
Faculty of Engineering

Faculty Publications

Implementation of ACRM in StartPlan

Kevin Yu and Thomas Froese

© 1996, Copyright, by the Canadian Society for Civil Engineering. With permission from the *Canadian Society for Civil Engineering*.

This article was originally presented at the:

Annual Conference of the Canadian Society for Civil Engineering

Edmonton, AB

May 29 – June 1, 1996

<https://csce.ca/en/publications/past-conferences/>

Citation for this paper:

Yu, K. & Froese, T. (1996). *Implementation of ACRM in StartPlan*. Published in the Proceedings of the 1996 Annual Conference of the Canadian Society for Civil Engineering, Edmonton, AB. <https://csce.ca/en/publications/past-conferences/>

K. Yu and T. Froese, "Implementation Of ACRM In Startplan," 1996 CSCE Annual Conference, Proceedings of the 1996 Annual Conference of the Canadian Society for Civil Engineering, Edmonton, Canada, May 29 to Jun. 1, 1996. CSCE: Montreal, Canada, 1996. Vol. 1, pp. 416-427.

IMPLEMENTATION OF ACRM IN STARTPLAN

Kevin Yu¹ and Thomas Froese²

¹ Ph.D. Candidate, e-mail: yu@civil.ubc.ca

² Assistant Professor, e-mail: tfroese@civil.ubc.ca,
URL: <http://www.civil.ubc.ca/~tfroese/>

Department of Civil Engineering, University of British Columbia,
Vancouver, B.C., Canada, V6T 1Z4,

ABSTRACT

Computer-integrated Construction (CIC) must rely on the project information being shared among various computer applications used by many project participants. This information sharing is based on the development of AEC information models (Froese 94). To date, the central concepts of the models are sufficiently well developed for testing some of their implementation capabilities in a computer program using a real-world example (Froese and Yu 94). This paper introduces AEC Core Reference Models (ACRM) that combine good features of many currently developed AEC information models and our own opinions about the models. It then presents a computer program, StartPlan, that has been developed based on the ACRM to test their implementation capabilities. The paper also describes a construction project that has been used in StartPlan for the test. The results from the test reveal specific advantages and disadvantages of the ACRM and are believed to be useful to the future development of standard AEC information models.

INTRODUCTION

Computer-integrated Construction (CIC) holds the promise of helping to resolve the problems of fragmentation of organizations and information flows that arise from the involvement of many specialized participants on construction projects (Froese 94). The goals of CIC require that project information be shared and exchanged freely by many project participants using various computer applications. This information sharing is based on information models of AEC projects. Our research interest is in the development and implementation of such models.

In this research we have evaluated many currently developed AEC information models and used some of the models to represent and manipulate real-world construction information. We believe that, to date, many of these models are sufficiently well developed to support the development of a computer program to test some of their central concepts and implementation capabilities using a realistic example (Froese and Yu 94). The results of the tests have revealed the advantages and disadvantages of the models, and thus have been useful and significant to the development of AEC information models.

The basic methodology of this research was to first investigate and summarize the good features of many existing AEC project models. The second step was to develop a set of AEC core models that combine those features together with our own perspectives about the models. These new models are called the *AEC Core Reference Models (ACRM)*. The ACRM were then used as the basis of the development of a computer program named *StartPlan*. Finally, an example was used in the program to test the implementation capabilities of the ACRM. Our intention was not to give final solutions to project-specified problems, but to obtain feedback information from the experience of coding the models-based program and from the results of the test.

StartPlan (Froese and Yu 94, Yu 95) is a program that attempts to deal with construction project management by creating template construction schedules. Based on a user's description of a project, StartPlan creates a schedule plan from a library of "good" schedule segments for similar projects and alters it to suit the details for the working project at hand. The system exports the schedule to one of a variety of formats to be used as a "starting template" for creating a new project plan in systems such as Microsoft Project. Not only does this save users significant effort in creating a new schedule, but it helps them take advantage of the scheduling expertise of others.

Also, StartPlan provides a useful test-bed for testing the implementation capabilities of the ACRM. The specific objectives of the program were: 1) to implement the ACRM in an object-oriented computer language, in this case C++, so that computer applications based on it could be developed; 2) to develop an object-oriented central AEC database based on the ACRM, so that many other AEC computer applications could use the central database as a core for sharing project information among many project participants; 3) to develop a dynamic computer program library containing a set of object-oriented central or shared AEC models, so that they could be used by many AEC computer applications during run-time; 4) to use a real-world example in StartPlan to represent a large amount of AEC project information and to translate it between different types of AEC computer applications.

In a broad sense, we believe that StartPlan also serves as a good test-bed for data modeling research, because: 1) it must represent construction information at a detailed

level and in a variety of forms; 2) the construction information is prototypical of real-world generic information in terms of scope, level of detail, and quantity; 3) it must represent a wide range of project information in a generic sense rather than only one type or one piece of specific information for one specific purpose; and 4) it must be able to translate information from and to a variety of existing applications. We consider not only typical scheduling software like Microsoft Project, but also systems like REPCON (Russell 95) that have a fundamentally different definition of a construction activity and applications like common databases or spreadsheets that have a different representation of data. In this regard, we see this application as a good practice arena for general data modeling research. StartPlan is developed in C++ and runs on a PC platform running Microsoft Windows.

This paper first discusses the development of the ACRM. It then introduces StartPlan by presenting its basic functionality and components, including how the ACRM support the development of the program and how their implementation capabilities are tested. Next, the construction project example used for the test is described. The paper finally outlines the results of the test that reveal the advantages and disadvantages of the ACRM.

AEC CORE REFERENCE MODELS (ACRM)

AEC project information models represent AEC project information that can be shared among all project participants using various computer applications. To date, the development of the project models has reached a certain stage in terms of their common objectives and quality (Froese 95a, b). Some of the major examples are the Unified Approach Model (Björk 92), General Construction Object Model (GenCOM) (Froese 92), Building Project Model (BPM) (Luiten 94), Information Reference Model for AEC (IRMA) (Luiten et al. 93, Froese 93a, 94), the Information/Integration for Construction (ICON) project (ICON 94), ATLAS LSE Project type Model (PtM) (Tolman, Bakkeren, and Böhms 94), Generic Reference Model (GRM) (Reschke and Teijgler 94), and the STEP Building Construction Core Model (BCCM) (ISO 94a).

Our investigation of these models, however, has indicated that the lack of a uniform approach and different focuses of each version of the models must be resolved in order to lead to standard AEC information models (Froese 95a). The ACRM are a set of AEC core models that are based on those existing models and attempt to encompass most of the good features of the models. The basic methodology used in the ACRM is to represent different perspectives from different models at various layered implementation levels. In particular, different types of AEC applications are represented as different views at high abstraction levels and more detailed application models are applied at lower levels.

A proposed AEC object layering is shown in Figure 1 (Froese 94). The AEC_Core_Model entity contains all of the AEC related classes that have been generally accepted in different domains in the AEC industry. All of the view models at the second level are refinements of the AEC_Core_Model, but focus on different types of AEC applications. For example, the Product_View_Model focuses on products (e.g., buildings and their components), while the Process_View_Model on construction process (e.g., construction activities), and so on. Each of the view models may still have some of the entities defined in another view model but this is essential only if the included entities affect those in the view model. The models at the upper two levels are conceptual models. They will support application models and information exchange;

and the actual application models will be implemented at lower levels. For example, the Scheduling_View_Model refines the Process_View_Model and sits at the third level.

The StartPlan computer program that we have developed for the test directly adopts the Scheduling_View_Model and also uses some other entities defined in the other views for the application purpose. The complete model used in StartPlan is depicted in Figure 2.

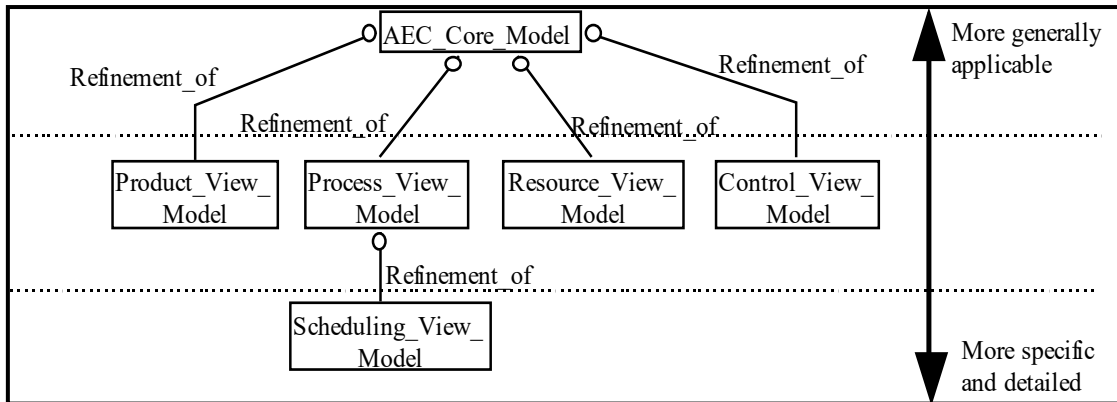


Figure 1: Layers of Standard AEC Reference Models

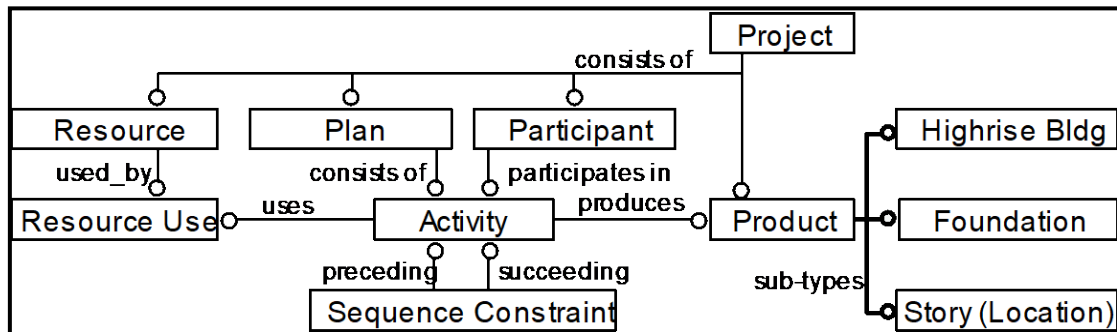


Figure 2: The StartPlan Models

STARTPLAN, SYSTEM FUNCTIONALITY

The fundamental functionality of StartPlan is to generate template construction plans (Froese and Yu 94, Yu 95). When the user executes the program, it asks questions about the project at hand. StartPlan then creates a template plan in a project document by first applying appropriate rules in a rule base document and retrieving good schedule segments from a reference document (these two documents contain experts' expertise and are part of the StartPlan expert system). After that, StartPlan opens one of its various viewers to display the project and plan information stored in the project document. The user can then alter the template plan and translate it out to a desired format such as MS Project through a StartPlan translator.

SYSTEM COMPONENTS

As shown in Figure 3, the StartPlan system consists of a Project Document, a series of Viewers and Translators, and the StartPlan Expert System (SPES) that is made up of a User Interface, an Inference Engine, the Rule Base Documents and the Reference Documents. This section describes each of these basic StartPlan components (a few of the features described here were not included in the prototype implementation, but most were implemented and tested as discussed later in this paper).

Project Document

The project documents are the documents that the user works on to make construction plans for the project at hand. The template schedules that StartPlan creates are placed into the project documents where they can be further modified. At run-time, a project document contains a project object and a plan list object, with the former holding specific project information such as project name, location and project type, and the latter containing plan information such as activity schedule plans. The project object also has objects such as a resource list, a participant list, and a product list—all of this information is linked into the project plans. The project and plan list objects in the document together represent a large amount of project information. The project documents are directly based on the ACRM models. That is, the data structure of the project documents is the direct implementation of the entities in the ACRM, e.g. the Activity, Resource, Component, Result, Participant, and so on.

Document Viewers

The document viewers are used to display the various types of AEC information in the ACRM-based documents. These viewers provide a several user interfaces for different purposes, such as tabular activity breakdowns or Gantt charts for scheduling or

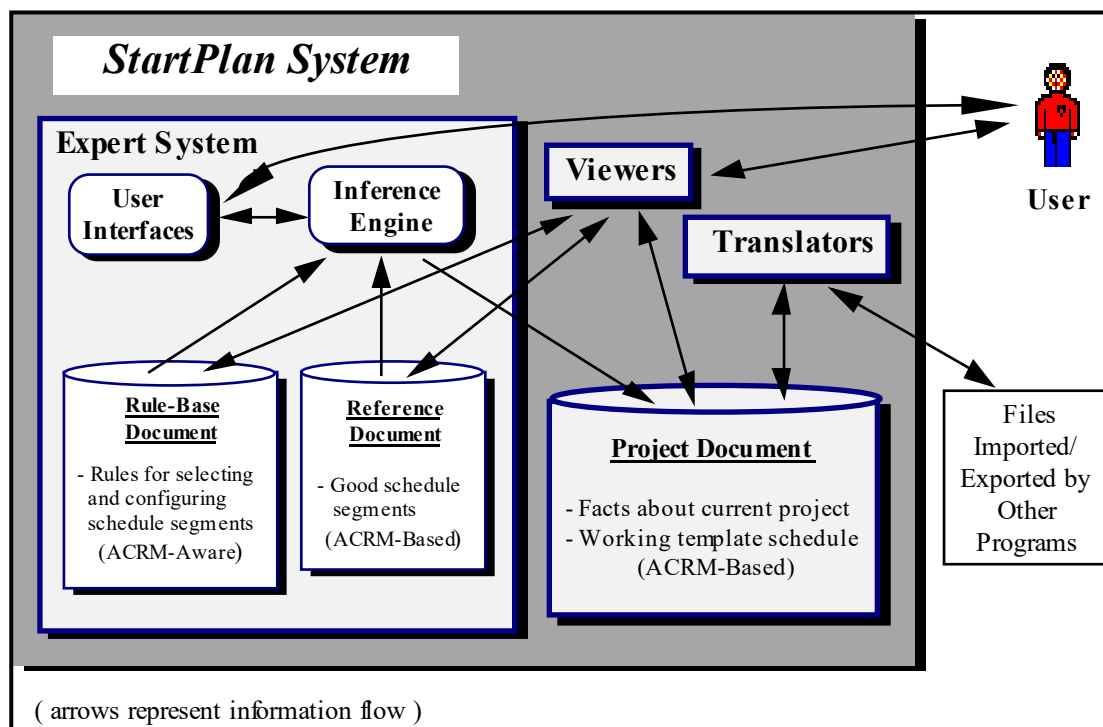


Figure 3: StartPlan System Components

resource lists for estimating. While the document-based databases are implemented independent of any specific viewers, the document viewers are largely modular components of the system and are normally document type-dependent. This means that each type of viewer is designed to work for a certain type of document. When the system is invoked, a series of viewers will be automatically associated with a document. Once the user modifies the document through one of the viewers, all of the others will be automatically notified of the changes. In the future, generic viewers could also be developed to present all types of objects in all different documents.

Translators

The translator module in StartPlan serves to integrate different kinds of applications on the basis of the ACRM concepts. The information sharing is achieved by translating project information between StartPlan and other applications. There are a number of translators in StartPlan with each working only for a unique type of application. The StartPlan translators must understand the data structure defined by the models and thus are able to interpret the StartPlan information. Through the translators, project information in StartPlan can be shared by many other computer applications. Moreover, the information from other applications can be imported into StartPlan through the appropriate translator and exported to another application through a different translator. Hence, the translators play an important role in achieving the information sharing and exchange between various computer applications and testing the implementation capabilities of the ACRM.

StartPlan Expert System (SPES)

The StartPlan Expert System (SPES) (Yu 95) consists of a user interface, an inference engine, the reference documents and rule base documents. It captures expertise about construction control and planning. Through SPES, the knowledge can be retrieved, manipulated and translated in order to be used. SPES is the component that actually constructs the template construction plans in StartPlan.

Reference Document

The reference documents are databases that store the library of good schedule segments. In particular, each schedule segment is represented as a plan object that contains a list of activities, e.g., all activities for a typical floor. At run-time, when the expert system is executed, a series of plans in the document are selected by the inference engine according to the rules. These plans are then combined into one complete plan and stored in a project document object. The reference documents also contain information such as resources, project participants, and products. These objects are linked to the activities in each of the segments and the links are retained when the segments are transferred to the project document.

One of the features of the reference documents is that they are not encoded in the program, rather, they are separate files that are loaded at run-time. Therefore, they can be re-configured and revised whenever needed. Moreover, since the reference documents are written in ASCII file formats, they can be modified outside of StartPlan by any ASCII editors. This provides a convenient and flexible way of modifying the system intelligence.

Although the overall design of the reference documents in StartPlan is not restricted to any specific type of project, the contents of each reference document is likely to be project type-dependent. That is, each reference document is only suitable to one type of project. For example, one document could be made to create plans for a typical reinforced concrete high-rise building, and another to deal with highway construction. The users can create reference documents for different types of projects and save them in different files accordingly. Only one of the documents will be selected at run-time to suit the project at hand. This feature provides flexibility to deal with different types of projects for different types of users.

Rule Base Document

The rule base documents contain rules that the inference engine uses to reason. The rules are needed to retrieve and configure the appropriate schedule segments from the reference documents. They are project type-dependent-thus each rule base document only works for a specific reference document.

The design of the rule base documents in StartPlan is similar to that of the reference documents. First, they capture expertise; and for the same reasons, they must be able to be updated over time. Second, they are not encoded in the program so that they will be read at run-time and can be modified any time to suit the needs of different types of projects. Third, the file format is of ASCII type; hence, they can be flexibly edited by any ASCII editors.

Inference Engine

The Inference Engine in StartPlan is a software “machine” that contains a set of processing methods to query users, reason with rules, retrieve information from the reference and project documents, and write new information into the project document. It takes a reference document, a rule base document and the information about the project as parameters. During user interactions, the inference engine creates questions about the project according to the rules and retrieves the answers from the user through the user interface. It then applies the answers to the corresponding rules as the premises to find out conclusions. The conclusions are used to identify a set of schedule segments. Then all the schedule segments are retrieved and assembled with appropriate accumulation of each segment by its respected repeating time (e.g., a schedule segment for a typical floor will be accumulated 10 times for a building with 10 typical floors) as one complete plan into the project document.

SPES User Interface

The expert system user interface is designed to bridge between the rule base documents and the users. It provides enough fields to display questions generated from the rule base documents and to obtain answers from the user. Through the window type dialogs of the user interface, the users are able to provide information about the working project by answering the questions. The user interface is rule-independent. This means that it is not restricted to certain types of questions. Any question contained in the rules can be viewed and asked in one of the dialog fields. With this feature, when the rule documents are modified, there is no need to change the user interface.

Motivation of Using the Templates And SPES

Using the templates and expert system enhances the testing ability of StartPlan to implement the ACRM models in the system. First, the templates are represented as the model-based project documents. Since the templates are intended to give the user a good starting point for a good construction schedule, they should contain sufficiently comprehensive and detailed information about the project. Hence, they are a good test for the capability of the models to represent project information correctly and completely. Second, the templates must be built from the ACRM-based reference documents that consist of good schedule segments. Thus, the data stored in the documents must be well structured and organized. Likewise, the experts' knowledge must be well organized so as to be represented in the reference documents. Consequently, the reference documents are another good test for the ACRM to represent complete and well-organized real-world information.

Also, using the templates and the expert system can save a great deal of the user's time to create a complete project schedule for a relatively large project. For example, to create a similar activity list for a 15 floor condominium using conventional means, i.e., typing in the activities individually from scratch, may need many hours. However, SPES can produce 300 activities in a 5 minute session including the user interactions with the user-interface.

Additionally, from our experiences, we have realized that many people do not use scheduling software to its full potential. Possible reasons are: 1) they do not have the experience to design a fully detailed schedule that contains all the information needed; 2) the effort required to develop such a schedule might be too great; 3) the software they use is not capable of allowing users to create such a schedule because it is not able to represent information to the desired detailed level. One solution would be to provide schedule examples or templates that the users can start with and work on to modify the specifics of their own projects. In fact, a good schedule should have various appropriate representation formats along with the detailed descriptions about the scheduling information. Such schedules can be easily obtained from the templates with few chances of losing important information that good schedules must have. Once the templates are prepared by the system, the users can start to modify data to suit the project details at hand by imitating the sample data descriptions and formats in the templates. However, there are enough details of good schedules that are dependent upon project specifics that it would be difficult to develop either a small number of templates that can cover many applications, or enough typical templates to suit many types of projects. The expert system in StartPlan is intended to overcome this problem by allowing the user to provide project information and applying appropriate rules to create a new template suitable for the type of the working project.

Moreover, using the expert system allows the user to take advantage of others' expertise. In the reference and rule base documents, human experts' knowledge-that is considered to be perishable, scarce, vague and thus difficult to apply-is captured and organized in an appropriate way. Hence, through the expert system, the users can easily, quickly and directly adopt this knowledge in their own practice. This can be of great help in terms of ensuring the quality of the schedules.

IMPLEMENTATION OF ACRM IN STARTPLAN

The implementation of the ACRM models in StartPlan was one of the major objectives of the research for several reasons:

- the models provided the underlying structure needed for StartPlan's database documents,
- StartPlan allowed us to test the use of the ACRM models for implementing model-based applications,
- StartPlan allowed us to test how well the ACRM models support central AEC databases and file exchange

Implementing ACRM In An Object-Oriented Programming Language (C++)

The ACRM models are high level models that support the development of data modeling at lower levels. The appropriateness of the ACRM models cannot be fully examined until they are implemented in computer programming practice. The implementation of the ACRM in a computer language, in this case C++, is achieved by representing two of their fundamental model elements, the entities and their relationships (Yu 95). In particular, the entities in the ACRM can be directly interpreted as C++ classes (Figure 4). The relationships between the entities in the ACRM fall into three basic categories: aggregation type (a "part_of" relationship), specialization type (a "Type_of" relationship) and Association Relationships (any types other than these two). All of the three types of the relationships can be implemented in C++ by using appropriate mechanisms such as class inheritance or run-time object pointers (Yu 95). The significance of this practice is that the ACRM models can be implemented in a computer language. Also, the fact that object-oriented techniques are used for both the modeling and the implementation language makes the implementation easier and the correlation of concepts between the models and the language closer.

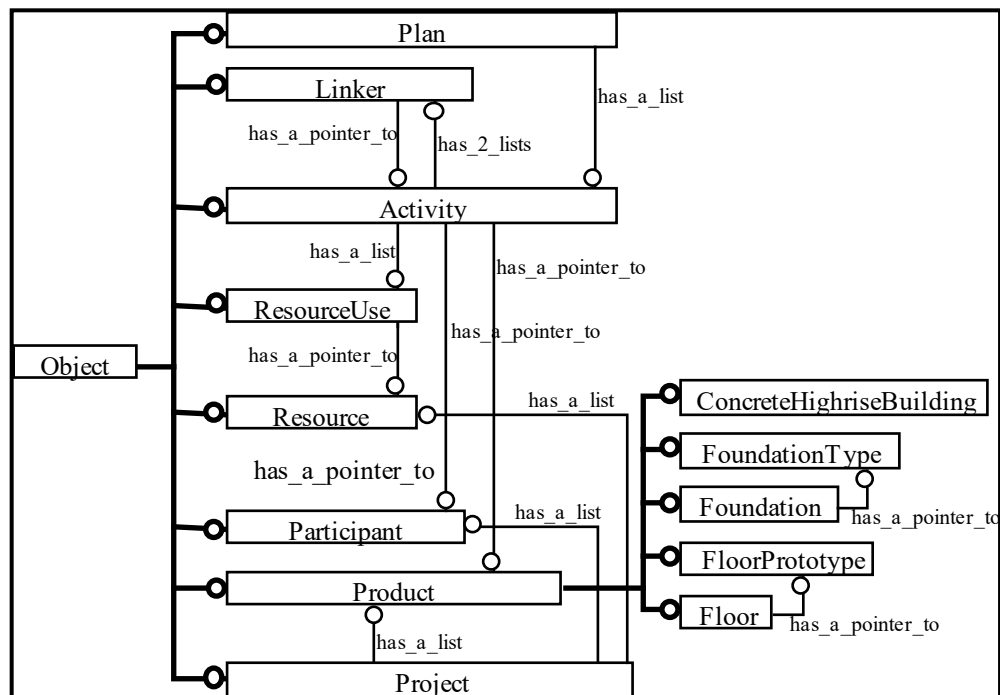


Figure 4: StartPlan C++ of the ACRM

Support For Shared AEC Models

The ACRM models, which define many basic concepts of the AEC domain such as Activity, Resource, Participant, Control, Product, etc., can be shared as central AEC information models by many computer applications at run-time. One of the implementation techniques to achieve this is to use a dynamic-link library (DLL), that is, an executable library module containing compiled C++ classes. The classes thus can be shared by multiple different applications to create their instances at run-time. As discussed above, the ACRM models have been implemented in the form of C++ classes in which data structures are defined according to the contexts of the entities in the models. A DLL that contains all of these classes has also been developed, thus many applications can be developed based on this DLL and can share it to create run-time objects for their own purpose. The importance of this test is that not only can they all take advantage of the standard models, but also this makes the information exchange between the applications easier and more efficient since they all adopt the same data structures for their project information. Moreover, since the ACRM define generic concepts, the applications that share them can be different types dealing with different AEC project functions. For example, an estimating program and a scheduling program can share one single model of “Activity” defined in the ACRM since the “Activity” is generic enough to provide all activity information needed by both applications.

Support For Central AEC Databases And File Exchange

Central AEC databases store information about all project objects. Unlike sharing the core models directly, shared databases provide the quickest and most direct way to share project information. Since the information in the central databases is to be transferred, translated, and eventually used by many different applications, its data structures must be generic enough to represent, manipulate and manage all types of AEC project information, and thus must be supported by generic AEC core models such as those defined in the ACRM. In StartPlan, both the reference and project documents that are developed based on the ACRM can be used as central project databases (Yu 95). StartPlan itself is in fact a program that not only creates but also uses the central databases directly, while through translators the databases can also be shared by other applications. The key point of this practice, however, is that since the StartPlan documents are based on the ACRM, they can store all types of the information that other different applications would need. In other words, the ACRM can support central AEC databases to be shared by many different AEC computer applications.

In addition, the project documents can also be used for testing the capabilities of the ACRM to support file exchange. In this sense, the project documents provide a vehicle to carry the data between two applications. This approach involves one application file being translated into a StartPlan project document file and then into another type. This mechanism has the following advantages. First, it requires only one translator between an application and the ACRM-based program to transfer the application files. Second, it ensures that no important data will be lost because the ACRM-based files are able to represent and hold all types of data provided and needed by many types of applications.

AN EXAMPLE

The Project And The Plan

A construction project that was originally developed in REPCON (Russell 95) was selected for the test in StartPlan. The example is a fictitious project that describes a mid-rise residential building with 1 main floor, 9 typical floors and 2 underground parkades. The building structure is of reinforced cast-in-place concrete. Although it is not a specific real-world project, it can be treated as a typical construction project since it has been used in REPCON as a standard template to create many real-world projects and has proven to contain most of the information that a similar real-world project would have (Russell and Lamman 96).

Any plan in REPCON must define locations that will be linked to REPCON activities for the scheduling purpose. A typical REPCON activity represents a type of work occurring at a series of locations. There are a total of 26 REPCON locations defined for this project. Aside from the locations, there are also 66 REPCON activities, each associated with multiple locations. These 66 REPCON activities would correspond to 350 tasks in a traditional CPM system that treats the work at each location as a separate activity. Other necessary information such as contractors and resources is also provided in the REPCON plan.

Translation And Use of The Example

To use the example in StartPlan, the REPCON plan was translated into a StartPlan reference document. First, the REPCON locations were translated into StartPlan Products since REPCON locations typically refer to the physical building locations being worked on. In particular, all the characteristics of a REPCON location were represented as the attributes of a Product object in StartPlan.

Second, the REPCON activities were translated into StartPlan activities. A typical REPCON activity associated with multiple locations was translated into a number of StartPlan activities with each representing the work at one location and holding an attribute of a StartPlan Product object representing the location. Figure 5 shows the mechanism of this translation. In addition, the REPCON precedence relationships need to be broken down to be represented between the StartPlan activities. These REPCON relationships could, in some cases, represent the relations between two activities with one preceding the other at only one location, but in other cases with one preceding another at all locations where both occur. The test has shown that the ACRM-based C++ class Activity has no difficulty in interpreting this type of information and the ACRM-based reference document has no difficulty in representing the information. All the StartPlan activities with the same Product were grouped as a schedule segment (a plan object) in the reference document.

Finally, the REPCON contractors and resources were directly translated into the StartPlan Participant and Resource objects respectively with the same attributes. In particular, both the contractors and resources objects are related to a Project object and are linked to certain activities in the Plan objects of the reference document.

Once all of the REPCON project information was translated into the StartPlan reference document, it was able to create new template plans for mid- or high-rise concrete buildings through SPES in StartPlan. In the test, a plan for a 25 floor building containing approximately 600 StartPlan activities was created in StartPlan in less than 10 minutes on a 486 PC. The template plan has also been translated into a Microsoft Project file providing the necessary information for MS Project models.

Results

Some of the advantages of the ACRM models revealed by the test in StartPlan are outlined as follows (Yu 95):

- The basic concepts of the entities in the ACRM such as Activity, Resource, Participant and Product, and their relationships are appropriately defined. First, they are able to describe the real-world situation accurately, completely, and concisely. Second, they hold the promise to support more detailed or lower level models in computer applications. In our case, the Scheduling_View_Model is used as an application model to deal with a construction scheduling problem.
- The implementation capabilities of the ACRM are at least as good as that of other similar types of software, such as Microsoft Project, because StartPlan can basically describe the same type and amount of information. The test shows that all the project and scheduling information in REPCON and Microsoft Project can be successfully represented in StartPlan. In some situations, the test indicates that the models used in StartPlan are better than those in other software. For example, StartPlan recognizes the importance of building products using the entity Product and emphasizes the relationship between it and the Activity. However, Microsoft Project does not have a distinctive definition of a building product and, therefore, is unable to take the relationship between a product and a task into account when scheduling. Also, the definition of a product in StartPlan is more generic than a location in REPCON. StartPlan emphasizes the nature of building products instead of only their physical spacing relationships. Thus, a product object in StartPlan can be used not only for scheduling, but also for other purposes such as estimating in an estimating program.

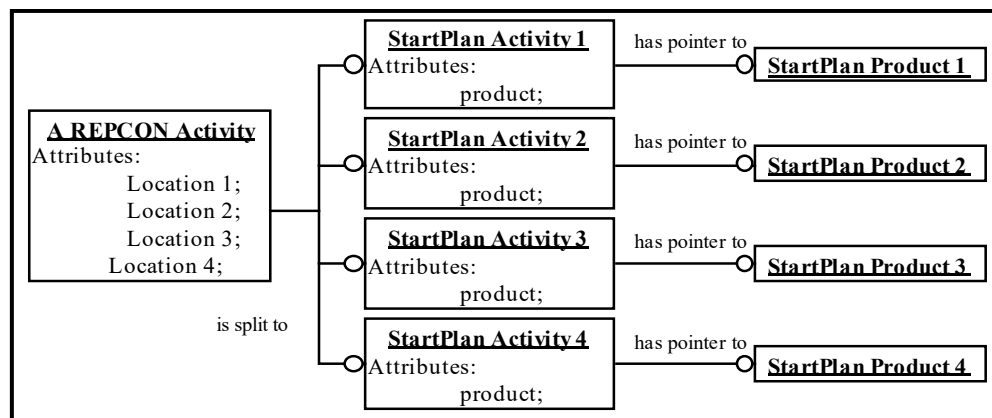


Figure 5: Translation Of A REPCON Activity To A StartPlan Activity

- The entity Activity, which is the core of the ACRM, is generic enough to describe the nature of a unit of construction work in a computer information form and to support different views of it (in our case, the scheduling activity view). It can also support the translation between different types of activities. In the test, we have successfully translated REPCON activities into Microsoft Project tasks through StartPlan activities.

Despite these advantages, the test also shows some points where improvements are needed to the models:

- Although the Activity entity is generic enough to represent any type of work in a construction project, its lower level implementation still needs to be detailed enough to suit the purpose of a specific application. For example, the Scheduling_View_Model is the lower level view of the AEC_Core_Model. We use the Scheduling_View_Model as the application model of StartPlan. Nevertheless, subtypes of the Activity are needed to suit the scheduling purpose, such as a kind of summary activity that represents a group of activities with the same characteristics so that the tasks of scheduling can become quicker and easier. In fact, most scheduling software packages support one or more types of activity grouping mechanisms (e.g. the Summary_Task and MasterProject in MS Project, or a typical REPCON activity in REPCON).
- Although the ACRM stress the relationship between Activity and Product, the test clearly indicates that in order to fully take advantage of this approach, more detailed implementation of the Product_View_Model is essential. For instance, when the learning curve effect needs to be considered in construction planning, the products associated with a group of a same type of activities must also be of the same type, e.g., similar types of columns. In this case, some subtypes of the Product are useful to classify the building components. In other words, good process models must be based on good product models with each having its own focus. While StartPlan uses the Process_View_Model because it deals with construction activity plans, other applications may apply the Product_View_Model. Nevertheless, an AEC central database must be based on the combination of both models.

CONCLUSION

A research approach to help develop standard AEC information models has been presented. The objective was to develop a set of AEC core models based on current work in this area, and to use these models in the development of a computer program with the use of an example project to test the central concepts and implementation capabilities of the models.

The paper first introduced the AEC Core Reference Models (ACRM) that are a set of AEC core models combining good features of many existing models developed by researchers in the area. It has also presented a computer program, StartPlan, that has been developed based on the ACRM and used to test the basic implementation capabilities of the models. StartPlan produces schedule templates as starting points for scheduling new construction projects. An example project used in StartPlan for the test was then described. Finally, the paper outlined the results from the test that are believed to be significant and useful to the future development of standard AEC information models.

Acknowledgments: We gratefully acknowledge financial support for this work by the Natural Sciences and Engineering Research Council of Canada.

REFERENCES

- Björk (92), "A Unified Approach for Modeling construction Information". Building and Environment, Vol. 27, No. 2, 173-194.
- Froese (92), "Integrating Project Management Software through Object-Oriented Project Models". Ph.D. Thesis, Dept. of Civil Eng., Stanford University.
- Froese (93a), "Transcripts of IRMA-tica'93, An International E-Mail Conference on the IRMA Model: An Information Reference Model for AEC". available via anonymous ftp from irma.js.jhu.edu.
- Froese (93b), "Project Modeling and Data Standards for AEC", Proceedings of the Fifth International Conference on Computing in Civil Engineering, L. Cohn, Ed. ASCE, Anaheim, California. Vol. 2, pp. 1730-1737.
- Froese (94), "Developments to the IRMA Model of AEC Projects", Proc. of the First Congress on Computing in Civil Engineering ASCE, Washington, D.C., June 1994.
- Froese and Yu (94), "StartPlan: Producing Schedule Templates Using IRMA", Proc. of First Congress on Computing in Civil Eng., ASCE, Washington, D.C., June 1994.
- Froese (95a), "Models of Construction Process Information". the Second ASCE Congress for Computing in Civil Engineering, Atlanta, USA, June 5-7, 1995.
- Froese (95b), "Core Process Models And Application Areas in Construction". "Modeling of Buildings through their Life-cycle," International workshop organized by the Working Commission on "Information Technology in Construction" (W78) and the Task Group on "Computer Representation of Design Standards and Building Codes" (TG10) of the International Council for Building Research Studies and Documentation (CIB), Stanford University, August 21-23, 1995.
- Froese, Yu and Shahid (96), "Project Modeling in Construction Applications", to appear in Third ASCE Congress for Computing in Civil Eng., Anaheim, CA, June 1996.
- Gielingh and Suhm (92), "IMPPACT Reference Model: An Approach for Integrated Product and Process Modeling of Discrete Parts Manufacturing" Springer - Verlag.
- ICON (94), "Integration Of Construction Information (ICON)", Integrated Databases for the Design, Procurement and Management of Construction, Dept. of Surveying & Information Technology Institute, University of Salford.
- Luiten et al. (93), "An Information Reference Model for Architecture, Engineering, And Construction". Proc. of First International Conference on the Mgt. of Information Technology for Construction, K. Mathur, M. Betts, and K Tham, Eds. pp. 391-406.
- ISO (94a), Building Construction Core Model, Project Proposal. ISO TC184/SC4/WG3 document N341, October 1994
- Luiten (94), "Computer Aided Design For Construction in the Building Industry". Ph.D. Thesis, Fac. of Civil Eng., Delft University of Technology, The Netherlands.
- Reschke and Teijgler (94), "Generic Reference Model for Life Cycle Facility Management (a proposal)". ISO TC184/SC4/WG3 Document N351, Rev.1. 1994.
- Russell (95), "Repcon 2 Educational & Research Version User Manual Version 2.00", Construction Management Group, Dept. of Civil Eng., University of British Columbia, Canada, 1995.
- Russell and Lamman (96). B.C. Construction Roundtable Draft Guide to Happy Projects. Report, Dept. of Civil Eng., University of British Columbia.
- Tolman, Bakkeren and Böhms (94), "ATLAS LSE Project type Model, Generic part of the ATLAS LSE Project Reference Model" ESPRIT 7280 -- ATLAS, 1994.
- Yu (95), "Application of Project Information Models to Computer-integrated Construction", MASC thesis, Dept. of Civil Eng. University of British Columbia.

