CIFE CENTER FOR INTEGRATED FACILITY ENGINEERING

Scheduling with Computer-Interpretable Construction Method Models

by

Martin A. Fischer
and
Florian Aalami

CIFE Working Paper
Number 41

June, 1995

Stanford University

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

Scheduling with Computer-Interpretable Construction Method Models

By Martin A. Fischer[1] and Florian Aalami[2]

ABSTRACT:
This paper presents computer-interpretable models for the representation of construction methods. These models support the automated generation of realistic construction schedules. They are activity-based to support the selection of meaningful activities and to link schedules at various levels of detail. Five attributes define a construction method: Domain, Constituting activities, Activity sequencing, Constituting objects, and Resource requirements. These construction method models act as a template to capture production knowledge specific to firms and projects. We illustrate the use and implementation of these models by scheduling the construction of masonry walls. These models assist architects, owners, and contractors in studying cost and schedule implications of a large number of design and construction alternatives.

INTRODUCTION

Planning, scheduling, and cost estimating are important activities in the project delivery process. They link the design of a facility to its construction. It takes many years to become an experienced scheduler and estimator, and even for experienced professionals, it is often a challenge to develop realistic and practical schedules and cost estimates. Among other things, a realistic construction schedule must be in equilibrium with the estimate, i.e., the estimate must reflect the cost of the resources required to carry out the proposed schedule, and it must be developed at the level of detail appropriate for its purpose.

Today, the link between a schedule and an estimate is often implicit. Even though estimators and schedulers consider construction methods and resources for scheduling and estimating, schedules often do not represent this information explicitly. Realistic schedules should, however, consider construction methods (Dzeng and Tommelein 1995) and resource availability (Fondahl 1991). Since 1992, M.S. students in the construction engineering and management program at Stanford University have studied this issue in case studies of project planning and control systems on 42 different construction projects in the San Francisco Bay Area. We have not found one case in which the contractor represented methods and resources explicitly in the cost estimate and the schedule to provide a link between the two.

Furthermore, to be useful for decision makers, realistic schedules and estimates must be represented at the appropriate level of abstraction or detail. For example, a subcontractor is concerned with the allocation of its crews on a daily or even hourly basis, whereas the project manager for the owner might be concerned with milestones for completion of major project elements. Again, in the 42 case studies, we have not found one project that integrated the different levels of detail at which project participants created schedules and estimates, i.e., each participant developed his or her schedule from scratch. At best, this practice leads to duplications of effort, at worst it results in inconsistent schedule versions at the subcontractor, general contractor, and owner level. On a few projects, we observed the transfer of data (e.g., material costs) between estimates and the exchange of floppy disks containing schedules. However, even when some

[1] Assistant Professor, and
[2] Graduate Research Assistant, Department of Civil Engineering, Stanford, CA 94305-4020

estimates and the exchange of floppy disks containing schedules. However, even when some estimating and scheduling data are shared, using the shared data is difficult, since the assumptions behind the shared data are usually not made explicit. For example, to produce a realistic update of a schedule developed by someone else, a scheduler needs knowledge about the original assumptions concerning construction methods, corresponding crew sizes and makeup, and productivity.

While automated schedule generation systems have been developed (e.g., Darwiche et al. 1989), they typically do not generate a realistic schedule since they often consider construction method knowledge at a cursory level only. This paper first discusses the process of developing a schedule in practice today and establishes the need for explicit construction method models. It then outlines how these models affect the scheduling process. A review of relevant literature and systems leads to a detailed discussion of our proposed construction method model and its use in the scheduling process.

CURRENT AND FUTURE PLANNING AND SCHEDULING PROCESS

The introduction outlined two shortcomings of construction schedules as observed in practice: the lack of integration of schedules and estimates and the difficulty of sharing information between schedules at different levels of details. To understand the origin of these limitations, we next discuss the estimating and scheduling process. We notice that construction professionals lack tools that make their construction method and resource assumptions explicit (Ikeda et al. 1991) and that support the dynamic and interactive sharing of information between schedules and estimates at various levels of detail. We further observe that scheduling and the resulting schedule depend mostly on the scheduler and that abstract production information (e.g., unit costs) is used to develop cost estimates before a realistic schedule is produced.

Current Planning and Scheduling Process and Software Support

The planning and scheduling process, as shown in Fig. 1, has three distinguishable steps. Given a project description, a 2D blueprint in most instances, the scheduler must interpret the information and determine the scope of the project. After determining WHAT to build, the scheduler needs to address HOW to build it, i.e., select appropriate construction methods and generate corresponding activities and their sequence relations. Finally, the scheduler determines WHEN activities will take place by calculating activity and project durations. Current scheduling software supports the representation and manipulation of activities and precedence relationships, but does not represent the reasons behind activities and their dependencies.

Implicit method and resource assumptions

Besides affecting activity selection and sequencing, construction methods, through their resource requirements and production rates, also affect activity durations. Even though resources are considered when calculating durations, they are often not represented explicitly as part of the activities. While scheduling software packages, such as Primavera's P3 system (Primavera Systems 1991), have supported resource loading of activities for some time, they still require manual input of resource information and do not represent the reasons behind the selection of a particular crew and crew size. This is a more serious limitation than is apparent at first. In the 42 case studies, we have found few practitioners that use resource-loaded schedules in their daily work. When asked why, they remarked that the conditions during construction are always significantly different from the assumptions during planning and scheduling and that it is easier to simply adjust activity durations than to re-enter the resource allocations. Thus, to be useful to construction managers, scheduling software needs to support frequent changes and updates of schedules. This is only possible if the reasons behind activity selection, precedence relations, and resource assignments are represented explicitly.

Islands of information

While most scheduling data is entered manually into scheduling software, some CAD and estimating packages also support the flow of data from a design description of a facility to estimating and scheduling. ASG/Softdesk, through its Datalink product (ASG 1992), and Timberline, through its Precision Estimating software (Timberline 1994), allow a user to assign estimating work packages to project components in the CAD file and to extract corresponding takeoff quantities. Precision Estimating then allows the export of these takeoff quantities and the resources (crews and equipment) assigned through the work packages to P3 activities. This assists in the generation of activities and automates the calculation of activity durations. The user then still has to enter the precedence relationships by hand. Nevertheless, this suite of project management software ensures consistency of takeoff quantities between the CAD model, the estimate, and the schedule and transfers resource assumptions from the estimate to the schedule. However, the links need to be established manually and are unidirectional and static. In other words, one always has to produce the estimate before the schedule, and changes to the CAD model, the estimating file, or the schedule don't propagate to the other two packages dynamically. Furthermore, the software can only show the resource assignments, but it cannot explain them.

Scheduling process dependent on scheduler

As shown in Fig. 1, the scheduling process depends entirely on the scheduler and his or her personal interpretation of the drawings or CAD models and his or her approach to scheduling. It is common knowledge that no two schedulers develop the same schedule for the same project, since each of them makes different assumptions, selects different construction methods, and represents the schedule at different levels of detail. This shows that scheduling is somewhat of an art, but it also hinders the company-wide transfer of lessons learned (Dzeng and Tommelein 1995). It also makes it difficult for personnel other than the original scheduler to update the schedule, since the underlying assumptions are not represented explicitly. This requires constant manual reinterpretation of the schedule and leads to errors and inconsistencies.

Unit costs: abstractions of production information

In current practice, a detailed estimate is typically produced before a detailed construction schedule. This always appears curious to a novice, since one would think that the duration of the project, its activities, and the resources required need to be known before a reliable estimate can be made. It is noteworthy that on first-of-a-kind projects (e.g., heavy civil projects) a detailed time-based analysis is performed as input to the cost estimate. On projects that offer more similarity of elements from project to project (e.g., elevated decks for parking structures), time-related costs have been abstracted to unit costs. As a result, cost estimates can be developed rapidly without a lengthy and costly detailed time or schedule study. This approach to estimating, planning, and scheduling made perfect sense without fast, affordable computers and computer-interpretable process models. Given the speed of computer systems today and in the future, practitioners should no longer have to rely mainly on heuristics, such as unit costs, to rapidly generate cost estimates. The missing link, however, is the lack of formalized, computer-interpretable construction method and resource models that form the basis of construction planning, scheduling, and estimating. By representing the assumptions underlying a particular schedule, such detailed models and corresponding scheduling principles support the rapid generation and adaptation of realistic schedules. This complements the current cost-based analysis of design alternatives with time-based analyses and ensures that cost estimates and schedules are in balance. Therefore, the rest of the paper focuses on computer-interpretable construction method models that aid in the rapid generation of realistic schedules.

In summary, with or without an integrated suite of project management tools, the scheduling process is still largely manual and lengthy. Thus, practitioners typically do not have the time and budget to develop several schedule alternatives to evaluate cost and schedule implications of design decisions and of construction method selections. Following is our vision of how computer-interpretable construction method models support the generation of a schedule.

Future Planning and Scheduling Supported by Construction Method Models

To assist schedulers and estimators in developing realistic and practical schedules and estimates, we envision a knowledge-based environment that integrates construction method and resource information with construction plans, schedules, estimates, and product models of a proposed facility at different levels of detail. Such an environment allows designers and construction planners to evaluate the effect of design and construction planning decisions on project duration and cost, providing a basis for improving the constructibility of a design. Many researchers have developed and described systems in support of a similar vision (see review of related work below). They have demonstrated the feasibility of automating the generation of construction schedules by linking the facility model with the schedule or process model at the component (e.g., slab, beam, column, wall) level. All researchers also recognize that construction methods affect planning and scheduling, and some describe initial construction method models. To help make the schedules generated by automated scheduling systems more realistic, the research presented here focuses on formalizing the representation of construction methods as symbolic, activity-based construction method models.

Fig. 2 shows how construction method models aid a construction manager in producing a schedule. These models capture construction method information—previously brought into the scheduling process only through the scheduler's mind—in computer-interpretable form. They also link the facility description to the schedule. They are the kernel of an integrated and dynamic environment that assists a scheduler in producing schedules rapidly for a number of design alternatives and for a number of construction method choices.

The input to the scheduling process is in the form of a 3D-CAD model. The CAD model is interpreted and linked to a building product model (Björk 1994) that is instantiated for the particular project. Systems such as SME (Clayton et al. 1994) and Design++ (Design Power 1995) provide this link. Via activities, components in the product model are then linked to construction method models. These models generate and sequence necessary activities and determine resource requirements. Once activities and their sequence relationships have been made explicit in a process model, they can be visualized as an activity network and displayed as a 4D model (Collier and Fischer 1995).

WORK RELATED TO KNOWLEDGE-BASED CONSTRUCTION PLANNING AND SCHEDULING

According to (Clough and Sears 1991), developing a schedule consists of three steps: (1) determining the activities required to construct the project, (2) ascertaining the sequential relationships among the activities, and (3) calculating activity and project durations by considering resources and construction methods. These steps depend on each other, e.g., activity sequencing depends on the activities generated, and construction method selection influences activity generation and sequencing. Nevertheless, we find these three steps useful to discuss the contributions of related work.

Activity generation

Research in activity generation has focused on isolating the fundamental principles underlying the identification and selection of activities needed to build a project. Much of the work has developed approaches to link activities to a facility description at the component level. Various approaches have been reported; these range from the user directly assigning an activity to a CAD element (Cherneff et al. 1991) to an expert system assigning an activity to a segment of a project (Gray 1986).

In Builder (1991), pre-defined tasks (activities) are classified according to a component-based work breakdown structure (WBS) and are assigned to CAD elements by a user while s/he is generating the drawing. As a result, a semantic network and a drawing of the project are created simultaneously. Builder successfully integrates construction scheduling and facility design.

However, all scheduling-specific knowledge relates to components, and Cherneff et al. conclude that adequate models are needed to reason about different construction technologies.

Gray (1986) identifies and formalizes construction activity identification and selection rules. These rules fall into three categories: type of work, i.e., distinct activities for different resources; operationally significant function, i.e., distinct activities for work on components with different functions; and operationally significant location, i.e., distinct activities for work carried out in different zones.

Construction Planex (Hendrickson et al. 1987) demonstrates the feasibility of integrating all three scheduling steps in one computer environment. The system first assigns element activities to design elements (project components), then aggregates element activities into project activities, and finally determines appropriate (construction) technologies. Construction Planex shows the importance of developing schedules at different levels of detail and the need for mechanisms to refine and aggregate schedules. In contrast to Construction Planex, we allow the selection of construction technologies before the generation of activities, i.e., the construction methods selected, and not the project components alone, affect the generation of activities.

Darwiche et al. (Darwiche et al. 1989) propose an Object, Action, Resource (OARPLAN) framework to support the generation of construction activities and to integrate schedules and facility models. In OARPLAN, activities are elaborated by reducing the scale of either the object or action in the object-action-resource triad. Winstanley et al. (1993) demonstrate the application of OARPLAN to a full-scale project.

Birrell (1980) discusses the chicken and egg question in construction planning, i.e., are resources needed because of activities—the traditional CPM approach to scheduling—or do activities exist because resources do something. Birrell argues for the latter interpretation, and our approach supports his view of and approach to scheduling. Since construction methods generate and refine activities, activities are based on what work crews do.

Therefore, to automate the generation of realistic schedules, we advocate the use of formal, activity-based construction method models for the generation of realistic schedules.

Activity sequencing

As noted above, CPM software does not represent the reasons behind precedence relationships. Research has focused on identifying and formalizing the factors that determine sequencing of activities and on reasoning methods to bring formalized sequencing knowledge into the scheduling process.

Gray (1986) introduces the following factors: fixing base (or supported-by), flexibility of components (e.g., pipes), covered-by, serviced-by, and protected-by. Echeverry et al. (1991) elaborate on these factors and group reasons for precedence relationships into the following four categories: physical relationships among components, trade interaction, path interference, and code regulations. Kähkönen (1993) reports similar factors from a study in Europe. For manufactured assemblies, Lee and Shin (1990) formalize connectivity relationships between parts and subassemblies to support assembly with geometric reasoning.

In Construction Planex (Hendrickson et al. 1987), knowledge sources for successor identification pre-define physical and resource-related sequence relationships. Similarly, in Builder, Cherneff et al. (1991) implemented precedence-generation rules. Both systems show that precedence knowledge can be implemented in computer systems to automate the generation of activity dependencies.

GHOST applies critic-based planning (Navinchandra et al. 1988). At first, all activities are generated in parallel. Critics then apply constraint knowledge to order the activity list. Critics are classified into different domains (e.g., physics, construction, etc.). Waugh (1990) uses a similar, constraint-based—pre- and post-conditions of activities have to be met—approach to scheduling. The scheduling system steps through time and updates the state of the project at each interval. This results in a one-step treatment of resource and non-resource constraints. OARPLAN (Darwiche et al. 1989) deduces precedence relationships from physical constraints between objects and from relationships between actions (constituents of an activity). Birrell (1980) advocates the consideration of work flow for the sequencing of construction activities and argues that, essentially, all activities in a work flow should be critical. Based on Birrell's suggestion, Fischer et al. (1995) distinguish between core and auxiliary activities to schedule a project. In the same spirit, in their HISCHED system, Shaked and Warszawski (1995) add crew and work flow constraints to the traditional types of CPM relationships.

Our system builds on this activity sequencing research. We make the sequencing factors explicit and categorize them much like Echeverry et al. (1991), grouping physical constraints into *component constraints* and other constraints, e.g., those resulting from trade interaction, into *activity constraints*.

Incorporation of construction methods

All references above acknowledge the importance of considering construction methods or technologies for planning and scheduling. 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 along with 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.

COMPUTER-INTERPRETABLE CONSTRUCTION METHOD MODELS

This section defines and describes our proposed computer-interpretable construction method models. To help overcome the limitations outlined above, these models capture construction method specific knowledge about activity generation, sequencing, and resource requirements. 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 noteworthy 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).

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 (Fig. 3). This project is currently under construction by Dillingham Construction Co. We chose this project because we have access to an extensive 3D-CAD model linked to the construction schedule for 4D visualization (Collier and Fischer 1995).

Definition of construction method

Fig. 4 shows how construction methods influence the generation and elaboration of a schedule. Methods elaborate (refine) higher-level activities into more detailed, or lower-level activities. After the user creates a seed activity, 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 schedules. 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. Fig. 5 shows a small construction method model hierarchy and sample methods for the medical gas room masonry walls. The following sections describe the attributes of the template in detail.

Domain

This attribute specifies the activities to which a method is applicable. The value of the attribute is a list of activities. For the construction method "Construct_Wall_In_Courses" (Fig. 5), the domain contains the activities "Build_CMU_Wall_Lift" and "Build_CMU_Wall". Although these two activities are at different levels of abstraction, 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.

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. 5), 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" precedes "Build_Course_2" since course 1 physically supports course 2. 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. 5, 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 process model was refined by reducing the detail of the component from wall to lift. In the second example, reducing the activity detail refined the process model.

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.

Planning with construction method models

We demonstrate the generation of realistic plans with computer-interpretable construction method models for the masonry wall construction of the medical gas room. Fig. 6 shows a portion of the product model of the San Mateo County Health Center's Central Utility Plant. We have adopted a building product model similar to the RATAS model (Björk 1994). The physical support relationships are explicit in the product model.

The user begins planning by interpreting the components associated with the medical gas room in the 3D-CAD drawing (Clayton et al. 1994). Fig. 7 (a) and (b) show the steps to generate the highest-level (seed) activity and to elaborate it into a sub-network at a lower level of abstraction. In the first step, the user defines the overall intent of the schedule by generating a seed activity; in this case "Build_Room". Alternatively, the user could have started with the seed activity "Demolish_Room". This would obviously lead to a very different choice of methods and result in a different schedule even though it would be based on the same initial product model. Once the seed activity is generated, the system suggests applicable construction methods. At this point, the user can choose to base the selection of applicable construction methods not only on matching domain activities, but also on resource availability. For example, if a contractor knows that s/he does not have access to a crane for a particular project, all construction methods requiring a crane can be omitted. In the example in the figure, the user chooses the construction method "General_Masonry_Construction" because its domain includes the "Build_Room" activity. The system now elaborates the higher-level activity into the constituting activities, e.g., "Build_Masonry_Wall_2".

To elaborate the activities further, construction method models are matched to the new lower-level activities. A search for applicable construction methods shows that the method

"Construct_Wall_Using_Scaffolding" applies to the activity "Build_Masonry_Wall_2". In this case, activity elaboration does not only yield new lower-level activities, e.g., "Set_Scaffold" (Fig. 7c), but also requires the introduction of lifts and temporary structures to the product model. These components, specified in the construction method's constituting objects attribute, are not part of the original product model and are added to reflect the construction method or production process used. Elaboration continues in this fashion (Fig. 7d) until the desired level of detail is reached.

The activity "Build_Masonry_Wall2_Lift2" is both component-constrained through a "supported_by" constraint and activity-constrained, i.e., "lifts 2 and higher have to have scaffold in place". For example, building lift 3 succeeds building lift 2 and raising the scaffold to the right height.

The interaction at different levels of detail of the construction method model and product model generates a hierarchical process model. The kernel of the process model is an activity, Fig. 8. The following triad defines an activity: the objects it acts on, the construction method it applies to, and its precedence relations. The hierarchical process model can be represented in graphical form through an activity network at multiple levels of abstraction. It is noteworthy that a schedule does not need to consist of activities from the same level of detail in the process model. For example, the scheduler elected to elaborate the activity "Build_CMU_Wall" into lower-level activities and to leave the activity "Set_Scaffold" unchanged (Fig. 8). A scaffolding subcontractor might elect to build on this schedule and elaborate the "Set_Scaffold" activity further. Once the activities have been generated at the desired level of detail, the system performs the necessary duration and network calculations according to the critical path method.

It is important to note that the construction method model supports component-based *and* activity-based elaboration of a process model. As a brief analysis of the well-known highway bridge schedule in Clough and Sears (1991) shows, only about half of the activities in the schedule relate directly to a component shown in the project plans. Thus, explicit elaboration mechanisms for components and activities are necessary to produce realistic process models at inter-linking levels of detail.

This example shows the use, versatility, and generality of the computer-interpretable construction method models. Construction methods match to the different levels of the process and product models. Fig. 9 summarizes the interaction among these symbolic models. This interaction leads to an evolving product model and to the generation of a realistic schedule. An activity-based cost estimate can now be developed easily, allowing for the evaluation of many more cost drivers than possible with estimates based on unit costs (Horngreen et al. 1994). This requires the addition of knowledge about costs to the construction method models.

We have implemented object-oriented proof of concept systems in KAPPA (IntelliCorp 1993) and in Design++ (Design Power 1995) on SUN workstations. We have tested these models for small reinforced concrete and masonry structures. Our next step is to extend the testing to larger structures and to additional types of methods (e.g., interior work) to confirm the generality of these method models.

CONCLUSIONS

While construction planning and scheduling literature has recognized that construction methods influence plans and schedules, we argue that construction method selection is a major step in the planning and scheduling process. Construction methods affect activity generation and sequencing and resource allocation. They also affect refinement of the product model with process-oriented elements, such as scaffolding and courses.

The construction method models presented here formalize the assumptions of schedulers through the five attributes: domain, constituting activities, activity sequencing, constituting objects, and resource requirements. The combination of these attributes serves as a template to capture process knowledge specific to firms and projects. For each attribute, knowledge can be represented at various levels of detail. When combined—at different levels of detail if necessary—

these attributes represent a particular scheduler's approach to a project. As firms and schedulers continue to plan with construction method models, they will build up libraries of reusable method models. This implies that schedulers will spend less time on schedule creation and more time on schedule analysis. They will also have to spend significant effort on creating and maintaining the models necessary to automate schedule generation.

The method models support dynamic transition between levels of abstractions of product, process, and method models. They enable the generation of schedules in early project phases when only a schematic product model is available and in later phases when more detailed project descriptions are available. Methods and schedules developed in early phases are thus reusable and form the basis for later schedules. A scheduler simply adds construction method models at the desired level of detail.

Computer-interpretable construction method models provide a link between cost estimates and schedules. They support the concurrent and rapid development of estimates and schedules, thus allowing practitioners to explore multiple design and construction method scenarios in more detail than possible today.

ACKNOWLEDGMENTS

We gratefully acknowledge the support of this work through the National Science Foundation (grant CMS-9309655) and the Center for Integrated Facility Engineering at Stanford. We also thank Bart Luiten and John Kunz for their input.

REFERENCES

ASG (1992). *ASG User Manual.* System ASG, Sausalito, CA.

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

Björk, B.C. (1994). "The RATAS project - an example of co-operation between industry and research towards computer integrated construction." *Journal of Computing in Civil Engineering,* ASCE, 8(4), 401-419.

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.

Cherneff, J., Logcher, R., and Sriram, D. (1991). "Integrating CAD with Construction-Schedule Generation." *Journal of Computing in Civil Engineering,* ASCE, 5(1), 64-84.

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

Clough, R. and Sears, G. (1991). *Construction Project Management*. 3rd. Edition, John Wiley & Sons, Inc., New York.

Collier, E. and Fischer, M.A. (1995). "Four-Dimensional Modeling in Design and Construction." *Technical Report 101*, Center for Integrated Facility Engineering, Stanford, CA.

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.

Design Power (1995). *Design++ User Manual V4.0.* Design Power, Inc., Cupertino, CA.

Dzeng, R.J. and Tommelein, I.D. (1995). "Case-based Scheduling Using Product Models." *Proceedings Second Congress on Computing in Civil Engineering*, Atlanta, GA, Mohsen, J.P. (ed.), ASCE, 163-170.

Echeverry, D., Ibbs, W., and Kim, S. (1991). "Sequencing Knowledge for Construction Scheduling." *Journal of Construction Engineering and Management,* ASCE, 117(1), 118-130.
Fischer, M.A., Aalami, F., and O'Brien Evans, M. (1995). "Model-Based Constructibility Analysis: The MOCA System." *Working Paper 34,* Center for Integrated Facility Engineering, Stanford , CA.
Fondahl, J.W. (1991). "Development of the construction engineer: Past progress and future problems." *Journal of Construction Engineering and Management,* ASCE, 117(3), 380-392.
Gray, C. (1986). "'Intelligent' construction time and cost analysis." *Construction Management and Economics*, 4(2), 135-150.
Ikeda, M., Ohkura, Y., and Waku, A. (1991). "The Development of Knowledge-Intensive Scheduling System." *Transactions of the Construction Management Committee*, Vol. 4, Japan Society of Civil Engineers, Tokyo, Japan, 187-193.
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.
Horngreen, C.T., Foster, G., and Datar, S.M. (1994). *Cost Accounting: A Managerial Emphasis.* Prentice Hall, Englewood Cliffs, NJ.
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.
Kähkönen, K. (1993). "Modeling Activity Dependencies for Building Construction Project Scheduling." *VTT Publication 153*, Technical Research Center of Finland, Espoo, Finland.
Lee, S. and Shin, Y.G. (1990). "Assembly Planning Based on Geometric Reasoning." *Computers and Graphics*, 14(2), 237-250.
Navinchandra, D., Sriram, D., and Logcher, R.D. (1988). "GHOST: Project Network Generator." *Journal of Computing in Civil Engineering,* ASCE, 2(3), 239-254.
Primavera Systems (1991). *Primavera Project Planner P3.* Primavera Systems, Bala Cynwyd, PA.
Shaked, O. and Warszawski, A. (1995). "Knowledge-Based System for Construction Planning of High-Rise Buildings." *Journal of Construction Engineering and Management,* ASCE, 121(2), 172-182.
Timberline (1994). *Precision Estimating Plus.* Timberline Software, Beaverton, OR.
Waugh, L.M. (1990). "A Construction Planner." *Technical Report 32*, Center for Integrated Facility Engineering, Stanford, CA.
Winstanley, G., Chacon, M.A., and Levitt, R.E. (1993). "Model-Based Planning: Scaled-Up Construction Application." *Journal of Computing in Civil Engineering,* ASCE, 7(2), 199-217.

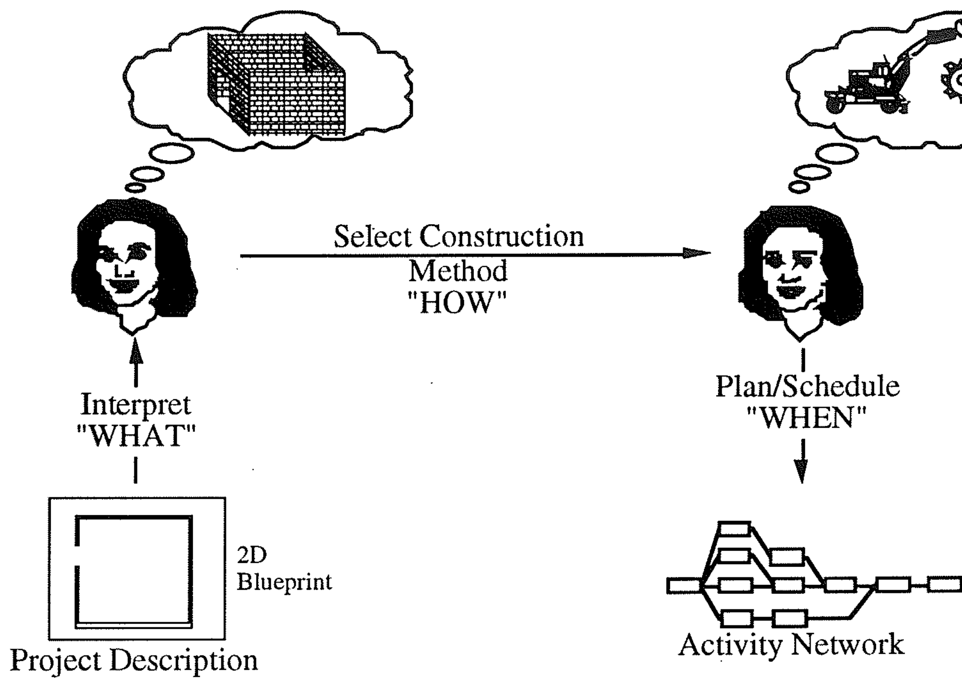


FIG. 1 Scheduling: State of Practice

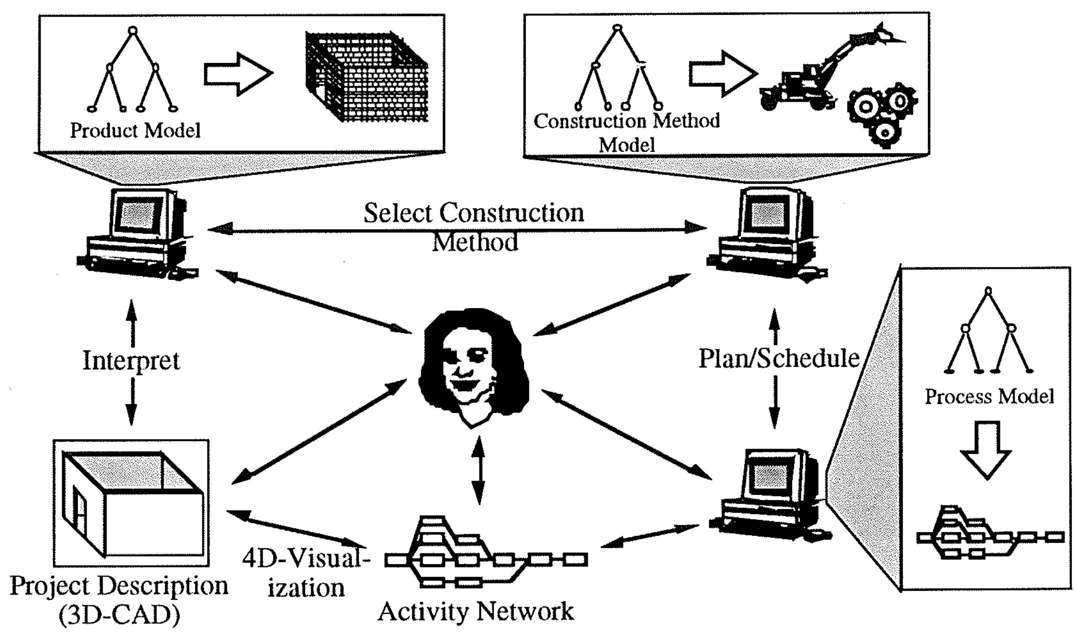


FIG. 2 Models for Construction Methods Support an Integrative, Dynamic Scheduling Environment

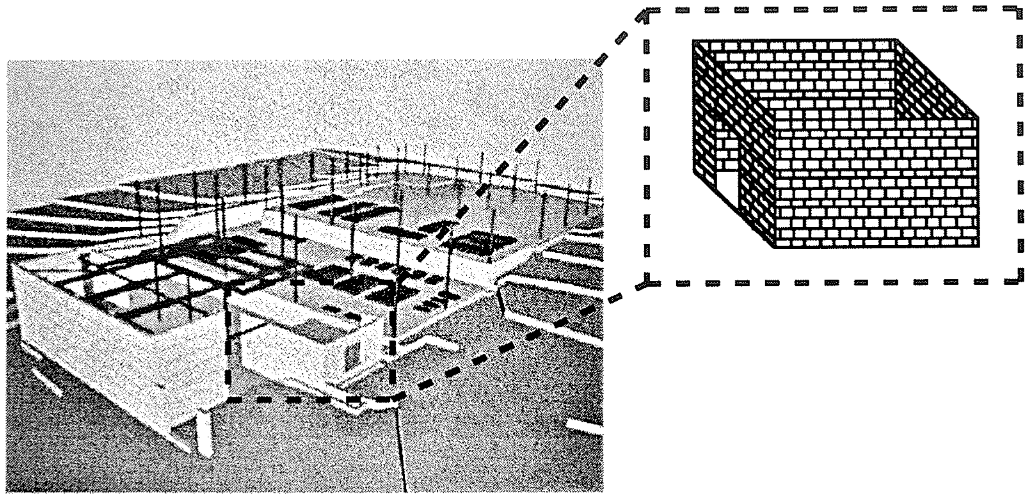


FIG. 3 Medical Gas Room Masonry Walls in San Mateo County Health Center

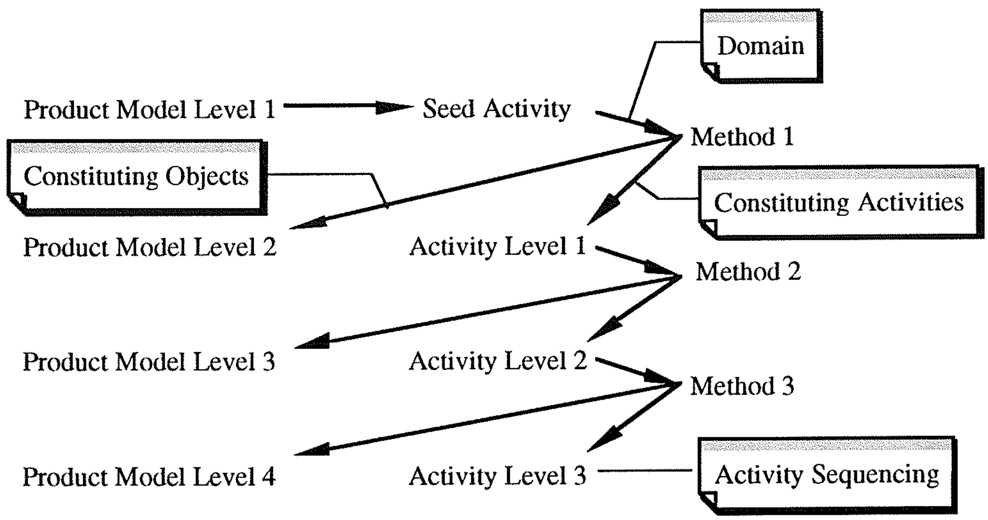


FIG. 4 Product model and activity (process model) elaboration strategy using attributes of construction method models

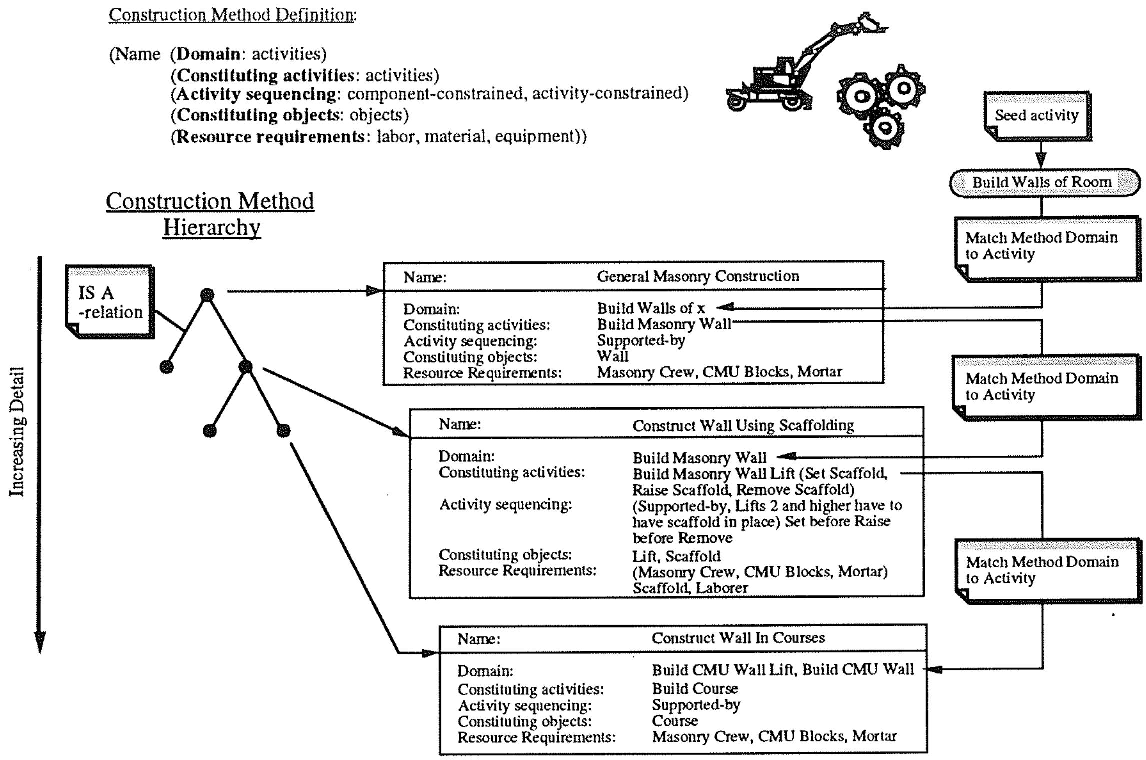


FIG. 5 Definition of construction method model and construction method hierarchy

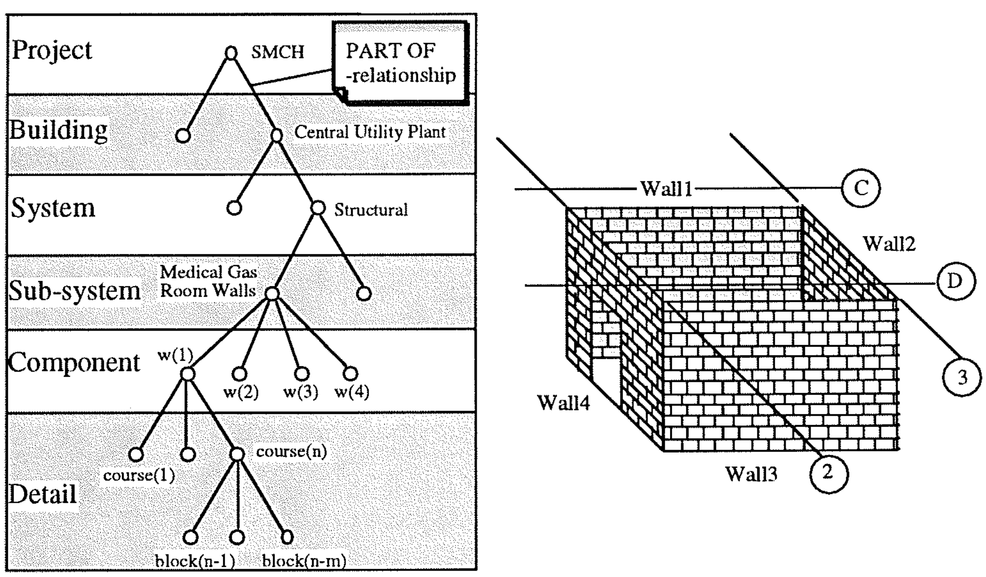
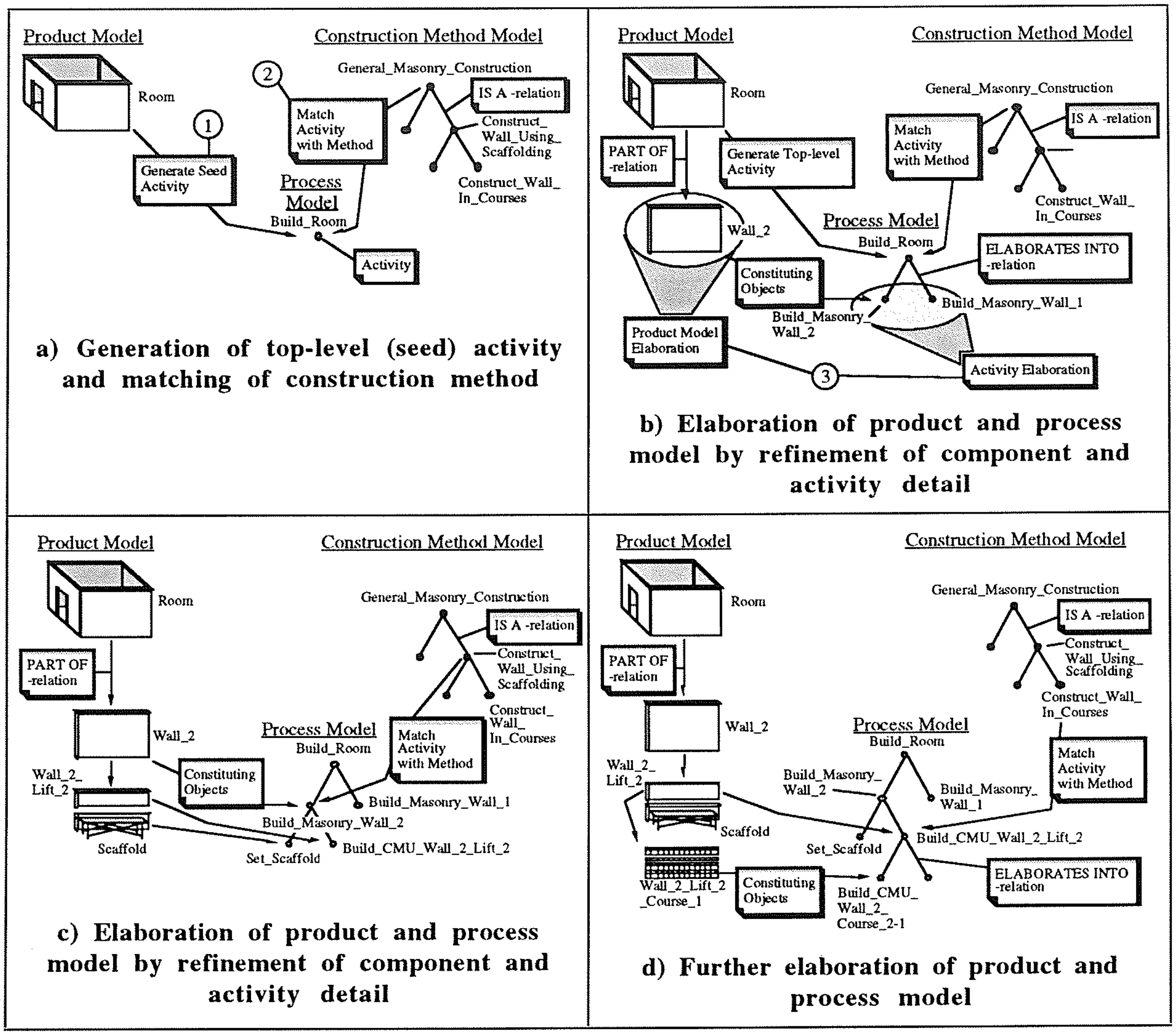


FIG. 6 Partial product model for medical gas room



a) Generation of top-level (seed) activity and matching of construction method

b) Elaboration of product and process model by refinement of component and activity detail

c) Elaboration of product and process model by refinement of component and activity detail

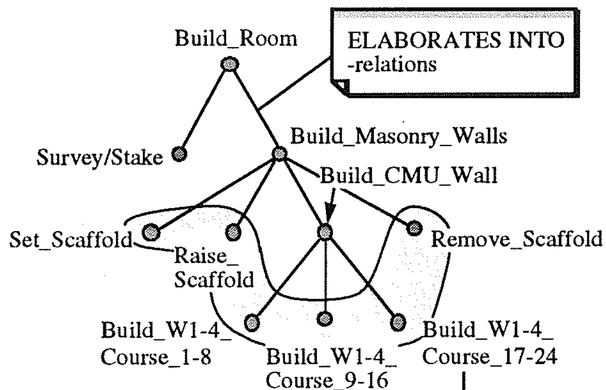
d) Further elaboration of product and process model

FIG. 7 Generation of a process model (schedule) using construction method models

Activity Definition:

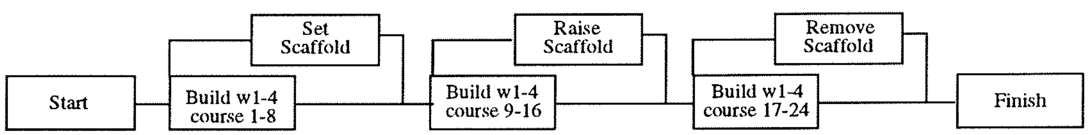
(Name (**Object**: objects)
(**Method**: method)
(**Sequence relations**: activity/relation type))

Process Model
(Activity Hierarchy)



Activity

(Build__W1-4_Course_17-24
(**Object**: Wall_1-4_Course_17-24)
(**Method**: Construct_Wall_In_Courses)
(**Sequence relations**: Build_W1-4_Course_9-16/FS,
Raise_Scaffold/FS))



Levels of detail of schedule shown in shaded area

FIG. 8 Definition of symbolic activity model and activity hierarchy

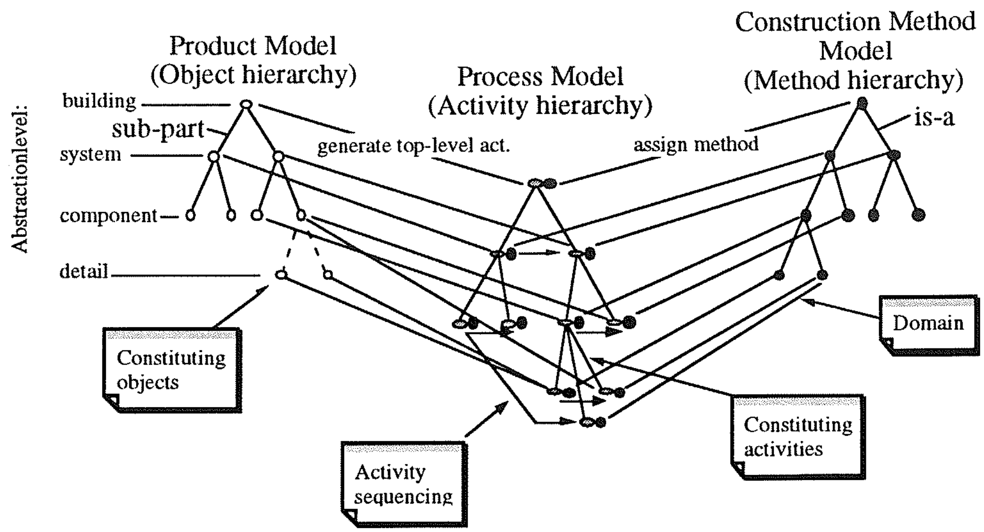


FIG. 9 Overview of product model, process model, and construction method model interaction