Theme:

Title: # A New Formal and Analytical Approach to Modeling Engineering Project Information Processes

Author(s): Charles Eastman, Ghang Lee, Rafael Sacks

Institution(s): College of Architecture, Georgia Institute of Technology

E-mail(s): Chuck.eastman@arch.gatech.edu

Ghang.lee@arch.gatech.edu

Rafael.sacks@arch.gatech.edu

Abstract: *A current research project within the North American Precast Concrete Industry aims to integrate information both within precast producer companies and between the companies and their suppliers, consultants, contractors, and clients. The first step was to undertake a process modeling study of the activities performed within each consortium company, so as to form the basis for the software specification and later data model.*

Existing process modeling methods and tools were considered . They do not support:

- *extraction of information used in the activities,*
- *analytical validation of the process model and its information flows,*
- *comparison of models collected from separate companies across an industry sector,*
- *effective use of process model information in deriving a product model.*

To these ends, we developed an analytic modeling methodology, and implemented a new tool for its use.

Keywords: *Process Model, Precast Concrete, Information*

Introduction

Process models are of increasing importance for studying operations within organizations, with the general broad purpose of planning, re-engineering, automating or augmenting them. Within this broad potential scope, the methods of defining process models and the model syntax can be highly varied.

The authors are the technical advisors of a consortium of precast concrete producers. We were hired to lead them in two related information technology initiatives (Eastman et al., 2001):

(1) specification of new software applications for precast concrete project-level design and engineering, supporting all production and downstream activities;

(2) development of a product data model that integrated the new software applications with enterprise-level IT, including plant production scheduling, construction planning, accounting, procurement and resource planning.

A process modeling exercise was recognized by the consortium and ourselves as an important initial undertaking, to plan the scope of the two above initiatives and to gather appropriate background information. The purposes of our process model undertaking had three components.

a. For each individual company, our goals were to encourage the company to develop its own process model, for planning the re-engineering of their organization in anticipation of the new IT technologies that will be introduced. The companies had quite different business models; some companies were strictly precast component subcontractors, others were design-build total package suppliers; some managed a single production facility while others integrated engineering and design across a set of regional production facilities. Some had large engineering shops, while others relied almost completely on contracted engineering services. Thus we expected the process models of their revised organizations to be highly varied.

b. For the engineering software specification effort, our objectives were to define the detail design and engineering processes in sufficient detail to derive an unambiguous software specification.

c. For the product modeling effort, our intention was to define the scope and information needed to derive an effective product model.

Process modeling involves both a procedural methodology and reliance on some language syntax and tool that defines the semantics that can be encoded for data collection and possibly analysis of the data later. Because process re-engineering is an important activity for improving company performance, a number of process modeling tools have been developed, reflecting various objectives.

Process modeling is the standard initial step to develop a product model in the ISO-STEP methodology (ISO TC184-P5, 1992), using the IDEF0 process modeling language (IDEF0, 1981). The accepted focus of process modeling within a product modeling development effort is to identify the generic, common processes followed by all companies. Because such models are generic to the industry described, they mostly address commonalities and are quite high-level; they do not include the contents of information flows between processes, and they seldom include important detail issues that usually vary across companies, such as when quality assurance processes or planning activities are undertaken. This process modeling methodology relies on social processes involving committee work and negotiation to derive the generic process model. Their results are influenced by the personalities of participants and the political power of participating organizations.

The IDEF0 models produced in this way are constructed to be evaluated, corrected, and interpreted by people. As a result, there are few means or checks to validate the resulting models. They provide at best a documentation of the context the committee members will rely on in developing a product model. Also, there is no mechanism to extract information from the process models that can be applied to the definition of a product model.

Other process modeling languages and tools reflect other objectives. A few depict information flows between activities. The UML Activity Diagram is “essentially a flowchart, showing flow of control from activity to activity” (Booch et al. 1999). Its use and functions are similar to those of traditional workflow diagrams (ANSI - IEEE standard 5807-1985, 1991). It does not support needed functions to collect information for product or data modeling. The Data Flow Diagram (DFD) (Osborne and Nakamura 2000) method is structured for describing information flows, similar to our needs. However, they have little emphasis on capturing actual sequencing. Also, in DFDs, information transactions are often described in aggregate form, such as a report or a specification. However, in the intra-industry product modeling effort we describe, no two companies use the same form of reports or specifications. Also, in this project, the amount of information exchanged between Activities is too large to be noted down beside a graphic flow. Therefore, a tool that can capture and manage information items individually and dynamically was required.

We were generally dissatisfied with the conventional process modeling methods used for product modeling. Our concerns involved both the social method for defining the process model and the lack of means to verify the accuracy or validity of the models generated. We were dissatisfied with the lack of emphasis on participating companies developing process models of their organization, for planning how to integrate the new IT technology. We were also dissatisfied with the low-level of usefulness of the information gained in process modeling for later specifying the product model.

Reflecting the context of our process modeling activities, our goals for process modeling were grouped into three areas. For the participating precast concrete companies, the goals of process modeling were:

- allow each participating company to easily define a process model of their current and/or proposed operations. These corporate process models should provide good means to assess the integration of new IT tools into their organization;
- provide a common framework, syntax and taxonomy of terms, so that the company process models can be compared and integrated;
- provide strong embedded rule checking and information tracking within the tool, supporting analytical validation of the process model and its information flows.

For the software specification activities, our goals were:

- clearly identify a set of activities and their information use and generation, supporting precast product design and engineering;
- gain documentation of the downstream uses of the design and engineering information;
- The software specification was intended to be functional, describing detail capabilities of the needed software; it was not intended to be architectural, defining data objects or structures; this was to allow acceptance of software products that were built on top of a range of different CAD platforms.

For the product model definition task, our goals were to:

- provide means to extract information from the multiple company process models, allowing assessment of a product model to support the different business models involved;
- collect information flow contents and properties used by the different activities in a manner allowing partial derivation of a product mode.

This paper describes how these goals were realized in the Georgia Tech Process for Product Modeling Tool (GT-PPM). The rest of the paper is organized as follows. Section 2 provides an overview of GT-PPM. Section 3 gives an overview of the rule checking and information tracking capabilities embedded in GT-PPM. Section 4 outlines how the process modeling tool is being used to derive a product model.

The GT- PPM Process Modeling Tool

Our tool was developed as a set of Visual Basic Applications on top of Vision 2000®. We defined a set of Shapes with new properties and associated macros that were incorporated into Visio's menu, called a Stencil. The shapes support activities of two kinds: aggregate and detailed. Aggregate Activities are collections of Aggregate and/or Detailed Activities; every Detailed Activity is associated with an Aggregate Activity. Activities are also distinguished as Internal or External. External Activities are those outside of the scope of the process model, but that interact (through the exchange of information) with Internal Activities. Thus it is assumed that External Activities are only partially defined, while Internal Activities are fully defined. Internal Activities are the main focus of the model.

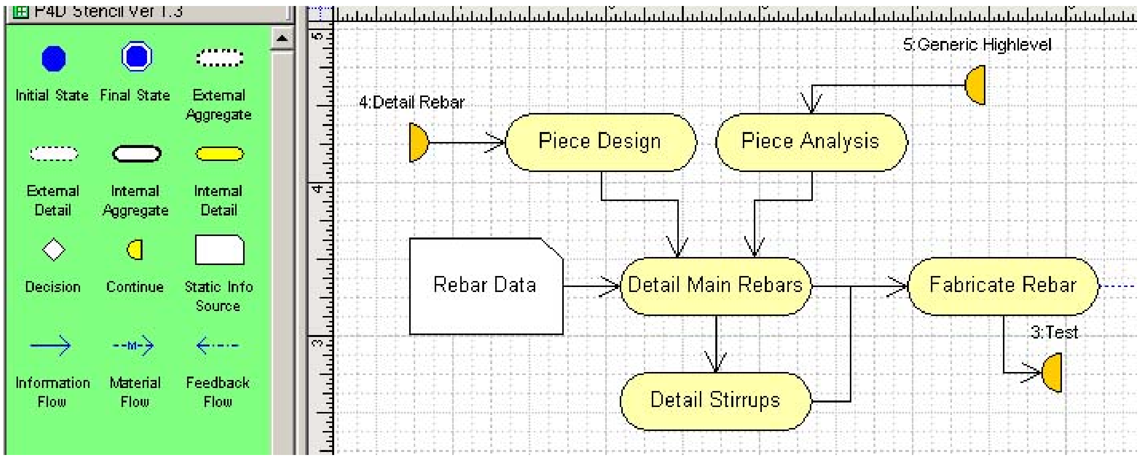


Figure One: The GT-PPM Tool's stencil and a portion of a process model.

Each Detailed Activity has associated sets of information categories that identify the information generated or used by the Activity. The information categories identified in these sets are presented to modelers in a menu structure composed from a data dictionary, pre-defined by the precast consortium, of information used in their industry. The information categories identify precast pieces and their features, assemblies and systems, structures, sites and projects. For each entity, they identify relevant properties and attributes, including the information used to derive these attributes. For each piece, the data dictionary identifies components such as reinforcing, material, plates, finishes, embeds and block-outs. It also includes information about molds and other production equipment. The Visio shapes in the stencil and a segment of a model are shown in Figure One.

Information flows between detailed Activities identify where the information used by an Activity comes from. In addition, Material Flows are also shown. They depict information in embodied form, such as piece measurements. Material flows also provide for identification of some other dependencies, beyond information dependencies. Static libraries are provided, that can be used to supply information such as code criteria, PCI Design Standards, Standard precast piece definitions, standard products, etc. (It is assumed that update of static libraries are outside the scope of the modeling process.)

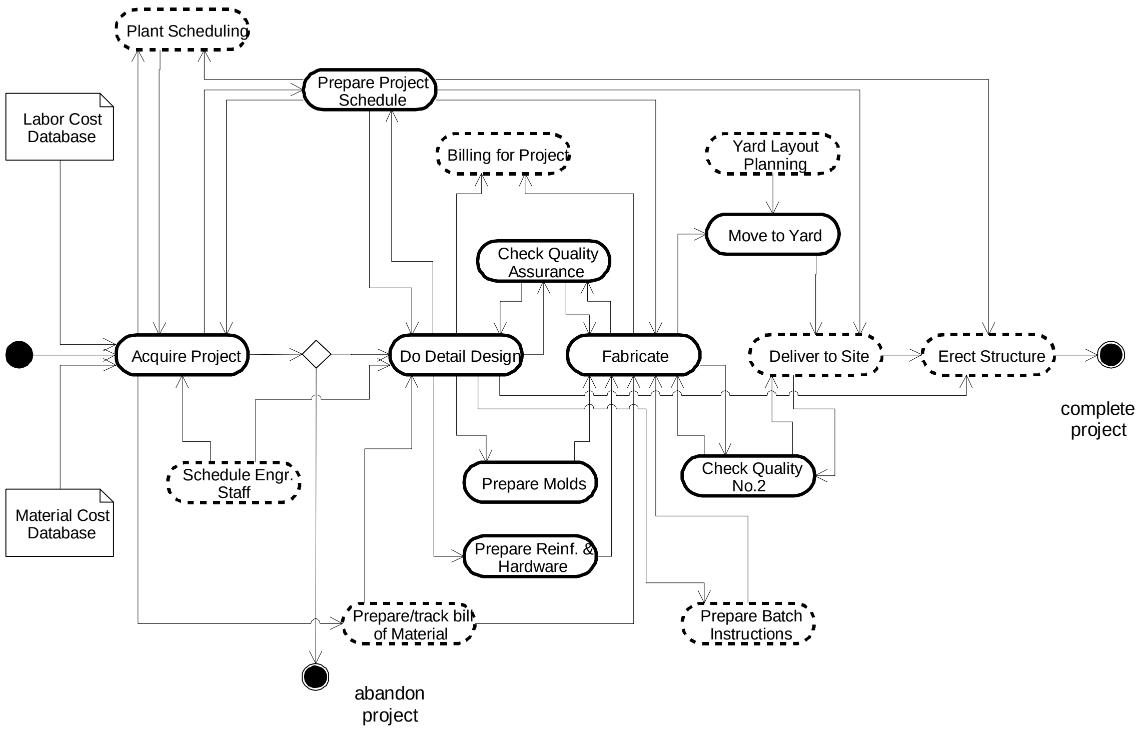


Figure Two: The initially defined top-level generic process agreed to by the precast consortium.

The process models to be developed were expected to be quite large, so 'Continue' Shapes were provided for linking the flows on one Visio page with those on another. Since we were interested in tracking information flows, Continue Shapes structurally linked an output flow of a Detailed Activity on one page with an input flow of a Detailed Activity on another page. They could also be used to denote where detailed flows were linked to an Aggregate Activity, but these were not traceable.

In order to provide a level of commonality across Activities, the precast consortium agreed to a common, generic top-level process. The purpose of this common definition was to allow comparison across company process models at some level of aggregation, so that commonalities and differences could be identified. The top-level definition of common Activities is shown in Figure Two. The software application addressed a single Activity, "Do Detailed Design". Later, a second level of common Activities was agreed upon, based on analysis of preliminary versions of some company process models. This articulated further the definition of "Do Detailed Design" into three more detailed activities: "Lay Out Building Assembly", "Detail Building Assembly", and "Detail Piece Designs".

The reason for needing to gain agreement for some generic top-level Activities is that these provide a set of common anchors that all process models reference. The second level definition also served another purpose; it provided the top-level of a more detailed process specification used in the software application specification.

The models generated by the precast producers ranged from simple ones of fifty or so detailed Activities to others involving design-build processes that consisted of over 500 detailed Activities.

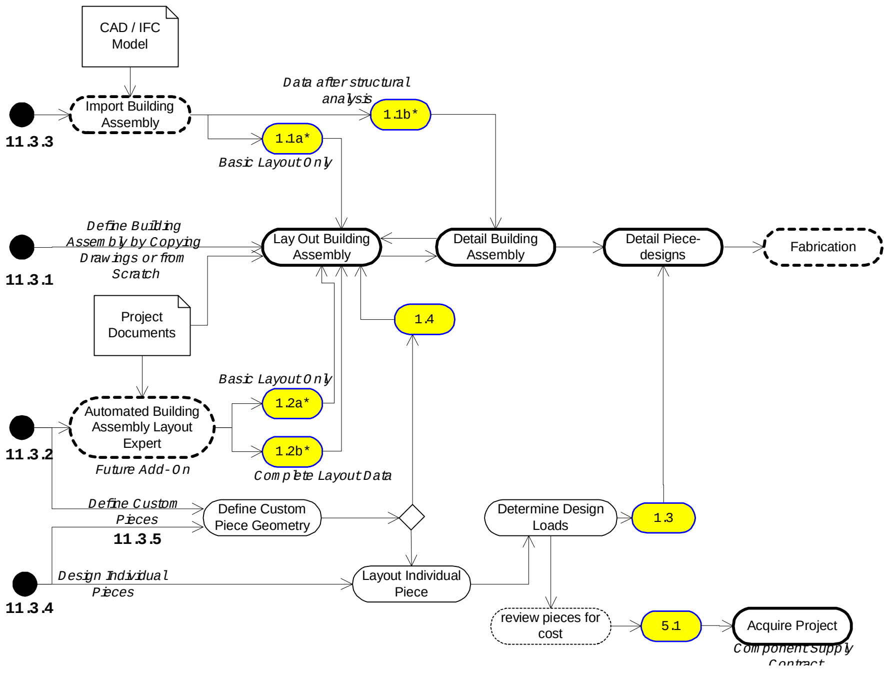


Figure Three: the additional detail of common high- level Activities. These Activities elaborate "Do Detail Design".

Information Flows, Rules and Tracking

It was reported that laying out Activities using an early version of GT-PPM was tedious but straightforward. The difficulty was in identifying the information the Activities generated or used. We developed several methods of displaying and coding the information categories. It was not adequate to know, for example, that the *piece_geometry* information was used; the context of use, whether the information was part of the information embedded in the structure, or of a standard library piece, or a custom-defined piece, was also important for determining the precast model semantics that would eventually be embedded in the product model. Combining context and information categories resulted in several thousand possible categories. These were defined in the context of their definition, for example, *piece_finish_material_standard_code_title*

or

structure_erection_tensioning_element_cable_geometry_partial_parametric_shape.

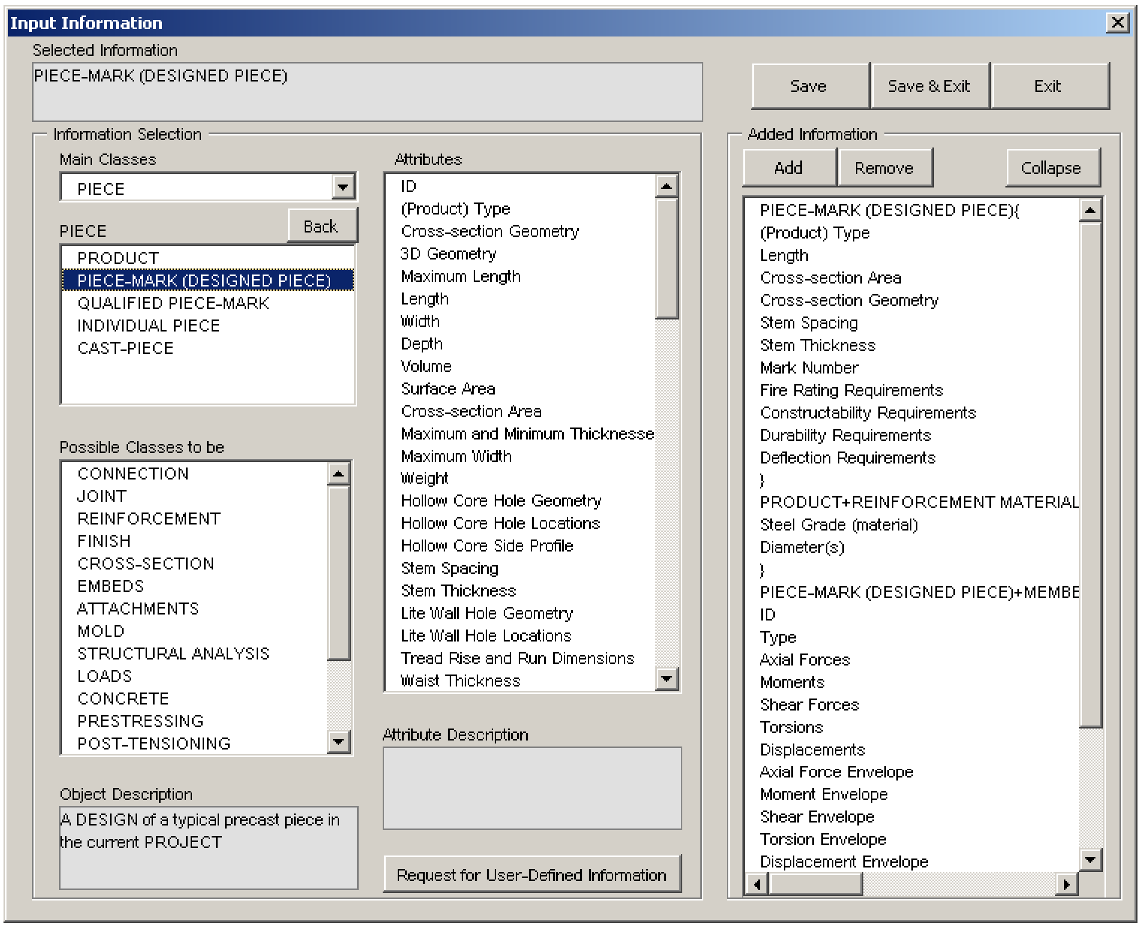
Such classifications were defined in outline format, moving from project scope to component, then to detail or feature, and finally to properties.

After several iterations, the context was saved using an outline system and sorting of categories, so that all categories at a given level were identified together under their shared top-level category, making search for information much easier. An example set of selected information categories is shown in Figure Four.

Main classes consisting of Project, Site, Structure, Assembly and Piece are selectable in the upper left of the scrollable menu. Subtypes are shown in the middle-left window, with subcategories below it, possibly several levels deep. Bottom-level attributes are dynamically loaded in the middle window. The current nesting of categories is shown in the top-left window. Already selected information categories are shown on the right, in nest parentheses format.

Using these categories, users identify the information used in each Activity. Information in an Activity is categorized input and output, and output is sub-categorized into referenced, changed, and newly generated information. Information from preceding Activities is called *Available information* and information necessary for following Activities is called *Required information*. The relations between these information categories are logically defined. Based on the logic, consistency of information can be checked dynamically (Lee et al., 2002). In addition to menus for defining the information used in an

Activity in isolation, an expanded menu allowed them to be displayed in context. The expanded display (for the 'Detail Main Rebars' activity from figure one) is shown in Figure Five. On the input side, it shows the information available as possible input, by compiling the information flowing through the arrows from previous Activities. It also shows the information utilized and information identified as required but unavailable. On the output side, it shows the information utilized by succeeding Activities from this Activity and the information needed by them, but unavailable. This context provides a rich display of the flow of information. Using the expanded information flow menu with careful definition of information flows, the information flow structure can be easily defined.


Input Information

Selected Information: PIECE-MARK (DESIGNED PIECE)

Save | Save & Exit | Exit

Information Selection

Main Classes: PIECE

PIECE (Back)
- PRODUCT
- PIECE-MARK (DESIGNED PIECE)
- QUALIFIED PIECE-MARK
- INDIVIDUAL PIECE
- CAST-PIECE

Possible Classes to be
- CONNECTION
- JOINT
- REINFORCEMENT
- FINISH
- CROSS-SECTION
- EMBEDS
- ATTACHMENTS
- MOLD
- STRUCTURAL ANALYSIS
- LOADS
- CONCRETE
- PRESTRESSING
- POST-TENSIONING

Object Description: A DESIGN of a typical precast piece in the current PROJECT

Attributes
- ID
- (Product) Type
- Cross-section Geometry
- 3D Geometry
- Maximum Length
- Length
- Width
- Depth
- Volume
- Surface Area
- Cross-section Area
- Maximum and Minimum Thicknesse
- Maximum Width
- Weight
- Hollow Core Hole Geometry
- Hollow Core Hole Locations
- Hollow Core Side Profile
- Stem Spacing
- Stem Thickness
- Lite Wall Hole Geometry
- Lite Wall Hole Locations
- Tread Rise and Run Dimensions
- Waist Thickness

Attribute Description

Request for User-Defined Information

Added Information (Add | Remove | Collapse)

```
PIECE-MARK (DESIGNED PIECE){
(Product) Type
Length
Cross-section Area
Cross-section Geometry
Stem Spacing
Stem Thickness
Mark Number
Fire Rating Requirements
Constructability Requirements
Durability Requirements
Deflection Requirements
}
PRODUCT+REINFORCEMENT MATERIAL
Steel Grade (material)
Diameter(s)
}
PIECE-MARK (DESIGNED PIECE)+MEMBE
ID
Type
Axial Forces
Moments
Shear Forces
Torsions
Displacements
Axial Force Envelope
Moment Envelope
Shear Envelope
Torsion Envelope
Displacement Envelope
```

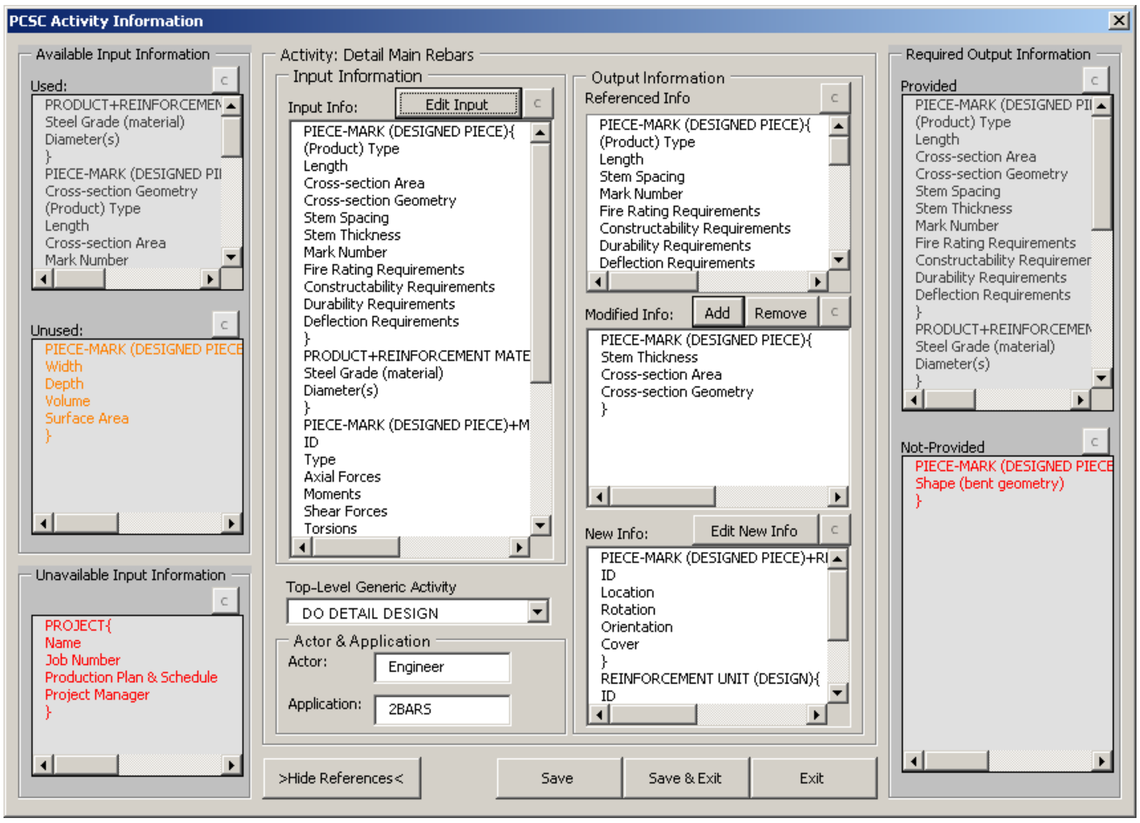
Figure Four: Information categories

Several details of information flows resulted in difficulties, based on the flow logic embedded in the model. Feedback and iteration processes, where later Activities pass back information to repeat an earlier Activity, can result in all the generated information for a series of Activities to be fed back into the beginning of the process, resulting in a high level of information flow redundancy.

In GT-PPM, feedback information flows are always connected to a Decision Shape. Success branches in a Decision Shape carry the full tracking of Information flows. Negative decisions, or those requiring iteration, by convention carry only procedural information (e.g. 'rejected', 'too low', etc.) noted on the arrow. Any information item stored in Activities can be graphically tracked. All Activity and Flow diagrams that include the targeted information item are dynamically colored in the diagrams.

Deriving a Product Model from a Process Model

We have just begun to develop methods for extracting data model information from the process models collected using GT-PPM. The data collected in a process model is structured in an Excel spreadsheet, ready for various analyses. The general strategy is to identify process model features that indicate the desired product model constructs to be used.



```
Available Input Information
Used:
PRODUCT+REINFORCEMEN
Steel Grade (material)
Diameter(s)
}
PIECE-MARK (DESIGNED PI
Cross-section Geometry
(Product) Type
Length
Cross-section Area
Mark Number

Unused:
PIECE-MARK (DESIGNED PIECE
Width
Depth
Volume
Surface Area
}

Unavailable Input Information
PROJECT{
Name
Job Number
Production Plan & Schedule
Project Manager
}

Activity: Detail Main Rebars
Input Information
Input Info:
PIECE-MARK (DESIGNED PIECE){
(Product) Type
Length
Cross-section Area
Cross-section Geometry
Stem Spacing
Stem Thickness
Mark Number
Fire Rating Requirements
Constructability Requirements
Durability Requirements
Deflection Requirements
}
PRODUCT+REINFORCEMENT MATE
Steel Grade (material)
Diameter(s)
}
PIECE-MARK (DESIGNED PIECE)+M
ID
Type
Axial Forces
Moments
Shear Forces
Torsions

Top-Level Generic Activity
DO DETAIL DESIGN
Actor & Application
Actor: Engineer
Application: 2BARS

Output Information
Referenced Info
PIECE-MARK (DESIGNED PIECE){
(Product) Type
Length
Stem Spacing
Mark Number
Fire Rating Requirements
Constructability Requirements
Durability Requirements
Deflection Requirements

Modified Info:
PIECE-MARK (DESIGNED PIECE){
Stem Thickness
Cross-section Area
Cross-section Geometry
}

New Info:
PIECE-MARK (DESIGNED PIECE)+RI
ID
Location
Rotation
Orientation
Cover
}
REINFORCEMENT UNIT (DESIGN){
ID

Required Output Information
Provided
PIECE-MARK (DESIGNED PI
(Product) Type
Length
Cross-section Area
Cross-section Geometry
Stem Spacing
Stem Thickness
Mark Number
Fire Rating Requirements
Constructability Requiremer
Durability Requirements
Deflection Requirements
}
PRODUCT+REINFORCEMEN
Steel Grade (material)
Diameter(s)
}

Not-Provided
PIECE-MARK (DESIGNED PIECE
Shape (bent geometry)
}
```

Figure Five: Expanded Information Menu with Input and Output Flows.

Some straightforward analyses include:

- For the Entities used in some Activity, identify their attributes (recorded in the menu shown in Figure Four, right).
- Identify all aggregation Entities and their parts; these can be defined in a process model as a chain of information items, as described in the section on Information Flows. Some aggregations imply relations between chained entities in the product model.
- Identify all the required Static Information Sources such as libraries; these are available as explicitly defined shapes; identify the entities carried in each library and their attributes, as defined above; these are standard “objects”, defined in the library, at the international, regional, industry, company, or supplier level.
- A single definition is often provided for a set of objects, for example when a standard piece is used for bidding purposes. Identify when such a set is defined by a single object, then decomposed into multiple sets; associate any change in the attributes carried by the two classes; such transitions indicate different classes of part and their incremental refinement during design or production. An example is when there is an object *piece*, a piece can be subcategorized into a *design piece* or a *cast piece* by the stage of a process.

These methods of derivation are still being developed and the work is ongoing. We believe that such methods provide a logical start in the development of a product model. However, even using multiple corporate models will not address the various special cases that arise, or special features or attributes, that may or may not be in common use. Such special cases will have to be sought out and added incrementally (by hand) (Eastman et al, 2002).

Integrating a Single Product Model Reflecting Multiple Company Process Models

The above procedures begin to allow the partial derivation of a product model from a process model. However, there will be many different process models generated, reflecting the information requirements of different company business models. How can we generate a single product model able to support all of the process models?

The process described above will generate multiple sets of constructs, identified by the process model features. We have not yet determined how complete a set of features we will be able to derive. However, the various constructs then need to be integrated. We believe this can be done, using various composition rules. Since many of the constructs from different companies will be highly overlapping and redundant, they will result in a few attributes being added, or a new relationship identified. The resulting model, at least initially, will be a union of the mappings of the individual process models. In the relational data model theory, there are strong rules of normalization, defining how to restructure an SQL model to satisfy various objectives. Normalization is realized by various analyses of the *functional dependencies* between attributes (Chap. 14-15, Elmasri and Navathe, 2000). We believe it is possible to develop such normalization rules for constructing an Express model from a *universal set* of collected information.

Summary

Process modeling is potentially a very powerful tool in aiding the re-engineering of construction organizations. To date, the potential contribution of process modeling has been limited by the tools and procedures available. In the Precast Concrete Software Consortium project reported on here, in contrast to the ISO STEP approach of forming one generic process model as a guide to product modeling, we use a generic model as a top-level basis on which industry participants develop detailed models of their business processes. We have also explored ways of making the information collected in process modeling more precise, and developed analytical methods for using the information collected.

Note: This work was funded in part by the North American Precast Concrete Software Consortium (PCSC – http://dcom.arch.gatech.edu/pci). The consortium's sponsorship, and the intensive effort invested in the process models developed by the representatives of the member companies, is gratefully acknowledged.

References

1. ANSI (1991). American National Standards for Information Processing **-** Documentation Symbols and Conventions for Data, Program and System Flowcharts, Program Network Charts and System Resources Charts. New York, American National Standards Institute.
2. Booch, G., Rumbaugh, J. and Jacobson, I.. (1999). The Unified Modeling Language: User Guide. Reading, MA, Addison Wesley Longman, Inc.
3. Eastman, C.M., Sacks, R., and Lee, G. (2001). "Software Specification for a Precast Concrete Design and Engineering Software Platform", *PCSC Research Report*, Georgia Institute of Technology, Atlanta GA USA.
4. Eastman, C.M., Lee, G., and Sacks, R. (2002). "Deriving a Product Model from Process Models", *Concurrent Engineering 2002*, Cranfield U. U.K., July 2002.
5. IDEFØ, (1981). "ICAM Architecture Part II, Vol. 5," Information Modeling Manual (IDEFØ), Report number AFWAL-TR-81-4023, Vol. 5 Mantech Technology Transfer, Center WL/MTX.
6. ISO TC184/SC4/WG4 N34 P5, (1992). Guidelines for the Development and Approval of STEP Application Protocols.
7. Lee, G., Sacks, R., and Eastman, C.M. (2002). "A Process Modeling Tool for Product Modeling: GT-PPM", *Working Paper*, College of Architecture, Georgia Institute of Technology, Atlanta GA, USA.
8. Elmasri, R., and Navathe, S.B. (2000). Fundamentals of database systems, 3rd Ed., Addison-Wesley, Reading, Mass.
9. Osborne, L. N. and M. Nakamura (2000). Systems analysis for librarians and information professions, Libraries Unlimited, Englewood, Colorado.