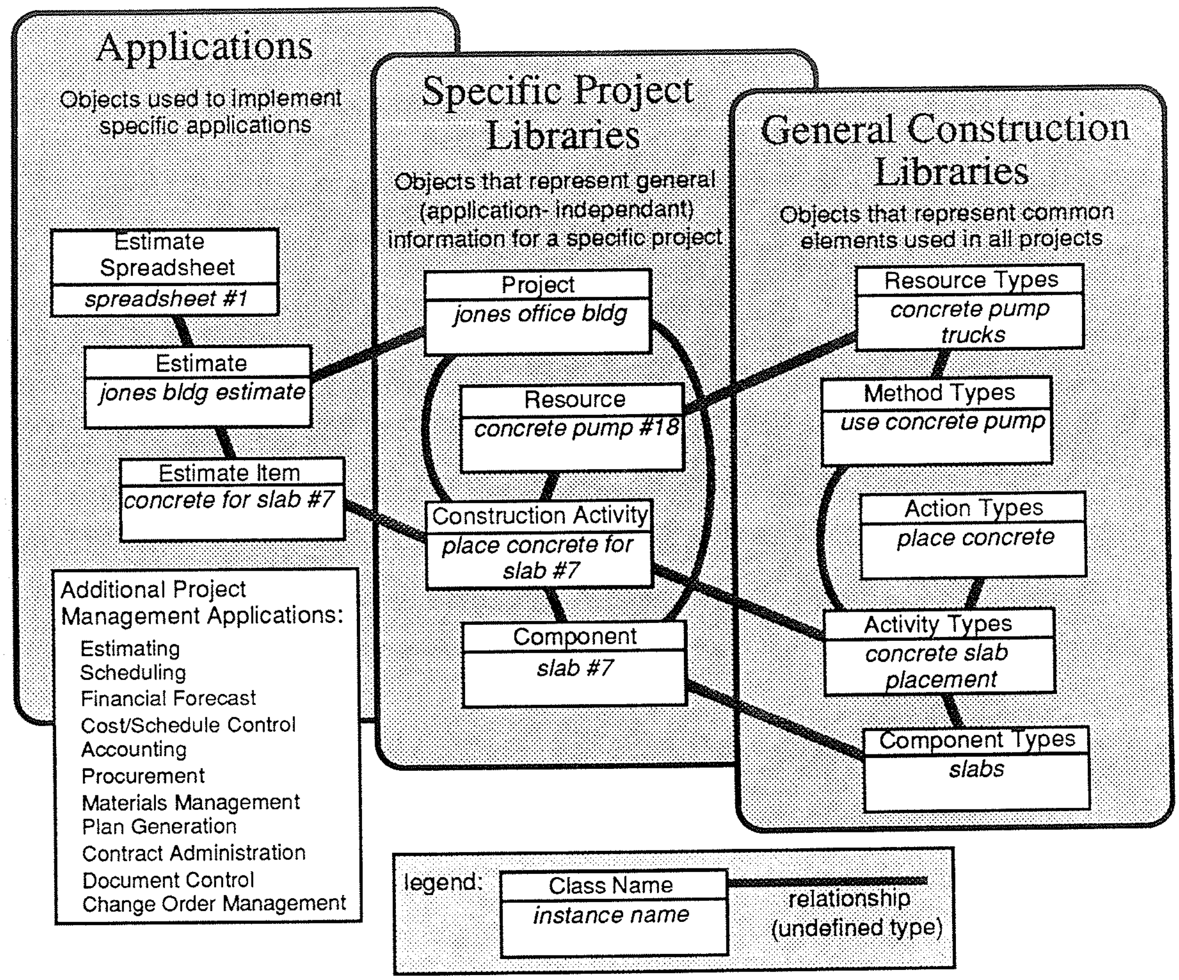

Figure 1: Overview of a conceptual object model for general project management applications. For each of the three main object groups, the figure shows examples of typical classes (definitions or templates for collections of like objects), corresponding typical objects, and typical relationships between objects (these are shown as generic undefined relationships at this level of detail).

associated resources, and cost codes in order to perform basic CPM scheduling with resource leveling and time-cost progress reporting). Furthermore, these low-level objects would usually be completely hidden to the users who interact with the applications that generate and utilize these primitive objects rather than interacting with these objects directly.

Third, specific **project management applications** involve objects that generate, refine, or utilize the basic primitive project elements. For example, an estimating system might be used to generate the project elements. For each item in the estimate, the estimator must select an activity type in order to get unit prices and must identify specific components (by component type, location, and quantity or specific enumeration) so that it will be clear what is being measured. The estimator must then measure these components' dimensions and calculate the costs. In the proposed estimating system, the estimator would select the activity types and component types from the standard libraries. This would lead to the automatic generation of components, activities, and resources as well as the automatic calculation of the items' costs. The links between the estimate items and the activities will allow mapping to budgets and schedules. Alternatively, the activities could be generated by the scheduling process or directly from a CAD model of the project.

It can be seen that this model incorporates both the standard data types used by traditional project management software (e.g., estimate line items or CPM scheduling activities) and the representation of basic construction primitives (e.g., resources or construction activities[1] that are needed for applications such as knowledge-based reasoning. The basic construction primitives also provide the common ground through which each application's data can be tied together. This model could be used for programming a single application (e.g., for a scheduling system, construction activities would "collapse" to scheduling activities and the system would need not be much different than traditional stand-alone scheduling programs), but it could also be scaled up to support a very wide range of different applications (in which case each application would manage its own links to a central body of information that existed independently of any one application.

The above description of an object model for project management is intended to provide a general overview only. The objects mentioned are approximately those that will be required, but no detailed definitions have been given. The actual model resulting from the proposed research will consist of very detailed object definitions that include full descriptions of each object's purpose, descriptions of all the object's attributes and relationships, and full documentation of the methods that the object must have. The model will also include documentation of conventions for using these objects to build applications. It is expected that the general model will include definitions for several dozen main object types, with heavy subclassing of certain objects (for example, the object representing "resources" will be subclassed to represent labor resources, materials, equipment, hoisting equipment, tower cranes, etc.) resulting in several hundred objects total.

Research Methodology

As shown above, the general approach and conceptual layout for the proposed research is already underway. This work has been performed with funding from a CIFE seed research grant. While the work to date has confirmed our expectations about the validity of the proposed approach, the CIFE grants are available for seed purposes only and it is time to expand this project. The following research activities have been pursued to date:

1. **Investigation of goals for future project management software:** In order to help focus the research efforts, a study of current project management software's shortcomings and of the expected and desirable directions of future advances has been made. As stated in Section A, these development goals can be summarized as increased flexibility, increased integration, and increased intelligence. This study's results are discussed in more detail in [Froese & Paulson 90] and [Froese & Waugh 91].

2. **Review of object-oriented programming:** Object-oriented programming is a large, ill-defined, and rapidly advancing field of study. The proposed research is not intended to make significant contributions to the object-oriented programming field in general, but rather to use this technology to make contributions to the field of project management

[1] Here, "construction activities" is a more general term used to represent any form of construction effort while "scheduling activities" typically implies a temporal view of a work package of a certain level of granularity or scope.

software. To this end, the current state of object-oriented programming has been studied and new developments in the this field will continue to be monitored.

3. **Basic conceptual design of overall approach:** The basic approach to improving project management systems using object-oriented technology has been mapped out and an initial conceptual object model has been designed. These have been outlined above.

The following activities are proposed for the continuing research:

4. **Design of a general object model:** The design of a general object model for project management will be completed. This will include identifying what objects will be required, how these objects will interact with each other, and how the objects are to be used to construct applications and to communicate between different application modules in integrated systems. Details such as object messaging standards, storage of objects, etc. will be worked out and documented.

5. **Detailed design of core objects:** Once the object model has been thoroughly mapped out, the detailed design of each object in the model must be performed. This stage will involve designing functional specifications for the attributes, relationships, and operations that are required for the objects.

6. **Creation of sample applications:** The object model design will include conventions and guidelines on how to use the general objects to construct project management applications. Several such applications will be produced in order to thoroughly test the proposed approach. It is expected that as a minimum, application modules will be constructed to perform general browsing and editing of the object libraries, project estimating, project scheduling, and some simple expert systems functions. The intended platform for this implementation is the NeXT computer. Figure 2 shows sample screens from an object-oriented CPM scheduling system that has been previously implemented and can be slightly modified to act as a scheduling module for the integrated systems. An object library browser module is also under development for creating, viewing, and editing the various object libraries. In this stage, the prototype system will also be linked to the existing KADBASE and OARPLAN systems developed by other CIFE researchers (see Section D).

7. **Documentation:** Because the proposed object model is intended to be used as a framework by various software developers who were not involved in its development, the approach cannot succeed unless the results are very well documented. An emphasis will therefore be placed on fully documenting all objects *as they are being designed and implemented.* Object documentation should convey the overall organization of the objects in the model, the specific details of each object, and the conventions and guidelines for using the objects. For this purpose, formal design representation and documentation techniques will be used that will include both graphical and textual components [Booch 91] and, if possible, will be delivered in an on-line, hyper-text type of environment.

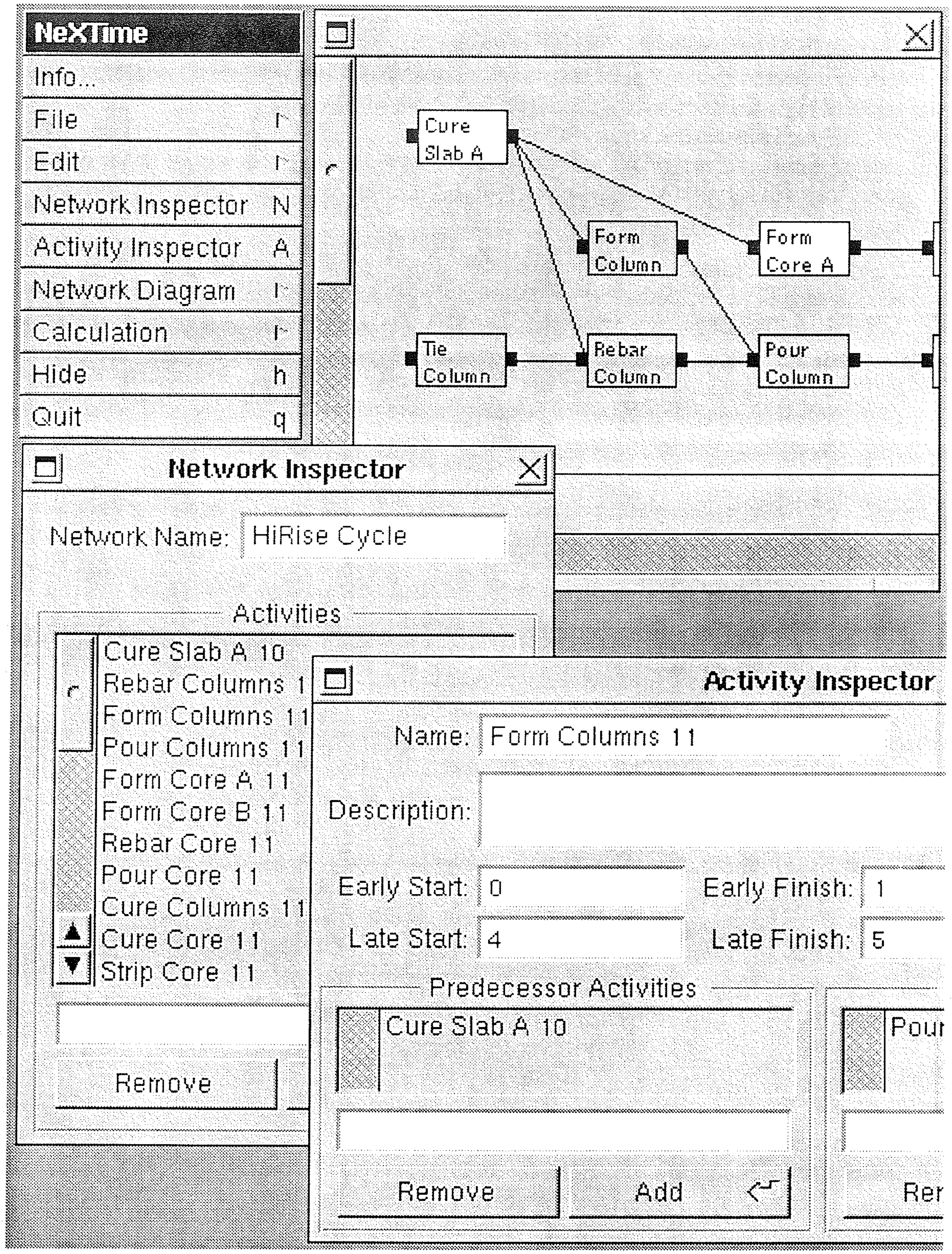

Figure 2: Sample screens from an object-oriented CPM scheduling program developed by T. Froese on the NeXT computer which will be incorporated into the proposed integrated systems.

8. **Investigate future potential:** The long-term vision behind this research, provided that the proposed approach can actually offer the expected advantages, is to move towards an industry standard set of general project management objects. Any programmers could make new or revised objects available for others to purchase; any application developers could use these generic objects to construct new systems; and all applications would be able to interact through the standard object breakdown. This vision is far beyond the proposed research's scope, yet the possibility of and the best path towards such capabilities will be investigated.

9. **Publications based on this research:** i) Papers. We will help disseminate the results of this research through appropriately placed articles and conference papers. ii) Interim and Final Reports. We plan to prepare reports on this project for distribution to interested researchers and practitioners at the end of the first and second years. Furthermore, the results will be disseminated, and the reports distributed, through the Center for Integrated Facilities Engineering.

D. Relation to Present State of Knowledge in Related Fields and Interaction with Other Researchers

The proposed research builds upon several areas of study. The work relies heavily on the underlying concepts of OOP (as outlined in many references such as [Cox 86] and [Booch 91]) but is not tied to any specific research thrusts in this broad and quickly advancing field. This project is more closely tied to data modelling work, particularly in related domains such as building design. Several data modelling efforts ([Abdalla 89], [Lavakare and Howard 89], [Baugh & Rehak 89]) have been studied during this proposal's development and progress in these areas will continue to be monitored. A wide variety of object-oriented and knowledge-based project management software projects have much to offer for suggesting representation approaches for project information. These include [Darwiche 88], [Levitt & Kunz 87], [Ito et al 89], [Axworthy 89], [Kartam 89], [Tommelein 89], [Waugh 90] at Stanford and [Logcher 87], [Cherneff et al 89], [Abu-Hijleh and Ibbs 90], [Best & Inkpen 90], and [Kellett et al 90] elsewhere. The main increments that the proposed work offers over these systems are the overall generality of the basic representation and the resulting potential for integration. Finally, some initial work has been performed previously on object-oriented integrated project management systems [Grobler 88], [Grobler & Kim 89]. The proposed approach is similar, but will attempt to go into much greater depth in designing specific components of a common object representation for project management.

This research has close ties to several projects ongoing within Stanford's Center for Integrated Facilities Engineering (CIFE). The proposed research will draw upon data modelling theory developed by Prof. Craig Howard and other CIFE researchers [Howard et al 90] and, in return, will attempt to contribute a construction perspective to their larger project data models. The proposed research will also benefit from numerous projects studying knowledge-based construction systems [Darwiche 88], [Ito et al 89], [Axworthy 89], [Tommelein 89], [Waugh 90]. These systems will be studied to find common themes in the models they use to represent construction environments. Specific plans include linking the proposed integrated project management system with the OARPLAN knowledge-based schedule generation system [Darwiche 88] through the KADBASE intelligent data exchange system [Howard & Rehak 89]. More generally, the proposed research directly supports CIFE's mandate of exploring technologies that will help increase integration in the fragmented engineering and construction industries.

The proposed research has evolved, in part, from recent work by the principal investigator, along with L. Chua and others, in areas of general intelligence for future construction robots, specifically in representation issues for construction knowledge [Paulson et al 89], [Paulson, Froese and Chua 89] [Chua 90]. Since it will lead to a general symbolic model of the construction domain, the proposed research will be useful for ongoing efforts in representations for intelligent automation. Also, concurrent research by the principal investigator and H. El-Bibany is underway in the area of managing the constraints that would be imposed by various parties on a shared body of project-specific information (e.g., a shared project database) [El-Bibany et al 90]. This constraint management project could provide an important piece of the underlying technology that would eventually be required for supporting large integrated object-oriented systems.

In addition, several meetings have been held with researchers at the University of California, Berkeley, in hopes of establishing a mechanism through which any researchers in the area of OOP for project management and construction applications can exchange both general ideas and specific software objects. Finally, the potential exists to offer the resulting object model as the basis for an industry information exchange standard through the PDES/STEP project. This possibility will be evaluated after the model has been developed and tested.

E. Significance of the Research

Advantages of the Proposed Approach

The proposed approach of creating a series of general project management objects to use as a basis for constructing application software offers the following advantages:

1. Simply by using OOP, the software development process is expected to become more efficient. OOP offers several advantages such as more rapid and easily extendible prototyping, shorter development times, greater system modularity, greater code reusability, and easier system upgrading [Cox 86], [Booch 91].

2. In addition to the software development advantages offered by OOP in general, the use of a general object model will minimize future software development effort since the general framework and much of the common code will already exist.

3. The flexibility of the project management software is expected to be increased in several ways. The high degree of encapsulation (or hiding of the representation and implementation details) means that each portion of the system is less dependent on assumptions about how other parts work. This decoupling of components makes it easier to provide the user with more control over each portion of the system. Similarly, the high modularity will improve the software's customization capability since specific objects can be replaced to change certain characteristics with no effect to the rest of the system. Finally, the increased functionality (see 4 and 5 below) makes it more likely that the system will be able to support users in the ways that they wish to work, rather than forcing them to follow some specific routine.

4. System integration will be greatly increased because the various applications will all be constructed around a common view or schema of the domain, as well as physically sharing common software objects.

5. The potential for adding system intelligence is improved since the basic representation is made up of a symbolic model of the real world domain (i.e., a model-based representation) and because of the breath of data available to information-hungry knowledge-based applications through the integration mentioned in item 4. Furthermore, knowledge-based systems' accessibility and acceptance can be improved by merging them with more traditional applications in a single integrated environment.

It is expected that the advantages outlined in item 1 alone will lead to extensive use of OOP for software development efforts. However, items 2 through 5 represent much more fundamental changes to the way that computers are used to support project management. These changes will not come about by the use of OOP alone, but only if a significant effort is devoted to designing a common framework that can be shared by all applications. This is the purpose of this proposed research.

Deliverables and Contributions

The deliverables and corresponding contributions of this research will be as follows:

1. *A body of theory describing an overall approach to using OOP for integrated project management software:* As described above, the proposed basic approach is expected to offer many advantages for the development and the performance of project management software.

2. *A general object model for project management:* This will include an overall layout of the objects involved, detailed object designs for all major objects, and descriptions of usage guidelines and conventions. This model, delivered in the form of full graphical and written documentation, possibly as an on-line hyper-text reference, and as an implemented set of software objects (see number 3 below) will have many uses. It will provide a formal language or vocabulary for expressing issues related to computer applications for construction (e.g., documenting algorithms, structuring relational databases, or creating inter-application communications). It can be used directly as the a central software core around which new object-oriented applications can be constructed and through which various applications can be integrated. By proving such a model's utility, it may contribute to the eventual establishment of industry data or object model standards. Finally, at a level distinct from computer applications, it will offer insight into what the fundamental informational elements of the project management process are.

3. *Implemented general object libraries and specific applications:* The proposed object model's implementation will provide a platform for extensive experimentation of the approach. The system is also expected to act as a software foundation upon which future project management research systems can be constructed.

Note that in the proposed objected-oriented approach, there is less separation of data representation and data processing that there would be, for example, in a large relational database for project management information. The proposed data model is both a representational framework and a central architecture for the object-oriented programs that utilize the model. Yet the focus of this work—creating the overall layout and the application-independent common objects—deals much more with representation than processes. Thus the resulting system will not have extensive functionality beyond the sample applications, though it could be used to construct or to integrate virtually any conceivable project management application.

The short term significance of this research will be to demonstrate the effectiveness of the proposed object-oriented approach for creating applications in general and for enabling inter-application integration in particular. Once this has been established, the resulting object model and implemented object libraries will provide a framework for the design and construction of a wide range of new applications. In the long term, the proposed research could contribute to the eventual establishment of industry-wide standards for the way that objects are used for project management software (possibly through the PDES/STEP or similar efforts [Warthen 88]), universal data exchange capabilities, fully modular software components, and successful integration of AI programming into common usage. While the full achievement of these goals lies far in the future, it is hoped that the proposed research could make a preliminary contribution in this direction. The overall impact would be significantly better project information, communication, and decision support systems; thus reducing errors and uncertainty while improving efficiencies in the project management and construction industries.

Bibliography

Abdalla, Gamaleldin A., *Object-Oriented Principles and Techniques for Computer Integrated Design,* Ph.D. Thesis, Dept. of Civil Engineering, University of California at Berkeley, 1989.

Abu-Hijleh, Samer, F. and Ibbs, C.W., "An Intelligent Exception Reporting System," presented at the *11th International Cost Engineering Congress and 6th AFITEP Annual Meeting,* Paris, France, April 1990.

Axworthy, Alan T., *A Knowledge-Based Model of Time-Cost Tradeoff Analysis for Construction Schedules,* Ph.D. Thesis Department of Civil Engineering, Stanford University, November 1989.

Badger, A.A., Reinschmidt, K.A. and Gandt, A.R., "Project Control System Integration," *Project Controls: Needs and Solutions,* Ibbs, C. William, Jr. and Ashley, David B. Eds. New York: American Society for Civil Engineers, 1987, pp. 88-100.

Baugh, John W. Jr. and Rehak, Daniel R., *Computational Abstractions for Finite Element Programming,* Technical Report R-89-182, Dept. of Civil Engineering, Carnegie Mellon University, Sept. 1989.

Best, J.T. and Inkpen, C.J., "Resource and Activity Simulation in a Knowledge-Based Planning System," *First International Conference on Expert Planning Systems*, Institution of Electrical Engineers Conference Publication No. 322, Brighton, UK, June 1990, pp. 119-123

Booch, Grady, *Object Oriented Design with Applications*, Redwood City CA: Benjamin Cummings, 1991.

Cherneff, J., Logcher, R., and Sriram, D., "Automating Schedule Generation From Architectural Drawings," Proc. of ASCE Construction Congress, San Francisco, 1989.

Chua, Lai Heng, *A Theory of the Knowledge Environment for Construction Automation,* Ph.D. Thesis Department of Civil Engineering, Stanford University, January 1990.

Cox, Brad J., *Object-Oriented Programming, An Evolutionary Approach,* Reading Massachusetts: Addison-Wesley, 1986.

Darwiche, A., Levitt, R.E., and Hayes-Roth, B., "OARPLAN: Generating Project Plans in a Blackboard System by Reasoning about Objects, Actions and Resources," *Artificial Intelligence in Engineering Design, Analysis and Manufacturing*, Vol. 2, No. 3, 1988.

El-Bibany, Hossam, Paulson, B.C. and Chua, Lai Heng, "Coordination between Project Participants through Constraint Management," *Proceedings of the 7th International Symposium on Automation and Robotics in Construction,* Bristol, England, June 5-7, 1990. pp. 505-513.

Froese, Thomas M. and Paulson, B.C., "Object-Oriented Programming for Project Management Software," *Proceedings of the 7th International Symposium on Automation and Robotics in Construction,* Bristol, England, June 5-7, 1990. pp. 513-521.

Froese, Thomas M., and Waugh, Lloyd M., "Future Trends in Computers for Project Management," Accepted for *The Annual Conference of the Canadian Society for Civil Engineering,* Vancouver, Canada, May, 1991

Grobler, Francois and Kim, Simon, "Symbolic Unified Project Representation (SUPR) Model: An Environment for Automated Construction," *Proceedings of the 6th International Symposium on Automation and Robotics in Construction,* Burlingame, California, June 6-8, 1989, pp. 545-552.

Grobler, Francois, Object-Oriented Data Representation for Unified Construction Project Information, Ph.D. Thesis, Department of Civil Engineering, University of Illinois at Urbana-Champaign, 1988.

Hansen, Soren, "Hardware & Software Implications on: PM and the Computer: The Year 2001," *Project Management Journal*, Vol.18, No. 3, August 1987. pp. 47-48.

Howard, H.Craig and Rehak, H.C., "KADBASE: A Prototype Expert Systems-Database Interface for Engineering Systems," IEEE Expert, Vol. 4, No. 3, pp 65-76, Fall 1989.

Howard, H.Craig, Abdalla, Jamal A., and Phan, D. H. Douglas, "A Primitive-Composite Approach for Structural Data Modelling," *Journal of Computing in Civil Engineering*, American Society of Civil Engineers, Vol. 4, No. 4, September 1990.

Ibbs, C. William, Jr., Ashley, David B, Neil, James M. and Feiler, Frank W., "An Implementation Strategy for Improving Project Control Systems" *Project Controls: Needs and Solutions,* Ibbs, C. William, Jr. and Ashley, David B. Eds. New York: American Society for Civil Engineers, 1987, pp. 101-112.

Ibbs, C. William, Jr., *Proceedings of a Workshop for the Development of New Research Directions in Computerized Applications to Construction Engineering and Management Studies,* Urbana: University of Illinois at Urbana-Champaign, Department of Civil Engineering, 1985.

Ito, K., Ueno, Y., Levitt, R.E., and Darwiche, A., *Linking Knowledge-Based Systems to CAD Design Data with Object-Oriented Building Product Model*, CIFE technical report No. 17, Center for Integrated Facility Engineering, Stanford University, Aug. 1989.

Kartam, N.A., Investigating the Utility of Artificial Intelligence T*echniques for Automatic Generation of Construction Project Plans*, Ph.D. Thesis Department of Civil Engineering, Stanford University, 1989.

Kellett, J.M., Brown, N. and Boardman, J.T., "A Structure for an Interactive Project Management System," *First International Conference on Expert Planning Systems*, Institution of Electrical Engineers Conference Publication No. 322, Brighton, UK, June 1990, pp. 163-168

Lavakare, A. and Howard, H.C., *Structural Steel Framing Data Model*, CIFE technical report No. 12, Center for Integrated Facility Engineering, Stanford University, June 1989.

Levitt, Raymond E. and Kunz, John C., "Using Artificial Intelligence Techniques to Support Project Management," *AI EDAM*, Vol.1, No. 1, 1987. pp. 3-24.

Levitt, Raymond E., "Expert Systems in Construction: State of the Art," in *Expert systems for Civil Engineers: Technology and Application,* Maher, Mary L. ed., New York: The American Society for Civil Engineers, 1987, pp. 85-112.

Logcher, Robert D., “Adding Knowledge Based Systems Technology to Project Control Systems,” *Project Controls: Needs and Solutions,* Ibbs, C. William, Jr. and Ashley, David B. Eds. New York: American Society for Civil Engineers, 1987, pp. 76-87.

Logcher, Robert D., “Adding Knowledge Based Systems Technology to Project Control Systems,” *Project Controls: Needs and Solutions,* Ibbs, C. William, Jr. and Ashley, David B. Eds. New York: American Society for Civil Engineers, 1987, pp. 76-87.

Paulson, Boyd C., Jr., Babar, N., Chua, L.H., and Froese, T.M., “Simulating Construction Robot Agents and their Knowledge Environment,” *Journal of Computing in Civil Engineering,* ASCE, Vol. 3, No. 4, October, 1989, pp. 303-319.

Paulson, Boyd C., Jr., Froese, T.M. and Chua, L.H., “Simulating the Knowledge Environment for Autonomous Construction Robot Agents,” *Proceedings of the 6th International Symposium on Automation and Robotics in Construction,* Burlingame, California, June 6-8, 1989, pp. 475-482.

Skibniewski, Miroslaw J., “On the Use of Microcomputers by Small Contractors: Implications of a survey and Recommendations for the Future,” *Project Management Journal*, Vol.21, No. 1, March 1990. pp. 25-31.

Teicholz, Paul M., “Current Needs for Cost Control Systems,” *Project Controls: Needs and Solutions,* Ibbs, C. William, Jr. and Ashley, David B. Eds. New York: American Society for Civil Engineers, 1987, pp. 47-57.

Thisner, Anders, Teicholz, Paul M. and Havas, George, "PM and the Computer: The Year 2001," *Project Management Journal*, Vol.18, No. 3, August 1987. pp. 39-45.

Tommelein, Iris D., *SightPlan: An Expert System that Models and Augments Human Decision-Making for Designing Construction Site Layouts,* Ph.D. Thesis Department of Civil Engineering, Stanford University, August 1989.

Warthen, Barbara, “PDES—A CAD Standard for Data Exchange,” UNIX World, Dec. 1988, pp.103-104.

Waugh, L. M. and Froese, T. M., “Constraint Knowledge for Construction Scheduling,” *Proceedings of the First International Conference on Expert Planning Systems*, IEE Conference Publication Number 322, Brighton, England, June 27-29, 1990. pp. 114-118.

Waugh, Lloyd M., *A Construction Planner*, Ph.D. Thesis Department of Civil Engineering, Stanford University, June 1990.