Design Of Parametric Modelling Systems

R. N. Ibrahim, Dept. of Mechanical Engineering, Monash University
S. J. Boardman, Dept. of Mechanical Engineering, Monash University

Abstract

This paper describes a parametric system for Computer Aided Manufacturing (CAM). Existing parametric systems enable the development of simple Graphical User Interfaces (GUIs) for generation of parametric models. They do so at the expense of model generation flexibility, ease of entity parameter modification and non-declaration of entity relationships. Functional modelling (FM) allows unrestricted modelling by developing models using mathematical functions. The functions can directly refer to any point or attribute of any entity in the model. Relationships are automatically defined by dependent point structures and entity mathematical function declarations, eliminating the need for additional relationship. The paper also details the structure of entity relationships in a parametric system. In addition the structure of Multilayered Entity Relationship Matrices (MERM) which ensure relationship integrity and a method for developing parametric systems based on the MERM structure are defined. This paper details the theoretical system design work conducted during development of the Caddsman Parametric Modelling System (PMS).

Introduction

Parametric Modelling (PM) is the controlling, modifying or manipulation of graphical entities using mathematical variables (parameters). In PM, mathematical variables describe the attributes of an entity rather then a numeric value. By describing CAD models using parameters, the user is able to adjust or update the attributes of the entity by changing a parameter. Vector graphical entities, such as lines, circles and squares are usually designed to describe real world objects. By making entities parametric, the model may be tailored to meet a design for multiple situations and/or design requirements without the need to generate an entirely new model.

The concept of PM lies at the root of CAD systems. CAD systems use GUIs such as computer monitors, plotters and printers to display visual representations of mathematical defined vector entities stored on computer. GUI tools such as extend, trim and stretch modify an entity's parameters by redefining the value inside the parameter using a graphical interpreter. The user has no control over the interpretation process.

The objective of CAD based PM is to allow the user to modify the parameters by directly inputting a numerical value. The mathematical definitions for altering individual parameter values and the supporting system structure for a functional modelling system is discussed in [1]. This paper contains the conceptual design information for developing the structure and dynamics of a successful Parametric Modelling System (PMS) based on the FM process and the basic processing algorithm used to design the Caddsman Parametric Modeller program. With the introduction of recent Computer Aided Process Planning (CAPP) developments, the need for dynamic modelling and the ability to declare additional infomation directly on the model have become major issues [6 - 8 and 10]. This paper attempts to provide viable solutions to these problems.

Functional Modelling (FM)

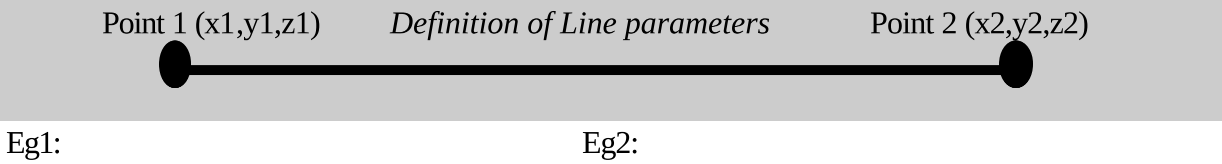
FM is the direct and accessible declaration of entity attributes or points using mathematical formulae in the form of model referencing functions. In order to ensure maximum flexibility of the modelling system, FM must allow direct reference access within a function to any point or attribute contained in the model. The generated function must be able to include more than one reference parameter (variable) from the model and allow inclusion of numeric values.

The Parametric Modeller program derived from the FM system also allows inclusion and use of any Excel worksheet function (including referencing of entity parameters inside Excel functions) that returns a single value, and the combination of multiple Excel functions, points or attributes and numeric values when forming the functions for defining the model. The advantages of defining a model by using referencing functions are the automatic generation of full relationship structures[1] and dynamic model structures, the ability to mathematically define object's structures and movements, and the ability to include any valid Excel mathematical functions when defining the model.

Such advantages derive from the FM capacity to store model information as dynamic functions. When called these functions are calculated, generating graphical entities for the CAM system. All the function information is then stored, allowing the direct modification of the function at a later time. This method is unlike that of existing models which use compiled PMSs that query the user for data then generate their own internal bit functions. The user never has access to the function and is restricted to using/altering parameters which are included in the compiled files. The user cannot access other model points or attributes in the generated functions and the system cannot ensure unrestricted access or definition of all Entity Relationships.

Entity Parameters

While entity attributes are characteristics of an entity, entity parameters describe an entity's attributes. Those parameters however are not entity attributes. Several entity parameters may constitute a definition of an entity attribute. Figure 3.1 displays the definition of the parameters that constitute the position of the start and end points of a line. Cartesian co-ordinates for Point 1 and Point 2 define the start and end points of entity Line 1. With PM the user is able to directly define the values for these co-ordinates. By varying the position of the line, and subsequently, the values of the parameters, the user can redefine the length attribute of the entity Line 1.



Eg1:

$\text{Line1}:(x1=0, y1=0, z1=0)\,(x2=100, y2=100, z2=0)$

$$\text{VectorLength} = \sqrt{(x2-x1)^2+(y2-y1)^2+(z2-z1)^2} = \sqrt{100^2+100^2+0^2} = 141.42$$

Eg2:

$\text{Line1}:(x1=0, y1=0, z1=0)(x2=100, y2=100, z2=100)$

$$\text{VectorLength} = \sqrt{(x2-x1)^2+(y2-y1)^2+(z2-z1)^2} = \sqrt{100^2+100^2+100^2} = 173.21$$

Figure 3.1 - Different Co-ordinate sets describing a line, and the resultant vector lengths.

Figure 3.1 demonstrates the effect of increasing the z parameter in example 2 from 0 to 100. The alteration of the co-ordinate position increases the length attribute of Line 1 by 31.79 units. Other

---

[1] Refer Multilayered Entity Relationship Matrices (MERM)

entity attributes such as Line Colour, Style, Weight and Layer are single parameter attributes. Single parameter attributes are attributes, which have only a single variable describing them. The situation described in Figure 3.1 is that of a multiple parameter attribute. The length of Line 1 may be altered by changing any of the entity's co-ordinate position parameters. Redefinition of an entity's parameters can occur at any of the three information levels where entity parameters are stored. The information levels are: 1) Operating system, 2) Application, 3) Data File. To use FM as a viable parametric method, the data generation and modification should be conducted at the tertiary information level.

Entity Relationships

Entity relationships dictate the effects that an alteration to the parameter value of an entity will have on the attributes or position of all other entities defined in the model. Relationships define the nature of the model, the effects and scope that changes to parameter values will have on other entities in the model, and the possible model complexity which can be achieved using the PM system.

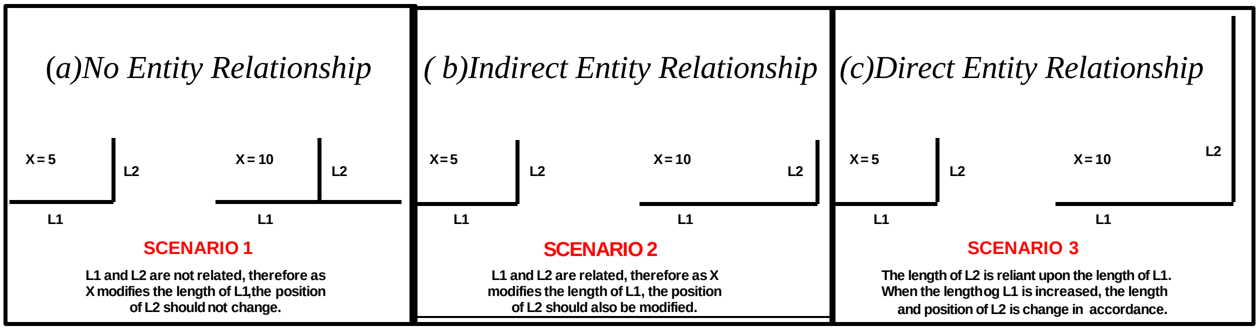


Figure 4.1 - Entity Relation Definitions

Each of the scenarios displayed in Figure 4.1 represents the relationship definition options. The three types of entity relationships possible are:

1) No Entity Relationship: Modification of the first entity's attribute does not affect any attribute or position point of the second entity in an entity pair[2] (Scenario 1).
2) Indirect Entity Relationship: Modification of the first entity's attributes results in a position change of the second entity in the entity pair (Scenario 2).
3) Direct Entity Relationship: Modification of the first entity's attribute changes an attribute of the second entity in the entity pair (Scenario 3).

Figures 4.1(a), (b) and (c) describe relationship types 1, 2 and 3 respectively. When determining entity relationships for a parametric system, the defining relationship should be considered as the highest in the relationship hierarchy that exists between the two entities. For example, in scenario three, Line L2's length attribute is reliant upon the length attribute of Line L1, creating a Direct Entity Relationship between the entities in terms of the length attribute. A second relationship exists in scenario 3. Line L2's position is also reliant upon the end point of Line L1 as is the length of Line L1. This situation creates an Indirect Entity Relationship between the two entities.

---

[2] Entity Pair: Any two entities contained in the model. It is essential that entity relationships can be, and are defined for all entity pairs.

When constructing relationship matrices, the definition of the relationship between Lines L1 and L2 should be considered a (3) Direct Entity Relationship as this is the highest defined relation between the pair.

Multilayered Entity Relationship Matrices (MERM)

The Multilayered Entity Relationship Matrix provides solutions to the problems of entity definition, solid modeling and ECV PM. MERM provides entity relationship definitions with the use of Dependent Point Variables (DPV). A Dependent Point Variables Matrix constitutes a collection of Dependent Points (DP). DPs define the position of a grouped set of entities defined as an object. Dependent Points anchor entities to themselves and define entity relationships between other entities and other objects. Unlike ECV PM, MERM allows parametric control over both related entities within an object and entities in other objects. Also unlike other parametric systems, entities can operate independent, semi-dependent or dependent upon their object.

The multilayered entity relationship matrix uses two methods to define relationships between all entities in the model. The first is Dependent Point Variables and Entity Point Variables matrix structure. The second is an overriding technique for directly referencing other entity attributes. Dependent Point relationships can only provide indirect entity relationships or no entity relationships between entities from two different objects. The overriding technique allows entity attributes to be modified by other entity's attributes, even entities not contained in the same object. This technique provides a direct accessing method for referencing any external attribute not contained in the current entity. If an external attribute is referenced, the value of the current parameter becomes dependent upon the value of the external parameter, therefore defining a direct relationship. If an external parameter is not referenced during the definition of the current parameter, then the entity relationship with all external entities reverts back to the DP relationship (indirect or no entity relationship). Figures 5.1(a) and (b) display an example of this concept. The position of L4 is reliant upon Dependent Point 1 (DP1).

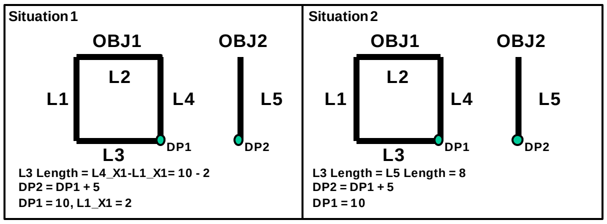


Figure 5.1(a) - Initial scenarios for 1) a relationship between L3 & L2; 2) a relation between L3 & L5

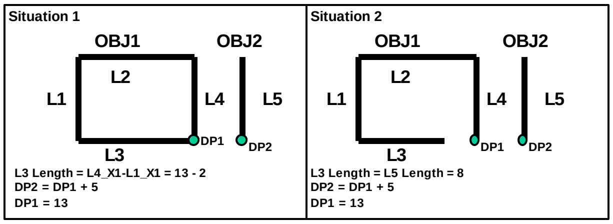


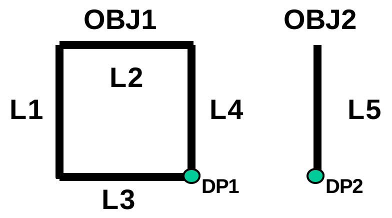
Figure 5.1(b) - Effects for each scenario of updating L2's length attribute

In Situation 1, the standard entity relationship exists, which may be defined using solid modeling techniques, etc. The length of L3 is dependent upon the position of L4 and L1. In Situation 2, L3's length is reliant upon the length of L5. The introduction of L5 as a parameter in the definition of L3's length has overridden the relationship, which existed in Situation 1. When the position of L4 is modified in Situation 1, the length of L3 increases from 8 to 11. In Situation 2, because the length of L3 is reliant upon the length of L5, not the position of L4, the length of L3 does not change. This method of overriding entity relationships is important to define before the matrix structure definition because it displays how the relationship problem can be solved, allowing every entity to have any other type of relationship with any other entity in the model.

Merm Matrix Structure

DPs are used to define relationships between groups of entities without the tedium of defining every entity relationship combination. The DP describes the Indirect position relationships, or lack of, between the entities contained in an object and entities contained in other objects. A direct relationship to an individual entity parameter will override the indirect relationship created by the DP (relationship hierarchy).

The relationship matrix structure occurs with at least two layers. The lowest or Primary Matrix contains the individual entity attribute relationships/information. This matrix is where DP relationships are overridden by the overriding technique of entity attribute referencing as described previously. The primary matrix will only contain more than one column if the object is not a basic entity type and internal relationships between 'sub-entities' are required. Primary matrices are defined by the method of entity definition by the CAD system. The Secondary Matrix defines entity relationships within the object. An example entity relationship matrix was developed for the objects displayed in Figure 6.1 and is displayed in Figure 6.2.



For OBJ1 (Dependant Variable = DP1)

Secondary Matrix 1:

$$\text{DP1} = \text{Length Relationship} \quad \overset{\text{Relationship}}{\begin{array}{c|cccc} & L1 & L2 & L3 & L4 \\ \hline L1 & 3 & 2 & 2 & 3 \\ L2 & 2 & 3 & 3 & 2 \\ L3 & 2 & 3 & 3 & 2 \\ L4 & 3 & 2 & 2 & 3 \end{array}}$$

For OBJ2 (Dependant Variable = DP2)

$$\text{DP2} = \text{Length Relationship} \quad \overset{\text{Relationship}}{\begin{array}{c|c} & L5 \\ \hline L5 & [3] \end{array}}$$

(Primary and Secondary Matrix 2)

Figure 6.1 - Model and Primary Entity Matrices

$$\text{Tertiary Matrix} = \begin{array}{cc} & \text{DP 2} \\ \text{DP1} & [2] \end{array}$$

$$\text{Resultant Secondary Matrix} = \quad \text{Length Relationship} \begin{array}{c|ccccc} & \text{L1} & \text{L2} & \text{L3} & \text{L4} & \text{L5} \\ \hline \text{L1} & 3 & 2 & 2 & 3 & 1 \\ \text{L2} & 2 & 3 & 3 & 2 & 2 \\ \text{L3} & 2 & 3 & 3 & 2 & 2 \\ \text{L4} & 3 & 2 & 2 & 3 & 1 \\ \text{L5} & 1 & 1 & 1 & 1 & 3 \end{array}$$

Figure 6.2 - Tertiary and Resultant Secondary Matrices

Where 1 = No Entity Relationship, 2 = Indirect Relationship, 3 = Direct Relationship
The Secondary Matrices displayed in Figure 6.1 consequently produce a Dependent Point Matrix - Tertiary Matrix. The Tertiary Matrix defines the relationship between the objects and consequently affects relationships between all entities. The Resultant Secondary Matrix is the effect modifications upon any entity will have on the other entities within the objects. The Resultant Secondary Matrix is not a combination of secondary matrices, it is the product of relationship calculations defined using the Relationship Formulae.

The Resultant Secondary Matrix defines all relationships for entities contained in two or more objects and can be redefined using the overriding technique defined previously. When defining parametric relationships, the relationship should default to the highest relationship defined in the parameter. For example, if the definition of the length of a line entity contains a complex function including a direct and indirect relationship with another entity, then the relationship should be described as a Direct Relationship. The entity relationships for any entity pair in the model may be calculated using the following formulae:

Definition of Relationship Forumlae:

$$X = \frac{\sum_{i=1}^{n+1} R_i}{n} \quad \begin{array}{l} \text{where } n = \text{number of entities between pair} \\ R_i = \text{entity pair relationship} \\ \text{and } i = \text{current entity pair} \end{array} \qquad \text{Therefore Relationship } (R) = \begin{cases} 1; & 0 \le X < 1 \\ 2; & 1 \le X < 2 \\ 3; & 2 \le X < 3 \end{cases}$$

Mathematical Definition of Parametric Attributes

The parametric attributes of all entities are described by numeric values. When CAD models are generated, usually these numeric values represent real life quantities. The result or this situation is that the virtual geometric space created in a CAD system may be equated to distances, for example a line entity 50 units long may equate to a corner of an object 50mm's long. The co-ordinate system and real life measurement of scale of the entities in the model should be specified on the model (ie. 50 units long may refer to a real length of 50 meters or 50 centimeters). An example of parameter points and attributes for a Line entity are displayed in Table 7.1.

Table 7.1 - Line Entity Definition

Entity Type:	**Line**
Attributes:	Start Point End Point Layer Style Colour Weight

In order to define the attributes of an entity, such as the ones displayed in Table 7.1, numeric values must be specified for the attributes' parameters. The parameters that define the starting point of a line entity are (x1, y1, z1). These parameters contain numeric values, which specify where the Line entity is to begin (in Cartesian Co-ordinates). To define the start position the user must define the parameters, for example

Ex 1. Start Point : x1 = 50, y1 = 80, z1 = 0

The end point is defined in the same manner

End Point : x2 = 100, y2 = 80, z2 = 0

The parameters that define the position of the entity may also be described using mathematical relationships and other parameters:

Ex 2. Start Point : x1 = (x2)/2, y1 = y2, z1 = z2;
End Point : x2 = 100, y2 = 80, z2 = 0

In Ex 1. the resultant starting and ending co-ordinates are : Start Point = (50, 80, 0), End Point = (100, 80, 0). Ex 2. results in exactly the same co-ordinates, Start Point = (50, 80, 0), End Point = (100, 80, 0) but the definition of the relationship is different. Ex. 2 uses a parametric relationship, where x1, y1 and z1's positions are reliant upon the position of x2, y2 and z2. If the position of x2, y2 or z2 is changed, the position of x1, y1 or z1 alters also. For example, in example 2, if y2 is changed from 80 to 100, because y1 = y2, and now y2 = 100, y1 = 100. In Ex 1., if y2 is changed to 100, y1 = 80. There is no parametric relationship between the two attributes.

Entity Relationships Using Parametric Definitions

The three entity relationship types discussed above are defined by the effect the modification of one entity will have on another. This method of definition may be described using mathematical definitions and parametric relationships. These entity relationships can all be described using mathematical relationships. In basic terms[3],

- Direct mathematical relationship is any parameter that is changed that does not produce a corresponding change in other parameters which describe the attribute. The change results from a mathematical relationship with another dependent parameter defined in a separate entity.
- Indirect mathematical relationship is when the dependent parameters describing an attribute have the same constant applied to each dependent parameter. The constant refers to a dependent variable defined in a separate entity
- No mathematical relationship describes the case where no relationship between the dependent parameter and the external entity. There is no direct or indirect inclusion of parameters from the second entity in the entity pair in the formulae for the first entity.

A detailed example of the mathematical relationships is defined in Figure 8.1.

---

[3] The term 'basic' is used in this situation because the ability to override relationships makes the possibilities for altering the situation virtually endless.

```
Entity Type: Line Entity
Entity Name: Line1
Atribute: Length
Length Dependant Parameters: Start Point, End Point
   * If y1= y2 and z1= z2
Dealing with x1 and x2
When External Entity = Line2

Situation 1: if
x1=0    &   x2= Line2.x1
Length = x2 - x1 = Line2.x1 - 0
Lenght = Line2.x1              ← Direct relationship - x1 does not contain Line2.x1 parameter,
                                  therefore the Line2.x1 term causes variation in the Length of line
                                  Line1 - it is not considered an attribute constant
Situation 2 : if
x1= Line2.x1 &  x2= Line2.x1 + 50
Length = x2 - x1=[Line2.x1 + 50] - Lin2.x1
Length = 50                    ← Indirect relationship - x1 and x2 both contain Line2.x1 parameter
                                  references, therefore the net length increase= 0
Situation 3: if
x1=0   &   x2=50
Length = x2 - x1 = 50 - 0
Length = 50                    ← No entity relationship - no reference is made to any external entity
                                  parameters
```

Figure 8.1 - Mathematical definition of Relationships

As described previously, Direct relationships alter the entity attribute, Indirect relationships alter the position of the entity and the No Entity relationship has no effect on the attribute. Indirect relationships require more effort to define than Direct relationships because all the parameters affecting the attribute must be changed to include the external parameter constant.

The parameter constant may be considered as a 'Virtual Constant'. The delta product of the attributes parameters = 0 and therefore, the position of the entity is modified but the numeric value of the entity remains the same. For the length attribute of a line, the effect of each relationship type may be considered as:

Direct Relationship → Increase/Decrease of Line length
Indirect Relationship → Movement of line, no length change
No Entity Relationship → No position change, No length change

With FM the model is almost entirely composed of referencing functions which can include reference to any parameter in the model. As a result, any relationship type can be generated between any pair of entities, making the model's structure and behavior possibilities only restricted by the geometric area defined by the CAD/CAM program and/or the size of calculations which can be computed by the spreadsheet. Additional restrictions may be placed on the model when using higher order GUI. These restrictions are eliminated by storing the definitions as mathematical functions and giving the user the ability to reference any entity parameter when defining a function, thereby maximizing the flexibility of the PMS.

Functional Modelling Algorithm

Using the MERM structure developed previously, an algorithm was created to produce a working Functional Modelling process. The algorithm utilized the calculation engine available in Microsoft Excel to conduct the complex matrix calculations for defining entities and increase the number of usable functions available when developing function declarations. Figure 9.1 displays the module structure of the processing algorithm available with the Caddsman Parametric Modeller system.

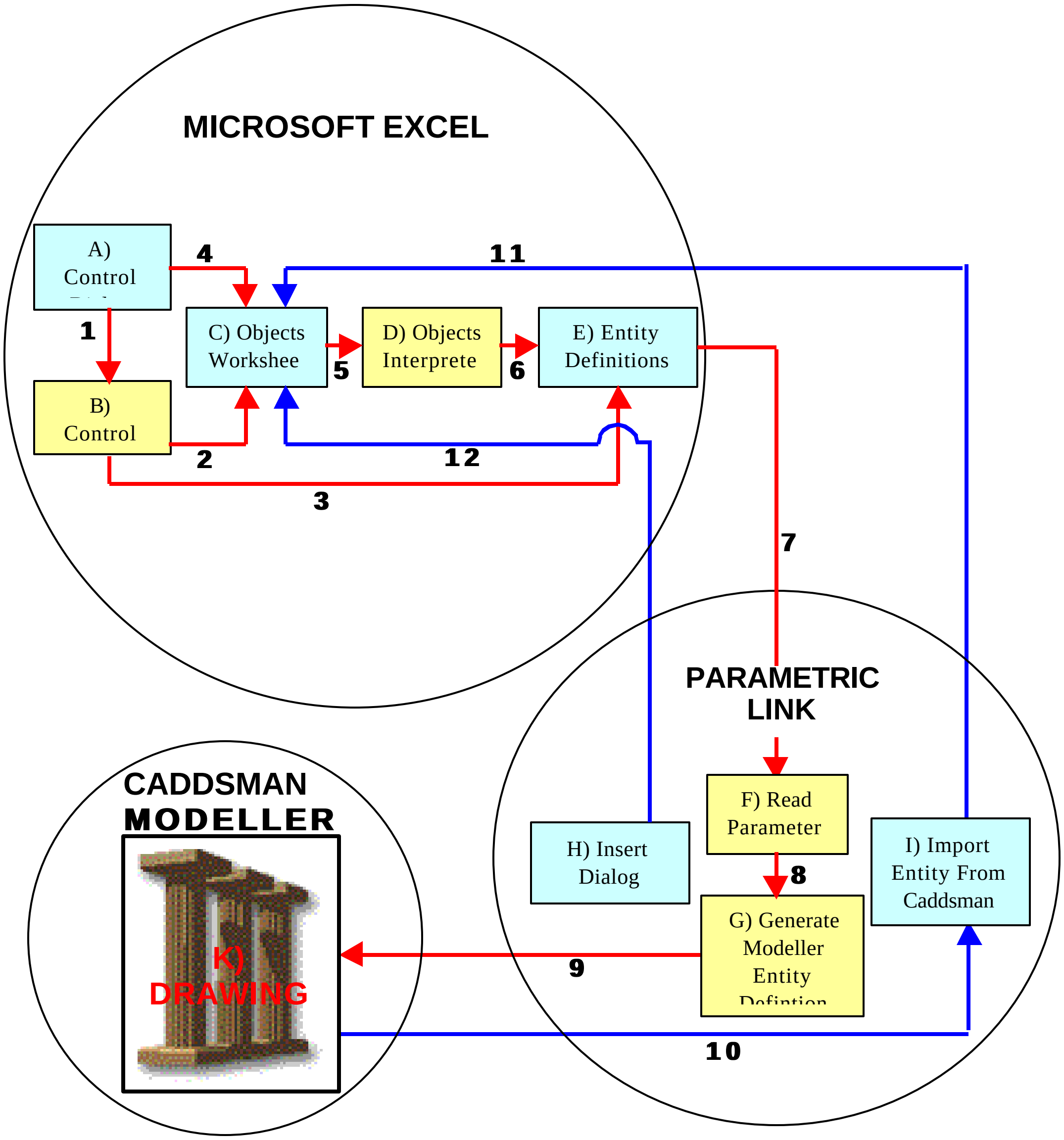


Figure 9.1 – Functional Modelling System

The following description of the algorithm displayed in Figure 9.1 uses reference to the bracketed letters {A…I} and numbering {1…12} to refer to aspects of the algorithm. A referring letter defines a component of the algorithm and a number refers to the information transfer between components.

Control dialogs {A} and control macros {B} provide a method for developing high end interfaces for general users to develop a generic model. The values in the dialog generally refer to central model aspects or animation settings. The information from the dialog can be interpreted by the macros {1,2} and then included on the object sheet {C}, interpreted by the macros and formatted directly {3} into parameter declaration then output to the Entity Definitions Worksheet {E} or placed directly {4} from the dialog box in to the object sheet {C}.

The object worksheet {C} contains the function declarations, the object structures and movement declarations for the model. All relevant information is declared on this sheet or directly to the entity definitions worksheet {E}. The interpreter {D} provides a calculation engine for the function declarations on the objects worksheet and is constructed from a generic excel Visual Basic for Applications (VBA) macro. The information is read by the interpreter {5} from the object worksheet, calculated on a dummy worksheet using an interpretation process [2] and the resultant function calculations are fed {6}to {E} for interpretation by the parametric link. [2] details the interpretation process and the format of object sheets, functions and component structures.

The parametric link provides a Component Object Module (COM) engine for reading entity parameters {7} from the Excel object sheet and creating the declarations in a model {9}. The interpretation engine which reads the parameter values {F} and generates the necessary entity declarations {G} cycles through each entity creation for each frame of the animation. The resultant entities are created by the modelling program and included in the model. Methods for defining functions {I} and importing entities {H} into an object worksheet {C} were also included in the parametric link. For information regarding these processes refer to [2].

Using the algorithm described above, the MERM structure of relationships was ensured due to the matrix structure of the calculations by the Excel engine. The program also allowed the inclusion of functions when developing models (FM) including most Excel working functions and the ability to define complex engineering calculations directly into a CAD model.

Conclusion

The ‘definitions of entities’ require the definition of large numbers of parameters. Graphical manipulation tools allow users to redefine entity attributes rapidly and with a minimum of effort. To reduce effort, the user surrenders control over direct manipulation of the entity parameters using mathematical manipulation. The objective of a successful parametric modeling system is to retain both. FM allows the automatic generation of relationship structures and ensures relationship integrity. FM also allows referencing to any point contained on the model and use of any function accepted by the calculation engine to be used when manipulating entities. In this way, these functions help to maximize the flexibility of the PMS. The resultant algorithm derived from the MERM structure offers unrestricted parameter manipulation, addition of functions to a model and the introduction of direct modelling using engineering calculations.

Reference

1. R. N. Ibrahim and S. J. Boardman. Design of Parametric Systems Using Functional Modelling. *World Manufacturing Congress* (1999).

2. R. N. Ibrahim and S. J. Boardman. Development of Parametric Design Systems using a Functional Modelling algorithm. *World Manufacturing Congress* (1999).

3. ISR Group. *Caddsman Modeller; Application Programming Language Reference Manual,* ISR Group Limited (1993).

4. D. J. Kruglinski. *Inside Visual C++: 4th Edition*, 4th edn. David J. Kruglinski, (1997).

5. G. James. *Advanced Modern Engineering Mathematics*, Addison-Wesley Publishing Company Inc. (1993).

6. Microsoft Corporation. *Microsoft Excel Help Topics,* Microsoft Corporation (1985-1996).

7. M. Kalta and B. J. Davies. Converting 80-Character ASCII IGES Sequential Files into More Conveniently Accessible Direct-Access Files. *Int J Adv Manuf Technol* 8:129-144(1993)

8. Li-Hong Qiao, Chun Zhang, Thu-Hua Liu, H. P. Ben Wang and G.W. Fischer. A PDES/STEP-based product data preparation procedure for computer-aided process planning. *Computers in Industry 21*, 11-22 (1993)

9. M. Kalta and B. J. Davies. CADEXCAP: Integration of 2D CAD Models of Turned Componenets with CAPP, *Int J Adv Manuf Technol* 8:145-159(1993)

10. M. Kalta and B. J. Davies. Product Representation for an Expert Process Planning System for Rotational Components, *Int J Adv Manuf Technol* 9:180-187(1994)

11. M. Kalta and B. J. Davies. Guidelines for Building 2D CAD Models of Turned Components in CAD-CAPP Integration, *Int J Adv Manuf Technol* 8:285-296(1993)

Title: Design of Parametric Modelling Systems

S. J. Boardman, Dept. of Mechanical Engineering, Monash University
R. N. Ibrahim, Dept. of Mechanical Engineering, Monash University

Dr Rafaat Ibrahim has a lead interest in the research and investigation of welding, heat treatment, pressure vessels design and structural integrity. He has led the design and investigation of failure and repairs of pressure vessels and critical structures in the electricity, chemical and food industries. Dr Ibrahim obtained his PhD from the UNSW in the area of fracture mechanics and his MEng (Material Engineering) in the field of heat treatment of ferrous alloys. He has also actively carried out research in the area of welding, materials and finite element analysis, which includes elastic, elastic-plastic and rigid-plastic/visco-plastic materials behaviour.

Dr. Ibrahim is an experienced supervisor of postgraduate research students and research activities and has supervised and examined for other universities at PhD level. He has more than 80 journal and conference publications in the optimisation of welding patterns to minimise residual stresses, finite element and fracture mechanics areas.

Prior to joining Monash, Dr Ibrahim gained research experience working as Senior Research Fellow at the Aeronautical Research Laboratories (ARL) and Melbourne University (Department of Mechanical and Manufacturing Engineering) in the area of fracture mechanics. Dr Ibrahim has been a consultant to BHP Research, Uncle Ben’s P/L Work-cover (Victoria and NSW), and Energy Development Pty/Ltd, and

He is a key researcher in the development of the round circumferentially notched K_{Ic} specimen. This specimen permitted the evaluation of plane strain fracture toughness from batch production material and failed components. The technique for testing plane strain fracture toughness using the small specimens is currently being considered by the relevant ASTM committee in the USA.