

A GENERIC DATABASE FOR WHOLE LIFE COSTING APPLICATIONS IN CONSTRUCTION

M. Kishk¹, A. Al-Hajj¹, R. Pollock¹ and G. Aouad²

¹Scott Sutherland School, The Robert Gordon University, Garthdee Road, Aberdeen AB10 7QB, UK

²School of Construction and Property Management, University of Salford, Salford M7 9NU, UK

This paper is the third in a series reporting on-going research within an EPSRC-funded research project undertaken by a joint collaboration between the Robert Gordon University and the University of Salford. This project aims to develop IT applications of whole life costing (WLC) to support the decision-making process in the design and management of construction assets. A WLC resource database has been designed and implemented in MS Access 2000® to accommodate WLC cost data and their time horizons and other data categories including quality and performance data. The database was designed on the basis of an in-depth analysis of the requirements of effective WLC decision-making during the design stage. The database relational structure is introduced and the data types and descriptions of fields of various tables are explained. In addition, three input forms designed to facilitate editing data are included. Finally, directions for further future research are introduced.

Keywords: automation, integrated databases, life cycle costing, whole life costing.

INTRODUCTION

In a previous paper (Al-Hajj *et al.*, 2001), the authors proposed a 5-step framework for implementing WLC in the design model within OSCON (Aouad *et al.*, 1997). The development of this framework was motivated by the fact that the absence of sufficient and appropriate data was, and still is, a major barrier to the application of WLC in the industry. The main feature of this framework is that it employs two databases: a resource database and a project database. The purpose of the resource database is to accommodate data for several options for various building elements to facilitate the development of a number of design alternatives. In this way, a meaningful WLC exercise can be undertaken to decide upon the optimum design alternative. Then, the set of options of the selected alternative are stored in the project database for later use in the effective management of the building over its life cycle.

The objective of this paper is to develop a generic WLC resource database suitable for data collection and recording. In the next section, the main requirements of the cost breakdown structure (CBS) are outlined. Then, the detailed design of the database is reported. Finally, directions for further research are introduced.

THE COST BREAKDOWN STRUCTURE

To undertake a WLC exercise, it is necessary to breakdown the building into elements whose costs can be distinctly defined and estimated. Various factors considered in the design of the cost breakdown structure (CBS) are reported in the following subsections.

The CBS Format

Within this project, the implementation medium will be an integrated environment with an object oriented CAD application (AutoCAD Architectural Desktop) to allow the user to create, manage and manipulate various components of the facility under consideration. In a typical CAD application, a facility is defined as a collection of objects. These objects are usually the components, elements, systems or subsystems of the facility. In other words, these objects represent a work breakdown structure (WBS) of the facility. This suggests that an elemental format is crucial for the implementation in the integrated environment. Furthermore, an elemental format relates well with the kind of decisions that are made at various design stages as noted by Kirk and Dell'Isola (1995).

Because the CBS should be coded to allow an analysis of specific areas of interest and to facilitate the flow of information around various life cycle phases, a selection of a standard WBS seems inevitable. The BMI codifications appear to offer a logical choice in the sense that BMI publications are the only regular sources of occupancy and maintenance data in the UK. However, it seems more reasonable to choose the well-known BCIS standard form of cost analysis because it is more element oriented. Besides, it is originally designed for initial costs and is combatable with OSCON.

Generic Cost Classification

By definition, cost data required for WLC purposes include initial costs and future follow-on costs that may include maintenance and repair costs, operating costs, replacement costs, disposal costs and resale values. The grouping of cost items into these categories allows producing various planning schedules and profiles and cash flow diagrams during various stages of the project. Thus, it is crucial to include the generic category of each cost item in the CBS.

Time Horizons

The times in the life cycle of the project when the cost-associated activities are to be carried out should also be recorded. These time horizons are crucial in the calculation of whole life costs. They are also necessary to develop various profiles and diagrams mentioned in the previous subsection.

It is also crucial to include a recurrence code to make a distinction between one-off, annual recurring, and non-annual recurring activities. In this way, the contribution of each cost item to whole life costs can be effectively calculated without the need to express the item by a number of cash flows over the analysis period (Kishk, 2001).

Other Crucial Data Requirements

Al-Hajj (1991) has shown that building-size and number-of-storeys as well as design-purpose influence the running costs of buildings. Even, different buildings used for the same purpose but with different physical aspects will incur different costs. Thus, the range of applicability of each cost for various building types, sizes, heights and locations should be recorded as well. In this way, cost data can be interpreted with physical data and the type of building that incurred them.

Data Normalisation

Hobbs (1977) and Flanagan *et al.* (1989) stressed the importance of the hours of use and occupancy profile as other key factors especially for public buildings such as hospitals and schools. This view was supported by Martin (1992) who showed that users and not floor-area had the greatest correlation with costs-in-use of hospitals.

Thus, other data normalisation methods, or rate codes, should be also employed depending on the basic nature of the cost under consideration. Another justification of this requirement is that no single source would provide the data for the database (Al-Hajj *et al.*, 2001). Examples of the required rate codes include cost per element area, per element length, per element volume, per gross floor area, per gross surface area and per building use.

Data Uncertainty

By definition, uncertainty is endemic to WLC. The inclusion of the effect of the building use, size, type and location as discussed above and the utilisation of different rate codes for various cost elements would eliminate some of the uncertainty in cost and time data. However, there is still a need to include the variability of cost and time data into the database. This variability can be represented by a distribution rather than by a single value. According to the type of uncertainty, either a probability distribution function (PDF) or fuzzy number (FN) is used (Kishk, 2001). For example, if multiple records of a data item for the same building type and size exist, the mean value, standard deviation, minimum and maximum values are recorded.

DETAILED DESIGN OF THE DATABASE

Based on the above arguments, the resource database has been designed. The relational structure of the database, names of tables and links between tables are outlined in figure (1). As shown, the database design was kept as general as possible. Data are stored in three main tables: (1) the element options table; (2) the option activities table and (3) the activities cost items table. Besides, eleven definition tables are also employed. These tables are explained in the following three subsections.

The Element Options Table

The element options table stores the basic data about available options of each building element. As shown in Figure 1, this table includes 18 fields to define the following 14 characteristics of each option.

The *ElementCode* field stores the BCIS code of the option as defined in the element codes table (Table 1), e.g. '2E' for an external wall element.

The *OptionCode* field is an automatically generated number that uniquely identify each option and is used to link the table to other tables in the database.

The *TypeCode* field stores a code that uniquely defines an element subtype, e.g. a 'pitched roof' for a roof element.

The *OptionName* field stores the name of the option.

The *OptionQuality* field stores the quality code of the option as defined in the Quality Codes Table (shown in Table 2).

The *IsResidential*, *IsIndustrial*, *IsRetailing*, *IsLeisure*, *IsOffice*, and *IsEducational* fields specify the applicability of the option to Residential, Industrial, Retailing, Leisure, Office, and Educational buildings, respectively.

The life expectancy of the option, like all uncertain variables in all tables, is defined by 5 fields. The first field, *OptionLifeDistributionCode*, stores the distribution code as defined in the 'Distribution Codes Table' (shown in Table 3). The other four fields stores 4 parameters that define the distribution (Table 4).

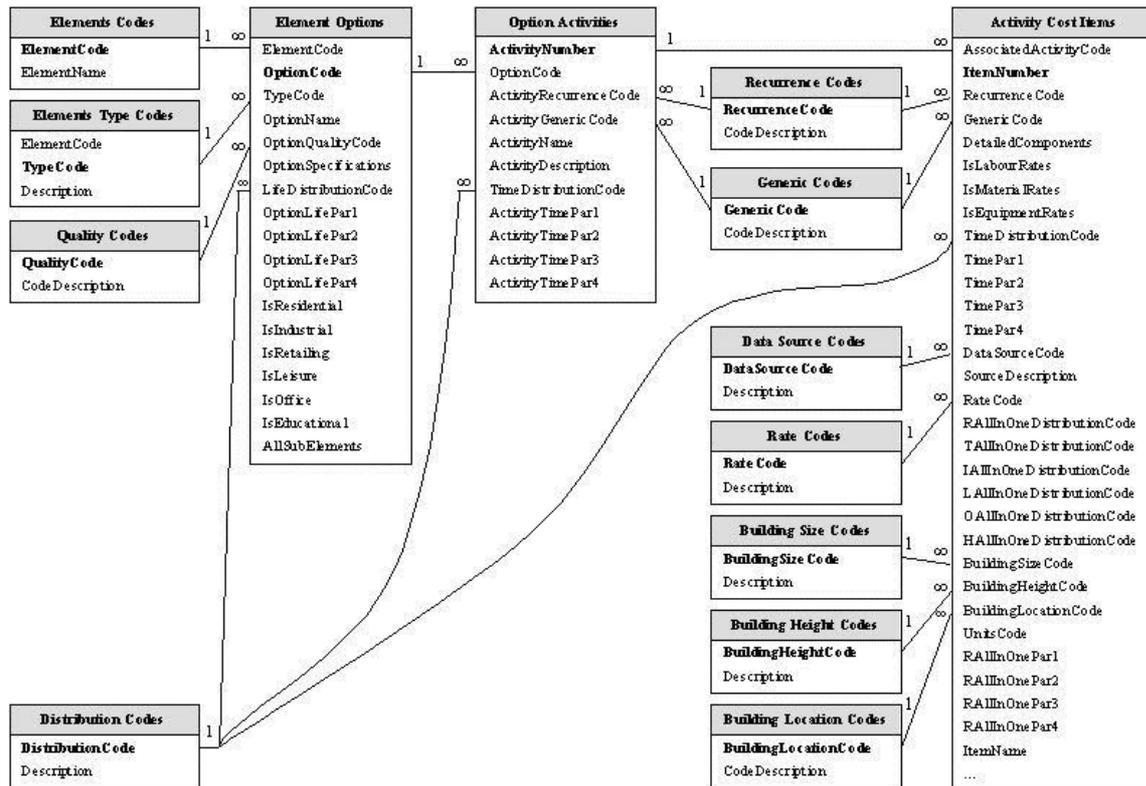


Figure 1: The resource database relational structure.

Table 1: The element codes table

Code	Element Name
2C	Roof Element
2D	Stair Element
2E	External Wall Element
2F1	Window Element
2F2	External Door Element
2G	Internal Wall Element
2H	Internal Door Element
3A	Wall Finish Element
3B	Floor Finish Element
3C	Ceiling Finish Element

Table 2: The quality codes table

Quality Code	Code Description
0	Low quality
1	Medium quality
2	High quality

The *AllSubElements* field specifies if the option records include all the activities and cost items of the corresponding object or not (the default value is ‘No’). For example, this field is set to ‘Yes’ for an external wall option of type ‘Glass Curtain Wall’ indicating that this wall needs no wall finishes.

The *OptionSpecifications* field stores the detailed specifications of the option.

Table 3: The Distribution codes table

Code	Description
0	Crisp (Certain)
1	Triangular Fuzzy Number (TFN)
2	Trapezoidal Fuzzy Number (TrFN)
3	Normal PDF (NPDF)
4	Trapezoidal PDF (TrPDF)
5	Triangular PDF (TPDF)

Table 4: Meaning of the distribution parameters

Type	Parameter 1	Parameter 2	Parameter 3	Parameter 4
Crisp	Certain value	N/A	N/A	N/A
TFN	Minimum value	Likeliest value	Maximum value	N/A
TrFN	Minimum value	Likeliest value 1	Likeliest value 2	Maximum value
NPDF	Mean value	Standard deviation	Minimum value	Maximum value
TrPDF	Minimum value	Likeliest value 1	Likeliest value 2	Maximum value
TPDF	Minimum value	Likeliest value	Maximum value	N/A

The Option Activities Table

The option activities table stores the basic data about the activities of the element options stored in the element options table. As shown in Fig. (1), this table includes 11 fields to define the following 7 characteristics of each activity.

The *OptionCode* field stores the option code to which the current activity belongs. This field links the table with the element options table.

The *ActivityNumber* is an automatically generated number that uniquely identify each activity and is used to link the table to ‘activities cost items table’.

The *ActivityRecurrenceCode* field stores the recurrence code of each activity as defined in the Recurrence Codes Table (shown in Table 5).

The *ActivityGenericCode* field stores the generic code of each activity as defined in the Generic Codes Table (shown in Table 6).

The recurrence time of the activity, like all uncertain variables in all tables, is defined by 5 fields. The first field, *ActivityTimeDistributionCode*, stores the distribution code as defined in the ‘Distribution Codes Table’ (shown in Table 3).

The *ActivityName* field stores the name of the activity.

The *ActivityDescription* field stores additional information about the activity.

The remaining four fields, *ActivityTimePar1*, *ActivityTimePar2*, *ActivityTimePar3*, and *ActivityTimePar4* store the 4 parameters that define the distribution (Table 4).

The Activity Cost Items Table

The activity cost items table stores the data about the cost items of the activities stored in the option activities table. This table includes 141 fields to define the following 15 characteristics of each activity.

The *AssociatedActivityCode* field stores the activity code to which the current cost item belongs. This field links the table with the option activities table.

The *ItemNumber* field is an automatically generated number that uniquely identify each cost item.

The *RecurrenceCode* field stores the recurrence code of each cost item as defined in the Recurrence Codes Table (shown in Table 5).

Table 5: The recurrence codes table

Recurrence Code	Code Description
0	Not Applicable
1	Annual Recurring
2	Non-annual Recurring
3	One-Off

Table 6: The generic codes table

Generic Code	Code Description
0	Initial
1	Maintenance
2	Replacement
3	Operating
4	Disposal
5	Resale

Table 7: The data source codes table

Data Source Code	Description
0	Historic Records
1	Price Books
2	Subjective
3	Supplier/Manufacturer
4	Other

Table 8: The rate codes table

Rate Code	Description
0	Per unit/job (i.e. lump-sum)
1	Per significant element dimension (e.g. length)
2	Per significant element area
3	Per element volume
4	Per building gross floor area
5	Per building gross surface area

The *GenericCode* field stores the generic code of each cost item as defined in the Generic Codes Table (shown in Table 6).

The *ItemName* field stores the name of the cost item.

The *DataSourceCode* field stores the data source code of the cost item as defined in the data source codes Table (shown in Table 7).

The *SourceDescription* field stores the description of the data source.

The *RateCode* field stores the data rate code of the cost item as defined in the rate codes table (shown in Table 8).

The *BuildingSizeCode* field stores the building size code of the cost item as defined in the building size codes table (shown in Table 9).

The *BuildingHeightCode* field stores the building size code of the cost item as defined in the building height codes table (shown in Table 10).

The *BuildingLocationCode* field stores the building size code of the cost item as defined in the building location codes table (shown in Table 11).

The *UnitsCode* field stores the unit system code of the cost item rate as defined in the unit codes table (shown in Table 12).

The recurrence time of the cost item, like all uncertain variables in all tables, is defined by 5 fields. The first field, *TimeDistributionCode*, stores the distribution code as defined in Table (3). The other four fields store 4 parameters that define the distribution as explained in Table (4).

Table 9: The building size codes table

Building Size Code	Description
0	All Sizes
1	< 100 Square Metres
2	100-200 Square Metres
3	200-500 Square Metres
4	500-1000 Square Metres
5	1000-5000 Square Metres
6	> 5000 Square Metres

Table 10: The building height codes table

Building Height Code	Description
0	All Heights
1	Single Story
2	2-3 Stories
3	4-6 Stories
4	6-9 Stories
5	9-12 Stories
6	> 12 Stories

Table 11: The building location codes table

Building Location Code	Description
0	All Locations
1	North
2	North East
3	North West
4	South
5	South East
6	South West

Table 12: The unit codes table

Unit Code	Description
0	Metric
1	Imperial

Table 13: The building types table

Building Type Code	Description
1	Residential
2	Office
3	Educational
4	Industrial
5	Retailing
6	Leisure

The *DetailedComponents* field stores the level of detail of the cost item. If this field is set to 'No', only all-in-one rate is given. If it is set to 'yes', on the other hand, the labour, material and/or equipment rates are specified according to the values of the *IsLabourRates*, *IsMaterialRates* and *IsEquipmentRates*, respectively.

The remaining fields define the all-in-one, labour, material and equipment rates for the six types of buildings defined in the building types table (Table 13).

EDITING AND ADDING DATA

In MS Access 2000, two ways of data entry are possible. In the first method, the targeted table is opened in datasheet view and new data can be entered directly. However, this method is not convenient especially when dealing with large amounts of data and/or with data related through two or more tables. This method has been adopted in adding data for simple auxiliary tables.

In the second method, data can be entered through a customised form that includes a selection of fields. This method has three advantages. First, it is user-friendlier. Secondly, additional information on the required data can be displayed. Thirdly, and more importantly, data for fields in two or more tables can be simultaneously entered and automatically included in these tables. Three user-friendly forms have been designed for adding data to the three main tables. Figures (2 to 4) show various controls of these tables. As shown, all forms are well organised. Besides, combo box controls are used, whenever possible, to further facilitate the input process. Moreover, event procedures written in Visual Basic[®] have been written for all controls to make these forms smarter. These features will be highlighted in a separate paper.

Figure 2: The option data input form

CONCLUSIONS AND FURTHER FUTURE RESEARCH

A generic WLC resource database has been designed and implemented into MS ACCESS 2000[®]. The database structure was designed on the basis of an in-depth analysis of the requirements of effective whole life costing. Data are stored in three main tables. Besides, eleven secondary tables including the definitions of various codes are employed. This flexible structure allows extracting data on four levels: the element, the activity, the cost item, and the cost component levels. In addition, three user-friendly input forms were designed to facilitate data entry.

An application is being developed to automate the design of building elements within an integrated environment. It will utilise the resource database to generate a set of design alternatives for an element and identifies the ideal option by minimising its whole life costs. Besides, it will update the element records in the project database to be used by another WLC management application. The development of these applications will be reported in two future papers.

Figure 3: The activities input form

Figure 4: The cost items input form

REFERENCES

- Al-Hajj, A (1991) *Simple cost-significant models for total life-cycle costing in buildings*. Unpublished PhD Thesis, Department of Civil Engineering, University Of Dundee.
- Al-Hajj, A, Pollock, R, Kishk, M, Aouad, G, Sun, M and Bakis, N (2001) On the requirements for effective information management in whole life costing within an integrated environment. *In: Kelly, J and Hunter, K (Eds.), The Annual Conference of the RICS Research Foundation, 3-5 September, Glasgow Caledonian University. RICS, Vol. 2, 402-413.*
- Aouad, G, Child, T, Marir, F and Brandon, P (1997) *OSCON final report*. University of Salford, UK.
- Flanagan, R, Norman, G, Meadows, J and Robinson, G (1989) *Life Cycle Costing - Theory and Practice*. BSP Professional Books.
- Hobbs, S (1977) *Collection and use of building maintenance cost data*. Building Research Establishment (BRE), Department Of Environment, Current Paper, October 1977.
- Kirk, SJ and Dell'Isola, AJ (1995) *Life Cycle Costing for Design Professionals*. New York: McGraw-Hill Book Company.
- Kishk, M (2001) *An Integrated fuzzy approach to whole life costing based decision making*. Unpublished PhD Thesis, Scott Sutherland School, The Robert Gordon University.
- Martin, J. (1992) Occupancy Costs. *Construction Papers*, No. 3, CIOB.