REPRESENTING PROJECT INFORMATION AND CONSTRUCTION METHOD KNOWLEDGE FOR COMPUTER-AIDED CONSTRUCTION MANAGEMENT

Martin A. Fischer[1], Gijsbertus T. (Bart) Luiten[2], Florian Aalami[3]

ABSTRACT: Currently available construction management software has serious limitations that hinder further integration and automation of construction management tasks. The main limitation is the low level at which project information and knowledge is represented. This means that integration of design and construction planning information and automated reasoning about, for example, planning, are difficult to implement. In two research projects at the Center for Integrated Facility Engineering at Stanford University, we are addressing these limitations. In the SPACECAKE project we propose a higher level representation of project information that explicitly represents the relations between products, activities, construction methods and resources. A prototype system shows that it is possible to implement our conceptual model and support project management decisions. In the MOCA project we elaborate further on the representation of construction method knowledge. In this paper we propose a template to make the knowledge explicit and computer-interpretable.

INTRODUCTION

Planning, scheduling, and cost estimating—in short construction process design—are important construction management tasks: construction managers are constantly asked "how much will it cost" and "how long will it take." Today, construction process design requires significant time and resources. Errors in schedules and estimates are not uncommon, and accurate and detailed feedback on cost and schedule implications of design decisions is often not available until late in the project delivery process. Concurrent engineering and design-build approaches are trying to overcome such problems largely with organizational means. Moving construction management tasks into early phases of project development, however, only increases the pressures on construction managers to develop accurate schedule and budget feedback in a timely manner.

Scheduling and cost estimating have been supported, for some time now, by project management software. Available software has, however, four significant shortcomings that hinder, in our opinion, the further integration of planning, scheduling, and cost estimating—or process design—with product (facility) and organization design. Such technical integration is needed to support organizational means for integration. These shortcomings also stand in the way of increased automation of planning, scheduling, and estimating. These shortcomings are:

---

[1] Assistant Professor, Dept. of Civil Engineering, Stanford, CA 94305-4020, fischer@ce.stanford.edu

[2] Postdoctoral Fellow, Dept. of Civil Engineering, Stanford, CA 94305-4020, and Dept. of Civil Engineering, Delft University of Technology, The Netherlands; now with HBG, The Netherlands, gluiten@hbg.nl

[3] Graduate Research Assistant, Dept. of Civil Engineering, Stanford, CA 94305-4020, aalami@ce.stanford.edu

(1) Relations between different types of information (e.g., between building components, estimating line items, and schedule activities) are not modeled explicitly. This clearly stands in the way of further integration.

(2) Scheduling is activity-based. While we don't object to activity-based schedules, we think that activity-based scheduling stands in the way of further automation. Based on Birrell's (1980) observation that "in construction there are: (1) the resources required to execute the work and (2) the end product to be constructed" we postulate that automation of construction scheduling will have to be primarily based on explicit product and resource models.

(3) Schedules are not easily extensible to different levels of detail. Because construction managers often want to refine the detail level of their schedules gradually, this hinders automation and integration of construction management tasks.

(4) The reasons behind dependencies between activities are not represented Since the reasons behind activity dependencies has to be explicit to enable computers to reason about activity sequencing, this again hinders automation of construction management tasks.

This paper first describes a conceptual project model we developed to overcome these limitations, then briefly introduces a suite of object-oriented software modules that automate and integrate construction management tasks, and finally proposes a template to build construction method models that support the generation of realistic schedules.

CONCEPTUAL PROJECT MODEL

Much like typical CAD tools are not design tools, but merely represent a design graphically and facilitate the manipulation of abstract design primitives (i.e., lines, surfaces, solids), so are today's scheduling tools not planning tools, but simply allow (graphical) representation and manipulation of scheduling primitives (i.e., activities and their dependencies). If we are to integrate project management tasks, we have to raise the semantic level of the information represented and manipulated to the level of project models—much like the level of product information during design has to be raised to product models to integrate design tasks (Tolman 1991; Teicholz and Fischer 1994). Therefore, the core of our integration approach is a conceptual project model, i.e., a neutral representation of all information used during project management, including design, activity networks, schedules, resource plans, construction visualization information, and cost estimates. Neutral means that the representation of information is independent of the participants and applications that use the information.

The basic approach to project modeling is to define classes for sets of objects with similar characteristics. Characteristics are either relationships with other objects, such as the 'decomposes-into' relationships between structures and components, or attribute values, such as the shape and material properties of products. The object-oriented paradigm (Meyer 1988) adds the notion of methods to a class of objects. Methods derive relationships or calculate values. For example, a method for an activity can derive what construction methods are applicable, or a method can determine the weight of a component from its shape and material attributes. Several researchers have developed approaches for project modeling (Björk 1991; Luiten and Tolman 1991; Froese 1992; Gielingh and Suhm 1993; Luiten et al. 1993).

We apply this project modeling approach to construction management and have developed a conceptual model of construction management classes and their relationships. Fig. 1 shows the main classes and relationships in a NIAM diagram (Nijssen and Halpin 1989). In NIAM, a class is represented by a circle and a relationship by a box on a line between classes. Class-inheritance relationships are represented by an arrow from a subclass to a superclass.

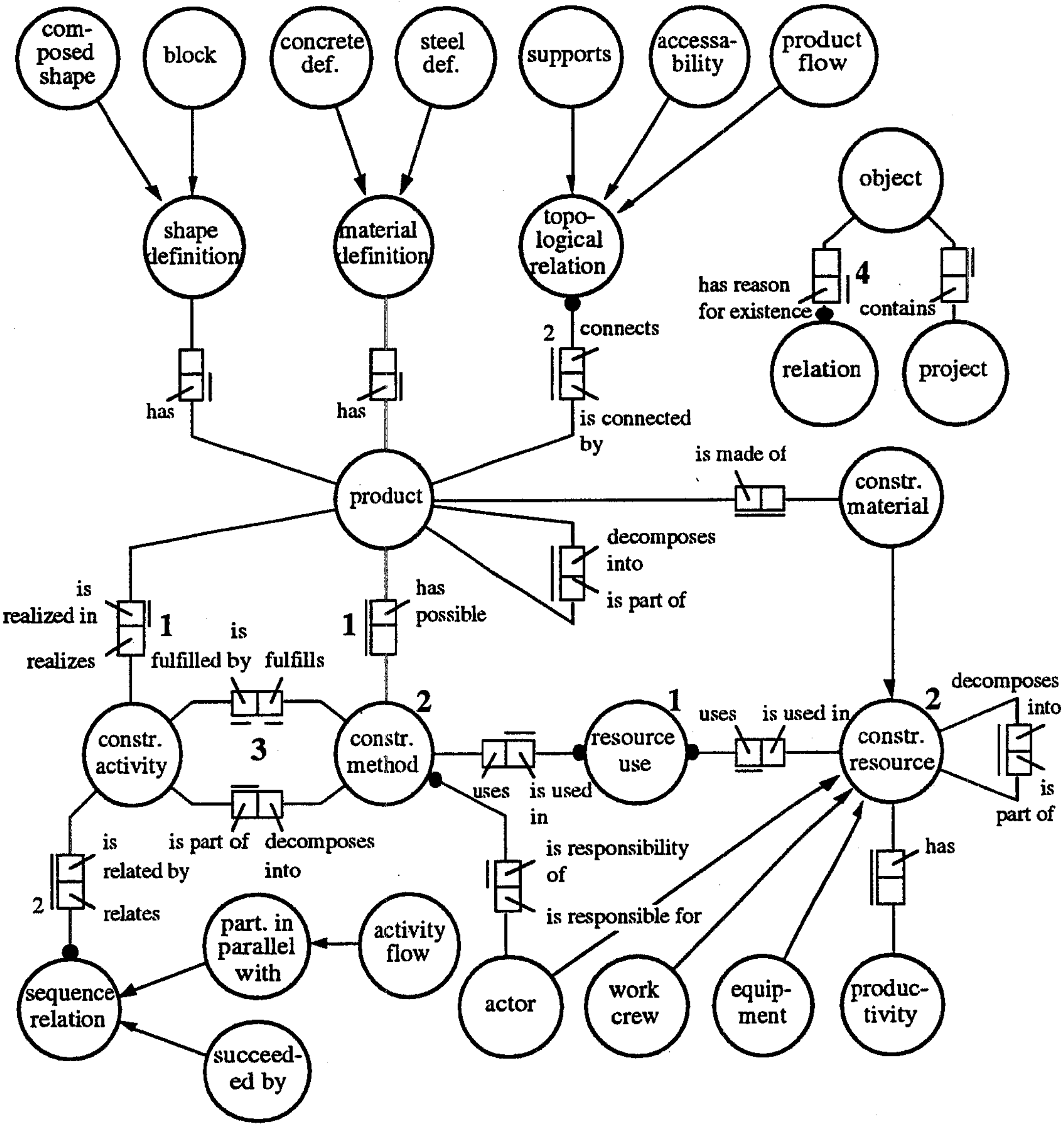

Fig. 1. **Neutral representation of project information; classes and their relations modeled in NIAM (Nijssen and Halpin 1989).**

The conceptual project model in Fig. 1 overcomes the limitations summarized above. In the figure, the classes and relationships that address these limitations are indicated with numbers corresponding to the limitations in the text. The model represents and relates information on a high semantic level and thus explicitly relates products, activities, construction methods, and resources. This addresses limitation 1. For example, with the model the shape of a product can be generally defined as a block with length, width, and height. From this general definition not only a 3D graphic can be derived for visualization, but also the volume and the surface area can be computed for cost estimation. In addition, the model represents resources and construction methods, thus overcoming limitation 2.

With respect to limitation 3, the relationships between activities and construction methods allow flexible reasoning about the schedule and cost estimate at different levels of detail. The model also allows extension of a specific project model with more details in the course of a project. To resolve limitation 4, the model explicitly represents the reasons behind dependencies between products and between activities.

The relationship between activities and construction methods needs some elaboration. Product models often distinguish between function, form, and behavior of a product (Clayton et al. 1995). For example, in the FU-TS decomposition as proposed by Gielingh (1988), a functional requirement of a product is modeled in a functional unit (FU) and the shape and material definitions in a technical solution (TS). The technical solution contains knowledge that evaluates whether the predicted behavior of a solution corresponds to the required functionality. A similar construct is used in our model. Activities are modeled as functions that have to be performed and construction methods as solutions for these functions. For example, the activity 'build-wall' with the requirement 'optimal duration and cost' can be fulfilled by the construction method 'mason-on-site' and its allocated resources. As in FU-TS decomposition, a construction method (a solution) decomposes into activities (requirements) at a lower level.

The project classes *and* their attributes and relationships form the conceptual project model. This conceptual model is very abstract because it is intended to be valid for many different types of projects. For use on a particular type of project, e.g., concrete structures, this abstract model can be specialized to a project type model (PtM). A PtM defines subclasses of the conceptual project model classes that have characteristics and knowledge valid for that type of projects (Tolman 1991). For a specific project, e.g., realizing the concrete structure for a new building of the School of Engineering at Stanford, the PtM is instantiated to a specific project model that contains information for that project in a neutral way. This means that objects are classified and related to each other in such a way that project management applications can derive information from the model and add information to the model. This neutral model of project information forms the basis for a computer tool that supports the management of scope, schedule, and budget.

OVERALL SYSTEM ARCHITECTURE

Based on the model presented in the previous section, we developed a computer tool—SPACECAKE (System for Project mAnagement in Civil Engineering using Computer-Aided Knowledge Engineering)—to support construction managers. Fig. 2 shows the construction management tasks and the information flows supported with our computer system. The construction manager first interprets the design to form a mental image of the building. With this image he/she chooses construction methods, activities, and resources. The system then generates a plan and a schedule for the project. Once these activities are scheduled, the construction process can be simulated and visualized. Based on the activities, allocated resources, and components the construction cost can be calculated. In our computer environment, the first task, interpreting the design, is supported with SME+ (Clayton et al. 1994), an extension of AutoCAD. The other tasks are supported with modules developed in Kappa (IntelliCorp, 1993) and Design++ (Howard, 1995). We use AutoCAD to visualize the simulated construction process.

In the prototype system called SPACECAKE (Luiten and Fischer, 1995), we have tested the usefulness of the conceptual model and system architecture as information models for integrated construction management software. In our experience, the main challenge for further integration and automation of construction management tasks is the development of computer-interpretable representations of construction knowledge.

Construction knowledge, in this context, is knowledge about the applicability of a method to a specific activity, resource allocation to a method, and the decomposition of a method into lower level activities including the linking of the lower level activities to corresponding portions of the product model. In SPACECAKE, we hard-coded this knowledge using redefinition of OO-methods for several classes of construction methods. While this made implementation easy, we foresee problems with maintenance, transfer of knowledge between systems, and customization of the system by end-user in industry.

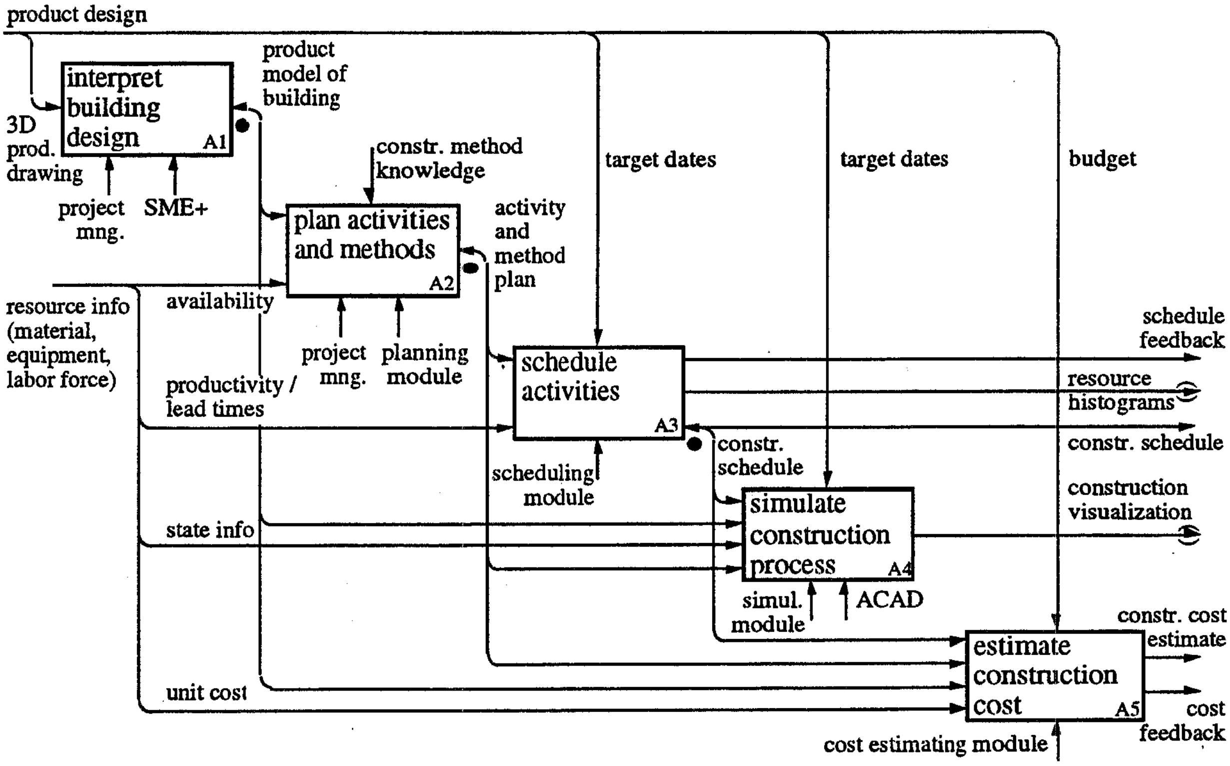

Fig. 2. Five tasks of construction management supported with SME+ (Clayton et al. 1994), **the Scope, Budget, and Schedule Management System, and AutoCAD.**

In a second project—MOCA (Model-based Constructibility Analysis)— we chose to represent construction methods in a model, which is easier to maintain, customize, and transfer (Fischer and Aalami, 1995). The next section reports on this ongoing research effort and proposes a template that makes schedule and cost-related knowledge about construction methods explicit and computer-interpretable.

COMPUTER-INTERPRETABLE CONSTRUCTION METHOD MODELS

In this section, we define and describe a computer-interpretable construction method model. To help overcome the limitations outlined above, these models capture construction method specific knowledge about method applicability, resource allocation, activity generation and sequencing. In addition, these models guide the evolution of a product model by introducing objects that are specific to construction methods, such as zones or temporary structures. Since it is impossible to capture the knowledge about every construction method available in practice, we propose this model as a template to represent construction method knowledge for firms and projects. It is interesting to note that, in medicine, a similar approach to treatment planning is under development. Based on patient models, treatment protocols formalize vocabulary and describe possible treatment methods (Campbell and Musen, 1992).

Several authors acknowledge the importance of considering construction methods or technologies for planning and scheduling. Duffey and Dixon (1993) propose PAR (product, activity, resource) matrices to make the relationships between product elements, activities, and resources explicit. In Construction Planex, construction technologies assign crews to activities (Hendrickson, 1987). In GHOST, construction critics help sequence activities and calculate activity durations (Navinchandra et al., 1988). In similar fashion, Jin et al. (1992) stress the existence of process-oriented knowledge in addition to product-oriented knowledge. They represent process knowledge as methods to represent process-based activity constraints and to complement product-based sequencing knowledge. In MDA Planner, Jägbeck (1994) defines methods "as sets of generic activities required to produce a building object." For the same building part, several methods might be applicable. These methods support the generation of activities. We agree with Jägbeck that methods not only affect resource allocation and activity sequencing, but also activity generation.

We build on prior research efforts by taking symbolic product models of facilities, automated activity generation based on components, and activity sequencing based on component relationships (e.g., supported-by, enclosed-by) for granted.

We illustrate the use of method models for scheduling the construction of masonry walls for the medical gas room at the San Mateo County Health Center's Central Utility Plant. This project is currently under construction by Dillingham Construction Co. We chose this project because we have access to an extensive 3D-CAD model, which has already been linked to the construction schedule for 4D visualization (Collier and Fischer, 1995).

Definition of construction method

Fig. 3 shows how construction methods influence the generation and elaboration of a schedule. In general, methods elaborate (refine) higher-level activities into more detailed, or lower-level activities. After a seed activity is created by the user, the system searches for construction methods that are applicable to this activity. The method model defines the necessary lower-level activities and lower-level components the activities act on. It also contains the necessary sequencing and resource knowledge. This process of activity and component refinement can be repeated as long as more detailed methods are defined for lower-level activities. This strategy builds on Gray's (1986) activity selection rules and supports the generation of process-oriented hierarchical construction plans. As the discussion of this broad schedule generation strategy reveals, a construction method model must contain information about what activities it applies to, i.e., its *domain*, how it elaborates the domain activity into lower-level activities, i.e., its *constituting activities*, how to sequence the lower-level activities, i.e., *activity sequencing* knowledge, what components the lower-level activities act on, i.e., its *constituting objects*, and what *resource requirements* each lower-level activity has. A small construction method model hierarchy and sample methods for the medical gas room masonry walls are presented in Fig. 4. The attributes of the template are described in detail below.

Domain

This attribute contains the activity domain for which a method is applicable; it specifies what activities can be carried out by this method. The value of the attribute is a list of activities. For the construction method "Construct_Wall_In_Courses" (Fig. 4), the domain contains the activities "Build_CMU_Wall_Lift" and "Build_CMU_Wall". Although these two activities are at different levels of abstraction, both a single lift (i.e., the height of a masonry wall a mason can place without raising the scaffold) and an entire wall can be built in courses.

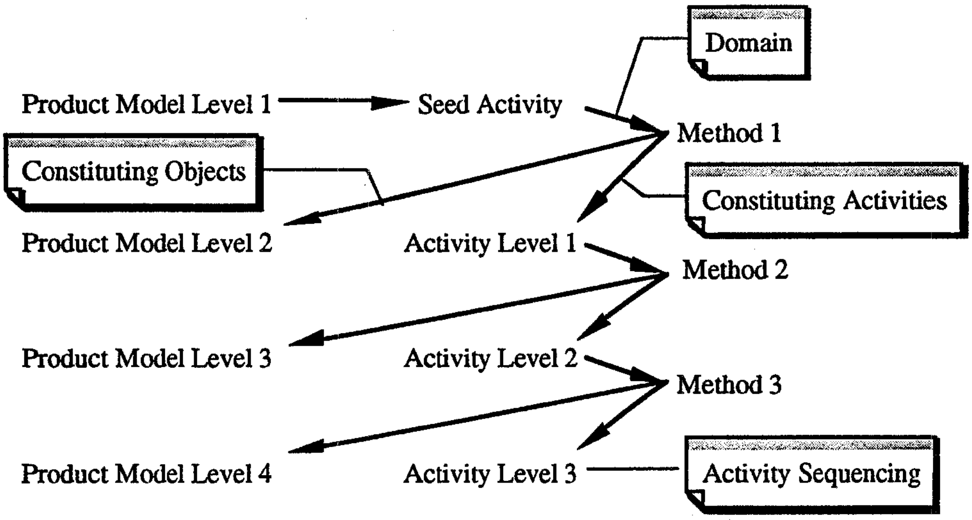

FIG. 3 Product model and activity (construction process model) elaboration strategy using attributes of construction method models

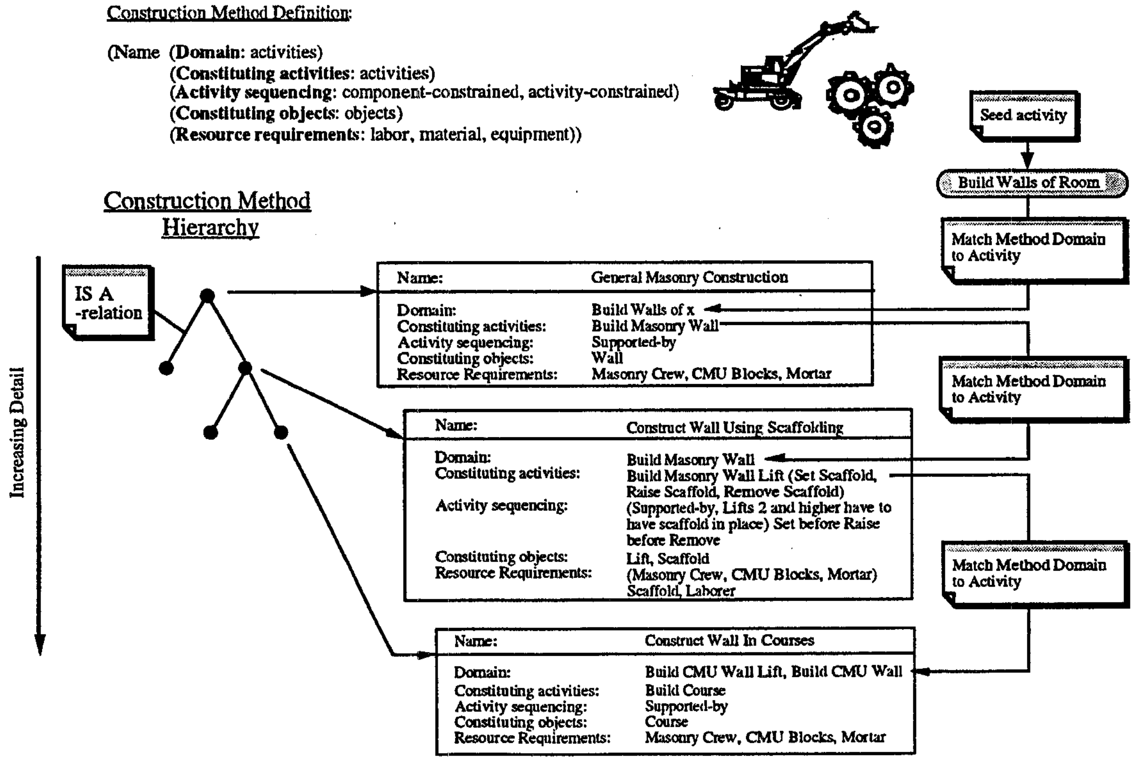

FIG. 4 Definition of construction method model and construction method hierarchy

Most systems discussed above classify construction methods by components they act on. We classify construction methods by activities, i.e., methods are defined for activities and not components. This has two main reasons. First, it is impossible to match a method to a component without knowing what activity needs to be performed on the component. Given a component, e.g., a wall, a planner cannot create a plan unless s/he knows whether the wall should be procured, formed, cast, built, painted, or demolished. For each of these possible activities, a number of methods exist. Each method-activity pair leads to a different schedule and resource needs. For example, planning knowledge about painting columns relates largely to painting, i.e., the activity, and not to columns. Second, many activities in a schedule don't apply directly to components in the product model. Preparatory work or actions to ensure site safety are examples of such activities. While it is possible to generate these activities with component-based planners, it is difficult to elaborate them further without activity-based methods.

Constituting activities

This attribute contains a list of more detailed, lower-level activities that together accomplish the same result as the higher-level (domain) activity. For example, for the method "Construct_Wall_Using_Scaffolding" applied to the higher-level activity "Build_Masonry_Wall" (Fig. 4), the constituting activities are "Build_Masonry_Wall_Lift", "Set_Scaffold", "Raise_Scaffold", and "Remove_Scaffold". To insert these lower-level activities into a schedule, a method needs to know how to sequence them. If an activity relates to a component in the product model, a method also needs to know how to link it to the appropriate component.

Activity sequencing

This attribute describes how the lower-level activities relate to each other and to other activities in the schedule. Presently, two general types of sequence relations are implemented: *component-constrained* and *activity-constrained.* Component-constrained sequence relations are physical constraints. Such constraints include support and enclosure. For example, the activity "Build_Course_1" proceeds "Build_Course_2" since course 2 is physically supported by course 1, the course below it. Activity-constrained sequence relations determine the sequencing of activities based on activity type and not on the components involved. For example, "Place_Formwork" always precedes "Place_Concrete". In this case, both activities refer to the same component, and are therefore not constrained by the topology of the components, but rather by the nature of the work.

The number of sequencing constraints represented for a method or an activity can affect the degree of parallelism or linearity achieved in a plan. For example, introducing enclosed-by constraints will make a plan more linear than a plan generated without such constraints. It is up to the user to turn certain sequencing constraints on or off for the generation of a particular schedule. It is also noteworthy that the sub-networks generated during the hierarchical planning process do not have to be fully self-contained. A fully self-contained sub-network is simply a substitution for the higher-level activity, and the higher-level precedence relationships remain intact. However, refining the network often requires the deletion of the higher-level precedence relationships and the introduction of entirely new sequence relationships to other higher-level activities and to new lower-level activities in other sub-networks. Thus, sequence relations to activities in other sub-networks can also be specified.

Constituting objects

This attribute contains a list of the component classes on which each of the activities in the constituting activities attribute acts. Referring to the construction method "Construct_Wall_Using_Scaffolding" in Fig. 4, the constituting objects slot contains the classes "Lift" and "Scaffold". Components can have a one to one or one to many correlation with the activities in the constituting activities attribute. This mechanism for product and process model elaboration is similar to OARPLAN's mechanisms (Darwiche et al., 1991). An example of a one to one correlation is the matching of the constituting object "Lift" to the constituting activity "Build_Masonry_Wall_Lift". An example of a one to many correlation is the matching of the component "Scaffold" to the activities "Set_Scaffold", "Raise_Scaffold", and "Remove_Scaffold". In the first case, the construction process model was refined by reducing the detail of the component from wall to lift. In the second example, the process model was refined by reducing the activity detail.

Explicitly representing the objects on which activities act in the construction method model allows for construction method specific refinement of the product model. In the medical gas room example, the 3D-CAD model and the corresponding initial product model only show the walls. If a construction method refers to temporary structures, (scaffolding), more detailed components, (blocks), or an aggregation thereof, (courses), the constituting objects attribute can introduce these into the product model. This leads to a process-oriented product model. Please note that zones can also be represented as constituting objects.

Resource requirements

For each of the constituting activities, this attribute specifies the resources, such as labor, material, and equipment, needed. Resources are matched to constituting activities in the same fashion as constituting objects to activities. Depending on the scheduler's choice, resource availability may affect construction method selection, and resource limits may affect activity sequencing in the same style as in Waugh's (1990) ACP (A Construction Planner) system.

We have tested the application of this type of construction method knowledge to support schedule generation for small reinforced concrete and masonry structures (Fischer and Aalami, 1994, 1995). Our next step is to test these models more extensively for a variety of other types of structures, activities, and materials.

CONCLUSIONS

We presented a conceptual model and overall architecture to integrate and automate construction management tasks. The model and the architecture showed the importance of formalizing construction method knowledge to achieve these integration and automation goals. We then proposed an initial template that represents the construction method knowledge we have, so far, found necessary to automate the generation of schedules and cost estimates and to integrate schedules and estimates with each other and with product models of facilities.

We would like to call upon the workshop participants to further develop and test this construction method template and to validate and extend the conceptual model and architecture that bring construction method knowledge to bear in construction process design.

REFERENCES

Birrell, G.S. (1980). "Construction Planning—Beyond the Critical Path." *Journal of the Construction Division, ASCE,* 106 (3), 389-407.

Björk, B.C. (1991). "A Unified Approach for Modelling Construction Information." *Building and Environment*, 27 (2), 173-194.

Campbell, K.E. and Musen, M.A. (1992). "Creation of a systematic domain for medical care: the need for a comprehensive patient-description vocabulary." *Proceedings Ifip (International Federation For Information Processing) World Conference Series On Medical Informatics*, Medinfo 92, Seventh World Congress On Medical Informatics, Geneva, Switzerland, September 6-10, 1992, Lun, K.C., et al. (ed.), Elsevier Science Publishers B.V., Amsterdam, The Netherlands, 1437-1442.

Clayton, M.J., Fruchter, R., Krawinkler, H., and Teicholz, P. (1994). "Interpretation Objects for Multi-Disciplinary Design." *Artificial Intelligence in Design '94*, Gero, J.S. and Sudweeks, F. (eds.), Kluwer Academic Publishers, the Netherlands, 573-590.

Clayton, M.J., Fischer, M.A., Kunz, J.C., and Fruchter, R. (1995). "Behavior Follows Form Follows Function: A Theory of Design Evaluation." *Second ASCE Congress on Computing in Civil Engineering*, Atlanta, ASCE, New York, NY, USA, 310-317.

Collier, E. and Fischer, M.A. (1995). "Four-Dimensional Modeling in Design and Construction." *Technical Report 101*, CIFE, Stanford.

Darwiche, A., Levitt, R., and Hayes-Roth, B. (1989). "OARPLAN: Generating Project Plans by Reasoning about Objects, Actions and Resources." *AI EDAM*, 2(3), 169-181.

Duffey, M.R., and Dixon, J.R. (1993). "Managing the Product Realization Process: a Model for Aggregate Cost and Time-to-market Evaluation." *Concurrent Engineering: Research and Applications*, 1(1), 51-59.

Fischer, M.A., and Aalami, F. (1995). "Scheduling with Computer-Interpretable Construction Method Models." *Working Paper 41*, CIFE, Stanford.

Fischer, M.A., Aalami, F., and O'Brien Evans, M. (1995). "Model-Based Constructibility Analysis: The MOCA System." *Working Paper 34*, CIFE, Stanford.

Froese, T.M. (1992). *Integrated Computer-Aided Project Management Through Standard Object-Oriented Models*. Ph.D. thesis, Stanford University, CA, USA.

Gielingh, W.F. (1988). *General AEC Reference Model (GARM)*. BI-88-150, TNO Building and Construction Research.

Gielingh, W.F. and Suhm, A.K. (eds.) (1993). *IMPPACT Reference Model: An Approach for Integrated Product and Process Modelling of Discrete Parts Manufacturing*. Vol. 1, Research Reports ESPRIT: Project 2165 - IMPPACT, Springer-Verlag, Darmstadt, Germany.

Gray, C. (1986). "'Intelligent' construction time and cost analysis." *Construction Management and Economics*, 4(2), 135-150.

Hendrickson, C., Zozaya-Gorostiza, C., Rehak, D., Baracco-Miller, E., and Lim, P. (1987). "Expert System for Construction Planning." *Journal of Computing in Civil Engineering*, ASCE, 1(4), 253-269.

Howard, C.H. (1995). "Modeling Process and Form for Process Plant Pipe Routing." *CIB Proceedings 180.*

IntelliCorp (1993). *Kappa User Manual V3.0*. IntelliCorp, Inc., Mountain View, CA.

Jägbeck, A. (1994). "MDA Planner: Interactive Planning Tool Using Product Models and Construction Methods." *Journal of Computing in Civil Engineering,* ASCE, 8(4), 536-554.

Jin, Y., Kunz, J., Levitt, R.E., and Winstanley, G. (1992). "Design of Project Plans from Fundamental Knowledge of Engineering Systems." *Proceedings AAAI Fall Symposium- Design from Physical Principles*, Cambridge, MA, 149-154.

Luiten, G.T., and Fischer, M.A. (1995). "SPACE CAKE: System for Project mAnagement in Civil Engineering using Computer-Aided Knowledge Engineering." *Working Paper 40*, CIFE, Stanford.

Luiten, G.T., Froese, T.M., Björk, B.C., Cooper, G., Junge, R., Karstila, K., and Oxman, R. (1993). "An Information Reference Model for Architecture, Engineering and Construction." *Management of Information Technology for Construction*, Singapore, Mathur, K.S., Betts, M.P., and Wai Tham, K. (eds.), World Scientific Publishing Co. Pte. Ltd., 391-406 (CIB W78).

Luiten, G.T. and Tolman, F.P. (1991). "Project Information Integration for the Building and Construction Industries." *Fourth IFIP TC5 Conference on Computer Applications in Production and Engineering: Integration Aspects*, Bordeaux, France, Doumeingts, G., Browne, J., and Tomljanovich, M. (eds.), Elsevier Science Publishers B.V., 469-476.

Meyer, B. (1988). *Object-oriented Software Construction.* Series in Computer Science, Hoare, C.A.R. (eds.), Prentice Hall, Cambridge, UK.

Navinchandra, D., Sriram, D., and Logcher, R.D. (1988). "GHOST: Project Network Generator." *Journal of Computing in Civil Engineering,* ASCE, 2(3), 239-254.

Nijssen, G.M. and Halpin, T.A. (1989). *Conceptual Schema and Relational Database Design: A fact oriented approach.* Prentice Hall.

Teicholz, P. and Fischer, M.A. (1994). "Strategy for Computer Integrated Construction Technology." *Journal of Construction Engineering and Management, ASCE,* 120 (1), 117-131.

Tolman, F.P. (1991). *Integration Core Model for Product Modelling.* C90-PMP-02, CAM-I (also published as TNO report B-91-0885).

Waugh, L.M. (1990). "A Construction Planner." *Technical Report 32*, CIFE, Stanford.