

Data Modeling for the Virtual Coach

By Eddy M. Rojas¹ and Amlan Mukherjee²

Abstract

The Virtual Coach is a web-centric virtual environment that creates temporally dynamic clinical exercises known as situational simulations, with the intention of exposing participants to rapidly unfolding events and the pressure of decision-making. This paper deals with the databases, which store all the data relevant to situational simulations executed by the Virtual Coach. It defines the nature of the data and describes the relational databases used. The Virtual Coach uses two types of databases, the 'as-planned' and the 'as-built.' While the 'as-planned' database stores information about how a particular project can be completed when executed on schedule and within budget, the 'as-built' database, dynamically records data regarding the actual execution of the project, by a particular participant. This paper discusses the design and the modes of functioning of these databases.

The Virtual Coach

Knowledge workers and decision-makers are essential for the success of construction processes. Unfortunately, experienced construction managers are retiring and taking their decision making skills with them, creating gaps of knowledge in the construction industry. Replacing these experienced professionals is a very difficult process, as most of these skills are acquired over the years. Web-based technologies provide solutions for various collaborative education environments (Rojas 2001), which can be used to

¹ Assistant Professor, Department of Construction Management. University of Washington. 116 Architecture Hall, Seattle, WA 98195-1610. E-mail: er@u.washington.edu

² Graduate Student, Department of Civil and Environmental Engineering. University of Washington. 116 Architecture Hall, Seattle, WA 98195-1610. E-mail: amlan@u.washington.edu

train construction professionals while exposing them to rapidly unfolding events and pressing situations requiring decision making skills.

The Virtual Coach environment is made up of three applications: the visualization engine, the emulation engine and the development engine. While the visualization engine manages the user interface, the emulation engine performs all computations necessary to implement the simulated environments. The development engine contains graphical components, which provides authors with a visual development environment to create simulation exercises.

The focus of this paper is the databases and their interactions in the Virtual Coach. The Virtual Coach has one main database, which contains the 'as-planned' project data. The 'as-planned' data indicates how a particular project can be completed when executed on schedule and within budget. There is another database, called the 'as-built' database, which dynamically records data regarding the actual execution of the project, by a particular participant.

The emulation engine props up situational scenarios in a random or planned fashion. 'Situational scenarios' are events that construction managers may encounter during a project. Instances include weather bad enough to bring work on the site to a halt, a sudden delay in material supply, labor scarcity due to a major nearby project, or any other unforeseen event that could potentially delay the schedule and topple the budget.

The functioning of the databases and all the related logistics is based on a set of mathematical equations that has been developed. These equations capture the logistics behind the construction process and form the basic framework of the Virtual Coach. The equations account for direct and indirect costs as well as project duration and productivity.

A Description of the Time-Activity Paradigm

Any project can be divided into a number of discrete time intervals of equal length. The total number of time intervals multiplied by the length of a time interval gives the total time of the project. The number of time intervals depends on the length and nature of the project. Usually, the length of a time interval is the period of time after which a construction manager would like to periodically monitor activities on the project. This period may vary, depending on circumstances, between a day and a month.

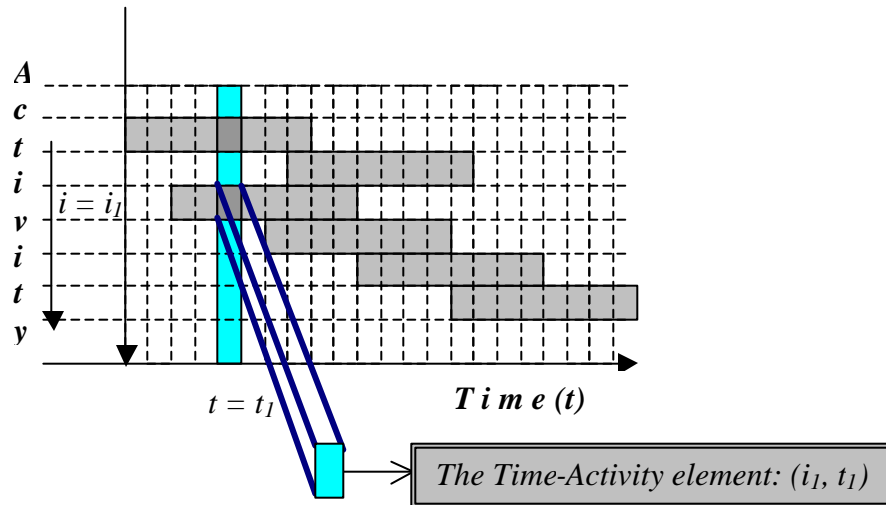


Figure 1: The Time-Activity Element

As a result of this time segmentation each activity is divided into a number of discrete elements. Hence, if a particular activity spans over ‘ n ’ time intervals then it may be divided into ‘ n ’ discrete elements. If ‘ n ’ is not an integral multiple of unity then, it is rounded off to the nearest integer greater than ‘ n ’.

As we see, this brings up the concept of the time-activity element (Figure 1). That is, the ordered pair (i, t) will always represent a unique element which belongs to the activity ‘ i ’ and spans over the time interval ‘ t ’. The set of equations and the databases have been developed with the discrete element (i, t) as the conceptual basis.

The equations assign properties to each of these elements. The properties include material, labor and equipment requirements of the activity ‘ i ’ over the time period ‘ t ’. The equations take into account a multiplicity in the identity of materials, labor and equipment. The time-activity element (i, t) has associated with it the different types of material, labor and equipment that are required by the activity ‘ i ’ during the time interval ‘ t .’ Each type of material, labor and equipment used in the construction project are uniquely identified. Figure 2, illustrates the point. Hence, a particular time-activity element (i, t) uses J different types of materials, L different types of skilled labor trades, and R different types of equipment.

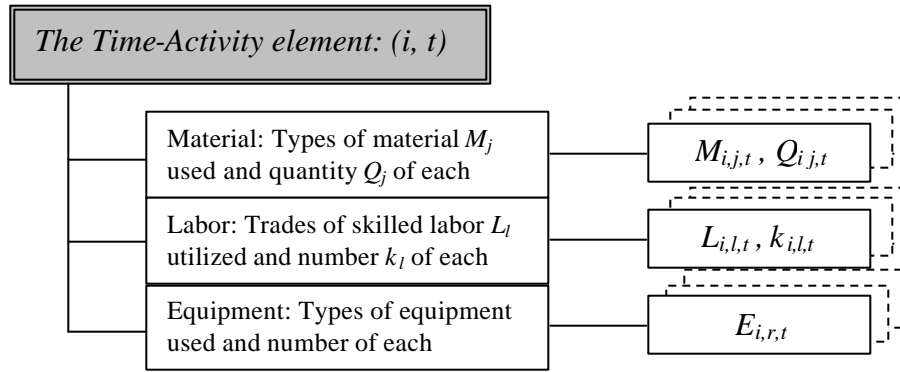


Figure 2: Properties of the Time-Activity Element

The j ' th material type is represented uniquely by $M_{i,j,t}$, while the quantity of it used is $Q_{i,j,t}$. The l ' th labor type is represented uniquely by $L_{i,l,t}$, while the number of journey persons employed is $k_{i,l,t}$. The r ' th equipment type is represented uniquely by $E_{i,r,t}$,

Effective storage and processing of the described information is very important for the functioning of the system. Using this information the equations can calculate the direct and indirect costs and also the productivity and remaining duration of an activity at any point of the project. During a simulation a comparison of the calculated parameters of 'what should have been' and 'what is actually happening' gives the participant continuous feedback regarding his/her performance. The 'as-planned' database contains stored information and remains undisturbed during the execution of a project. The 'as-built' database gets dynamically filled up during the execution of the project and reflects the various scenarios that may have popped up during the process. Once an event has occurred, the emulation engine refers to the 'as-planned' database to decide if the resource requirements are being fulfilled for the time-activity element at which the simulation has arrived. For example, atypical event may result in some kind of unavailability of different resource parameters such as material, labor and equipment. This needs the intervention of the participant to decide if he or she should stall the project, buy new resources or reallocate resources and carry on at a lower efficiency. The new schedule and cash flow pattern are stored in the 'as-built' database, by recording the different types of resources and related quantities associated with particular time-activity elements.

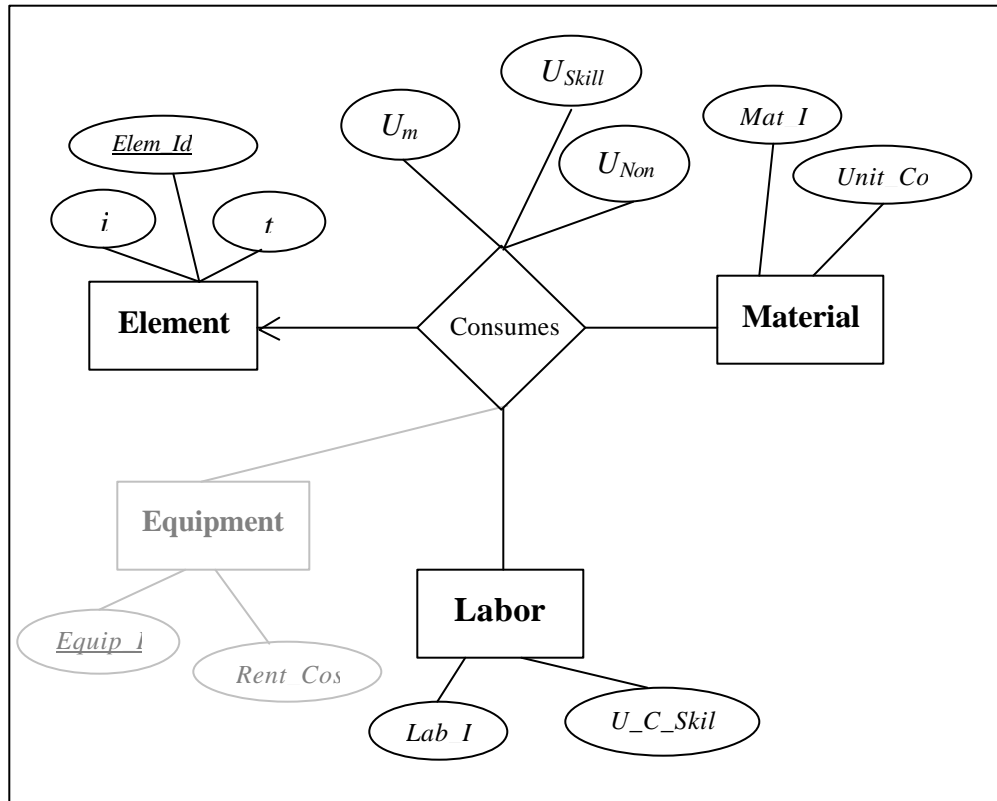


Figure 3: The Basic ER Diagram for the Virtual Coach databases

The Database Design

A very simple database schema, designed for the Virtual Coach databases, has been used to store the time activity element data. Figure 3, illustrates the ER-diagram for the databases. The entity *Element* represents the time-activity element. Even though the ordered pair (i,t) uniquely identifies an element, we have still used an unique element identification attribute, which will serve as the primary key to the *Element* table and as a foreign key to refer to elements from other tables. As their names suggest, *Material*, *Labor* and *Equipment* are entities, which represent the different types of material, labor and equipment used in the whole project and the unit costs of each such type. At this point of time, we have limited our study to the entities of *Element*, *Material* and *Labor* to simplify matters. Later on, the *Equipment* entity shall be included and dealt with using the same methods used to deal with the entities of *Labor* and *Material*. Henceforth, we shall ignore the entity *Equipment* for now.

Each of the entities *Material* and *Labor* have two attributes each. The first attribute uniquely describes a particular type of material or skilled labor trade that is utilized in the project. The *Lab_Id* and *Mat_Id* attributes are unique identifiers respectively in the *Labor* and *Material* entities. The *Unit_Cost* and *U_C_Skill* attributes describe the unit costs for the respective types of material and labor, as described by the named entities. Hence, the tuple, (*Mat_Id*, *Unit_Cost*) in the *Material* entity represents a particular material type uniquely defined by the Id, and its unit cost. The attributes *Lab_Id* and *Mat_Id* serve as primary keys respectively for the entities *Labor* and *Material*.

The relationship *Consumes* is a multi-way many-to-one relation from the entities *Material* and *Labor* to the *Element* entity. It presents a view of the types of material and labor and their respective quantitative metrics that a particular time-activity element uses. The attributes of the relationship, describe the units of the particular material used (U_m), the number of skilled crafts persons employed for the associated labor type (U_{Skill}) and the number of non-skilled labor associated with the element (U_{Non}). The relationship has foreign keys to each of the entities *Material*, *Labor* and *Element* through their primary identification keys. Hence, the tuple, (*Elem_Id*, *Lab_Id*, *Mat_Id*, U_m , U_{Skill} , U_{Non}) describes the quantity (U_m) of the material of type *Mat_Id* used, and the number (U_{Skill}) of skilled crafts person employed for the trade type *Lab_Id*; by the particular time-activity element, *Elem_Id*. The relationship is many-to-one since any particular time-activity element may use more than a single type of material and labor.

The model described in the previous section describes the very same logical sequence as the ER diagram does. There is a direct correspondence of the variables. The element entity directly corresponds to the Time-activity element (i, t). In the ER diagram ' i ' and ' t ' become attributes of the *Element* entity. $Q_{i,j,b}$ and $k_{i,l,t}$ represent the quantitative metrics of material and labor used. Hence, they directly correspond to U_m and U_{Skill} .

The Database Schema: The database schema can be best defined as follows.

Element (*Elem_Id*, i , t)

Labor (*Lab_Id*, *U_C_Skilled*)

Material (*Mat_Id*, *Unit_Cost*)

Consumes (*Elem_Id*, *Lab_Id*, *Mat_Id*, U_m , U_{Skill} , U_{Non})

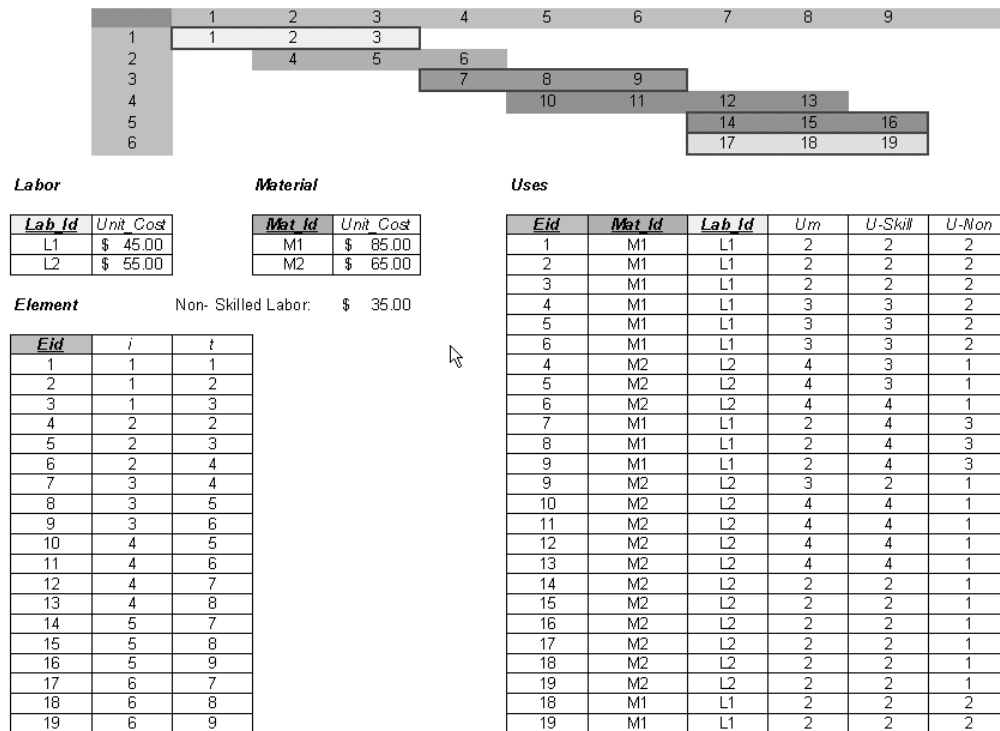


Figure 4: Instance of the database

Figure 4, illustrates an instance of the database. The ‘as-planned’ database consists of information regarding the schedule and the suggested allocation of material and labor resource for each time-activity element. It contains information regarding how a project manager would have planned the execution of the project. This information would not necessarily reflect project execution with optimal or maximal efficiency. Efficiency models would need to be calibrated according to the information in the ‘as-planned’ database. This is an issue discussed in the following section.

Implementation Issues

So far we have dealt with the design of the database. This section deals with the way in which the Virtual Coach interacts with the database in a situational simulation. We are now aware of the nature of the information and its logical arrangement within the database. The next issue is the relevance of the information with respect to a situational simulation. The information stored in the database regarding a particular construction project

has been termed as the ‘as-planned’ information for the project. However, the as-planned mode of execution of the virtual construction project need not be the most efficient and optimal method of execution. This gives the participant the opportunity, while in the environment, to come up with a comparably more efficient technique of executing the virtual project. The ‘as-planned’ information only serves as a benchmark to give the participant an idea of how far he or she is from achieving the original goals.

All the mathematical and logistical models dealing with the data are independent of the virtual construction project under consideration. Since the as-planned data may or may not be the most optimal execution technique all models are to be calibrated according to the data before the simulation begins. This method of calibration ensures that the same logistical models can be used across activities and across construction projects. The logistical models define certain trends and logical abstractions of behavioral patterns in the construction industry. The calibration dedicates the model to a particular activity. A typical trend described by a logistical model is the relationship between the composition and efficiency of a labor crew. A labor crew’s efficiency is directly proportional to the value of a ratio \mathbf{a} (the ratio of non-skilled to skilled personnel in a labor crew) till a particular value of \mathbf{a} ; after which increasing the number of unskilled helpers will not appreciably effect the efficiency of the crew.

For example, let us say there exists a time-activity element (i_l, t_l) in a particular project, which according to the as-planned database uses a crew size of $(n_1 + n_2)$ where n_1 is the number of skilled personnel and n_2 is the number of unskilled personnel associated with the time-activity element (i_l, t_l) . Now suppose the activity i_l , is worked on during the time interval t_l , with an efficiency of \mathbf{h} ($\mathbf{h} \leq 1$). Say $\mathbf{h} < 1$. Suppose, \mathbf{a} , the ratio of non-skilled to skilled personnel in a crew, is a crew factor that we define for all time-activity elements (i, t) . Hence, here $\mathbf{a} = \mathbf{a}_0 = n_2 / n_1$. We can say that for the time-activity element (i_l, t_l) with a crew factor of \mathbf{a}_0 the efficiency of work done is \mathbf{h} .

All logistical plots that are used during the simulation are calibrated using this information as the baseline. While executing the project, if participants decide to use a factor of \mathbf{a}_0' for the time-activity element (i_l, t_l) , then in reality they function at an efficiency level of \mathbf{h}' , but according to the calibrated model their level of efficiency is treated as \mathbf{h}'/\mathbf{h} . So if they function at the same efficiency as described in the as-planned database, they will be functioning at an efficiency of 1. The participant may at any point of time under-perform or over-perform with respect to the information in the

‘as-planned’ database. In this case the model, which is being calibrated, is the graph of \mathbf{a} versus \mathbf{h} .

We also need to take a closer look at the nature of the ‘as-built’ database. The ‘as-built’ database has the same schema as the ‘as-planned’ database. At the beginning of the simulation it is always empty and is then filled in as the simulation proceeds. The simulation has been modeled in terms of the variables defined in the database. Hence, a change to any of the variables will change the simulation. A change in the value of t in the tuple (i,t) , defining the activity i during the time interval t , will reschedule the activity. A change in the values of the number of skilled or unskilled labor personnel will reflect a change in attendance on site. A change in the unit price of materials could reflect a change in market rates. A change in the quantities of a particular type of material could reflect the unavailability of the material. Each situation that will be simulated can be expressed as a change in any combination of these variables. For example, a labor shortage during the time interval T , would mean that there is a reduction in the values of the number of skilled or unskilled labor personnel attached with all time-activity elements (i,T) . Similarly, if the participant decides to change the pace of the project by increasing efficiency of a labor crew he/she would change the value of \mathbf{a} , the ratio of skilled to non-skilled personnel in a crew. This is accomplished by changing the number of skilled or unskilled labor personnel associated with particular time-activity elements.

For the sake of data security, we do not want to disturb the data stored on the ‘as-planned’ database, which is a read-only database. Not to mention that the ‘as-planned’ database also contains the constraints of the model, that is issues such as the maximum space available to store materials on site, the maximum number of laborers available, etc. This creates the requirement for another database that notes the changes in variables during the simulation and records the new values. The changed values stored in the ‘as-built’ database reflect the events generated during the simulation and also the decisions taken by the participant, by recording the values of each variable as stored during the simulation of a particular virtual construction project. A comparison of the ‘as-planned’ and the ‘as-built’ database will at a later stage allow the participant to evaluate his/her performance. This is the very crux, of how the Virtual Coach will serve as an educational environment.

Conclusions

The development of a simulated construction management environment, such as the one created by the Virtual Coach requires both the definition of the underline relationships among the different variables that encompass the construction process and the implementation of a data management strategy. This paper only explains the very basic constructs of the databases used by the Virtual Coach and does not describe its mathematical model of cause-effect. Future publications will further elaborate in this area as it represents an important aspect of the situational simulation model.

The next step in the development of the Virtual Coach consists on the creation of the various events, which will simulate a variety of real life situations in the construction industry.

References:

Ramakrishnan R. & Gehrke J. “*Database Management Systems*” Second Edition, McGraw Hill Higher Education, 2000.

Rojas, E “*Fostering Collaborative Education Through Internet Technologies*” Journal of Engineering Education, October 2001, pg 623.