A Process-Oriented Scientific Database Model

Dr. J. Michael Pratt[1] and Dr. Maxine Cohen[2]

Addresses: (1) Assistant Professor of Computer Information Systems, Elmira College, Elmira, NY 14901-2099 (2) Lecturer of Computer Science, The Watson School of Engineering and Applied Sciences, SUNY Binghamton, Binghamton, NY 13902-6000

Abstract: A database model is proposed for organizing data that describes natural processes studied experimentally. Adapting concepts from object-oriented and temporal databases, this process-oriented scientific database model (POSDBM) identifies two data object types (independent and dependent variables) and two types of relationships (becomes-a and affects-a) between data objects. Successive versions of dependent variable objects are associated by the becomes-a relationship, while independent and dependent variable objects are associated by the affects-a relationship. Thus, a process can be viewed as a sequence of states (versions) of a dependent variable object whose attributes are affected over time by independent variable objects.

INTRODUCTION

One underlying reason for recent, growing interest in the object-oriented database model is its ability to capture and realistically represent complex, real-world entities (objects) and their interrelationships, including generalization (is-a), specialization (is-an-example-of), aggregation (is-a-part-of), and association (is-related-to). Database systems relying on the object-oriented database model (OODM), as well as most other models, typically organize data about real-world objects forming "snapshot" views of those objects and their relationships. OODM systems have been used effectively in data processing environments that have to store, retrieve, and manipulate complex, dynamic data objects, such as CAD designs, documents, and program modules (Banerjee, et al. 1987; Blaha, et al. 1988; Zhoa and Roberts, 1988).

The innumerable processes manifested over time in the behaviors of real-world objects are just as real as the objects that participate in those processes. A diverse repertoire of physical, chemical, and biological processes of varying complexity account for the dynamic nature of our world. Birth, life, and death of stars, life cycles of living things, and nutrient cycles occurring in our air, water, and soil are but a few examples of processes constituting sources of invaluable information that can help scientists understand and explain how and why such processes occur as they do.

Given the pervasiveness and significance of natural processes, it seems insufficient to capture data only about static objects and their interrelationships. What is also needed by scientists is a realistic modeling paradigm for capturing, organizing, manipulating, and retrieving data about processes being studied experimentally. This paper describes a process-oriented, scientific database model based on concepts from object-oriented (Banerjee, et al. 1987), temporal (Shoshani, 1986), and scientific databases (Shoshani, 1984) coupled with the notion of versioned objects (Katz and Chang, 1987; Beech and Mahbod, 1988).

DATABASES AS RESEARCH TOOLS

Laboratory experiments can generate volumes of data, which then have to be accurately recorded, carefully organized, easily retrieved, and eventually analyzed and summarized for dissemination. Automating experiments and data acquisition can afford scientists an efficient way to conduct their experiments and collect data. Commercial data acquisition systems store data in sequential files for later export to spreadsheets, to statistical and graphical packages, to word processors, and to interactive presentation systems. Sequential data files, however, lack many of the advantages of databases, including data sharing, controlled data redundancy, data independence, and query support. A database, therefore, would appear to be a better way to store and organize data captured by automated data acquisition systems.

OVERVIEW OF A PROCESS-ORIENTED SCIENTIFIC DATABASE MODEL

A database storing data about natural processes should take into account both time and change. Processes occur over time during which an object attains a particular state before changing into another state. Given the fundamental notion that a process unfolds as a sequence of object states, the proposed process-oriented database model adopts the concepts of state, state transition, input, and output from automata theory:

1. a *state* corresponds to the set of attribute values an object possesses at a particular instant in time

2. a *state transition* corresponds to a *change* in an object's attribute values

3. *inputs* correspond to the *attribute values* of objects that may affect the state transitions of other objects

4. *outputs* correspond to the set of attribute values that characterize an object in its new state.

A hypothetical experiment to study the effect of a newly developed fertilizer on tomato plant growth serves as an example of how the proposed process-oriented database model applies the concepts of state and state transition. Following traditional experimental design principles, the researcher's basic plan for determining the effectiveness of a new fertilizer to enhance growth involves exposing tomato plants to different concentrations of the fertilizer and observing the effects, if any, on their growth. Thus, tomato plant growth is the *dependent variable* and fertilizer is an *independent variable.* Plant growth is the dependent variable because the researcher wants to study how it is affected, if at all, by a particular fertilizer. The fertilizer is an independent variable because the researcher wants to ascertain its effect on growing tomato plants rather than the converse. The researcher recognizes that other independent variables such as soil moisture, pH, and temperature also may affect growth. To help isolate and measure the effect of the fertilizer separate from the effects of these other factors, the researcher applies the fertilizer in different concentrations (e.g., 0% and 10%) to separate groups of plants, while maintaining all other independent variables at the same levels for all groups. One group of plants, the *control*, receives no fertilizer, while one or more *treatment* groups each receive a different precise amount of fertilizer. During the experiment, the researcher periodically measures relevant independent variables and the parameters chosen as indicators of plant growth (e.g., plant height).

During this experiment, a seed (state 1) must germinate (transition 1) and become a seedling (state 2) before it can grow (transition 2) into a mature, flowering plant (state 3) and eventually into a fruit-bearing plant (state 4). Whether an object undergoes a transition from one state to another will depend on the states of other objects. Thus, a tomato seedling will grow into a mature plant only if there is sufficient water in the soil. The transition of an object from state to state can be characterized as (1) subtle or profound, depending on the degree to which successive states are similar and (2) slow or rapid, depending on the length of time it takes for a transition from one state to another to occur. Even when a transition alters an object so that the original object is no longer recognizable (e.g., as when a tomato seed becomes a seedling), the object still possesses a unique identity that is immutable as other attributes change, appear, or disappear. Based on the foregoing description of an experiment, the proposed process-oriented data model recognizes two distinct object types.

1. an *independent object type*, which corresponds to an independent variable. In the plant growth experiment, fertilizer would be an independent object, and its attributes (e.g., the proportion of fertilizer to soil) serve as inputs to the state transitions of some dependent object (e.g., a plant). In the resulting *interaction*, the inputs of an independent object may *affect* a state transition of a dependent object with which it interacts. An independent object's input may affect all, some; or none of the state transitions an dependent object undergoes. If the inputs do affect one or more of these transitions, then the independent object can be said to affect the process manifested in those transitions (Figure 1).

2. a *dependent object type*, which correspond to a dependent variable. A dependent object can undergo state transitions, which collectively represent some process. The attribute values characterizing a dependent object's state represent outputs resulting from a previous transition (Figure 2).

In addition to object types, a process-oriented database model needs to include a small set of basic relationship types.

1. Becomes-a. Object A_1 *becomes* object A_2 such that A_1 and A_2 are immediate *versions* of one another. A becomes-a relationship, symbolized as an arrow (→), corresponds to a state transition and, thereby, contributes to a process. Example: A tomato seed (A_1) becomes a seedling (A_2), which becomes a flowering plant (A_3), which becomes a fruit-bearing plant (A_4). This sequence of versioned objects and becomes-a relationships ($A_1 \rightarrow A_2 \rightarrow A_3 \rightarrow A_4$) represents the growth of a tomato plant. The becomes-a relationship has embedded within it an implicit *dependency* in which the existence of version A_n depends upon the existence of a previous version A_{n-1} while the converse is not true. A given version of an object only implies one or more potential successor versions.

Reversibility is another property of the becomes-a relationship. A becomes-a relationship is reversible if once A_n becomes A_{n+m}, then version A_{n+m} (eventually) can become a version essentially identical to A_n (i.e., $A_n \rightarrow \ldots \rightarrow A_{n+m} \rightarrow \ldots \rightarrow A_n$ as in water → water vapor → water). A becomes-a relationship also may be irreversible such that once A_n becomes A_{n+m} then A_n can not be restored (e.g., living plant → dead plant, but not the reverse). The becomes-a relationship unfolds in two distinct ways based on the degree of similarity between successive versions of a particular object:

a. version A_n becomes version A_{n+1} through *modification* of one or more attribute values. Thus, versions A_n and A_{n+1} possess the same attribute set, but one or more corresponding attribute values differ. Example:

A 2-inch tall tomato seedling grows into a 3-inch tall tomato seedling.

b. version A_n becomes version A_{n+1} by *transformation* in which A_{n+1} has acquired or lost one or more attributes, such that A_n and A_{n+1} have different though not necessarily disjoint sets of attributes. Though versions A_n and A_{n+1} possess different attributes as a result of version transformation, the underlying identity of the object remains immutable. Example: A particular 3-inch tomato seedling grows into a flowering plant in which the number of flowers is a new attribute.

The sequence of becomes-a relationships comprising a given process may include only version modification, version transformation, or a combination of both.

2. Affects-a. An independent object *affects-a* dependent object. The attri-bute value(s) of independent object B serve as inputs to the becomes-a relationship between A_n and A_{n+1} and thereby affect the attributes of A_{n+1}. Object B may modify the attribute values exhibited by A_{n+1} or transform the attribute set of A_n into that of A_{n+1}. Examples: Water affects a seed, which transforms into a seedling; fertilizer enhances seedling growth. An inde-pendent object, however, may not affect the dependent object under investigation. Thus, A_n becomes A_{n+1} regardless of the independent object B. Example: A tomato plant grows one inch in three days regardless of whether it received any fertilizer.

   The affects-a relationship between an independent object and a dependent object exhibiting change becomes a prime candidate as a causal relationship. The independent variable causes or induces the dependent variable to change its state and consequently influences the process in question.

3. Is-a-replicate-of. One object is a *replicate* of another object. Each independent and dependent object type may exist in two or more different independent states at the same time. This situation typically occurs in experiments in which the researcher wishes to determine if different replicates (e.g., B[1], B[2], B[3]) of an independent object affect replicates of a dependent object (e.g., A[1], A[2], A[3]) differently. Example: The researcher applies no fertilizer (B[1]) to tomato seedlings in the control group (A[1]...A[4]) and two different fertilizer levels (B[2] and B[3]) to seedlings in two treatment groups (A[5]...A[8] and A[9]... A[12]), respectively. Exposing seeds and plants to these three different fertilizer levels provides a basis for determining whether a particular fertilizer has any effect on tomato plant growth and, if so, to what extent (Figure 3).

APPLYING THE PROCESS-ORIENTED MODEL

The independent and dependent variables in a typical experiment can be mapped to an integrated set of tables (Table 1). More specifically, all the replicate versions (Rep No.: 1, 2, ..., 6) and temporal versions (Date mmddyy: 121791, 122091) of the independent experimental variable map to one table (Table 1F) containing a single row for each replicate-temporal version (1-121791, 1-122091, ..., 6-122091). The primary key for this table is a composite one, consisting of a value designating the independent experimental variable's relation to a particular experiment (Exp No.), replicate number (Rep No.), and date. An independent variable inherits a set of data from from its class type (Table 1E), which describes it in terms of static class data name (IVar: Fertilizer), source (Acme), chemical makeup (Formula: N28P2K2, i.e. 28% nitrogen, 2% phosphorous, and 2% potassium), cost ($10/100 lbs), and name (GrowFast). All the controlled independent variables (water, light, pH, temperature) collectively map to a separate table (Table 1G), with experiment number, replicate number, and date forming the primary key. The rows in this table normally store the same values for each independent variable because the researcher attempts to hold these variables constant during an experiment. Consequently, it would be more efficient to store only data in a row that deviate from the default values chosen for those variables. Each controlled independent variable also would inherit a set of data from its corresponding class type (tables not shown), describing that independent variable in terms of relevant information that would tend to remain unchanged during an experiment (e.g, source of water).

The dependent variable under study maps to one or possibly two or more tables, depending on whether temporal versions of the dependent variable arise through modification only or from both modification and transformation. If versions result from modification, they map to one table containing a single row for each replicate-temporal version (Table 1A shows growth data of three tomato seedlings in a control group: Rep. No. 1, 2, 3; and three seedlings, in a treatment group: Rep. No. 4, 5, 6). The primary key for this table is a composite one, consisting of a particular experiment number (Exp. No. 1), a replicate number (Rep. No. 1, 2, ..., 6), and a date (Date: 121791, 122091). If temporal versions also arise from transformation, then mapping involves two or more tables. Each time a dependent variable undergoes transformation (i.e., loses or gains attributes), a new table with a different set of attributes is created and data describing the dependent variable's transformed state fill a row in the new table (Tables 1C: flowering tomato plants and 1D: fruiting tomato plants). The primary key of each and every table created for a dependent variable also consists of experiment number, replicate number, and date, allowing joins between or among any and all dependent object tables describing a given dependent variable.

Each dependent object also inherits relevant attributes of its class type (Table 1A). The class attributes of the dependent variable in the example experiment might include its class-type name (DVar: tomato), subclass-type name (Variety: BigRed), source (Source: Seeds, Inc.), and the age (1) of the seeds.

With a database of versioned objects, one could store and organize data in ways that realistically model an experiment and the process under study. For example, a researcher wanting to know when seedlings exhibited maximum rates of growth might issue the query, "Select time period of maximum seedling growth." Later data analysis and interpretation also could be facilitated by querying the database to extract sets of data for statistical analyses. One may want to determine, for example, whether the differences in growth observed among the control and treatment seedlings are significant. To test the data statistically and then show the results graphically, a researcher might issue the query, "Select date-matched seedling growth data from control and treatment groups and perform analysis of variance test and plot." The selected rows then would be exported to statistical and graphing modules for further processing.

CONCLUSIONS

The growing volume, diversity, and complexity of experimentally generated data can overwhelm traditional "notebook" methods for organizing scientific data and their interrelationships. Consequently, the familiar lab notebook is not the ideal tool for processing all the data relationships important to interpreting research findings efficiently and accurately. Addressing this problem, Lander, et al. (1991) state that "computing methods are needed that allow efficient and accurate processing of experimentally gathered data." To this end, Lander and his colleagues identify computerized scientific databases as vital tools that scientists can and should use to help process their research data. The immediate challenge they identify is developing methods for effectively organizing, storing, and retrieving data along with their associated relationships. Meeting this challenge successfully will involve a collaborative effort of computer scientists and laboratory scientists. The principle contribution of computer scientists will be "the invention of languages [and data models] for describing complicated processes that occur in some order..." (Lander, et al, 1991).

The proposed process-oriented, scientific database model represents a conceptual tool that may help scientists organize, process, and retrieve their data in ways that facilitate its interpretation and dissemination. By coupling object-oriented and temporal database concepts with versioning capability, it is possible to model not only static objects and their relationships but also dynamic objects that change and interact as they participate in and define some natural process studied experimentally.

REFERENCES

Banerjee, J., Chou, H., Garza, J., Kim, W., Woelk, D., Ballou, N., and Kim, H., "Data Model Issues for Object-Oriented Applications." *ACM Transactions Office Info. Systems*, 5(1), pp. 3-26, 1987.

Beech, D. and Mahbod, B., "Generalized Version Control in an Object-Oriented Database." *Proceedomgs IEEE Data Engineering Conference*, pp. 14-22, 1988.

Blaha, M. C., Premerlani, W. J., and Rumbaugh, J. E., "Relational Database Design Using an Object-Oriented Methodology." *Communications of the ACM*, 31(4): 414-427, 1988.

Katz, R. H. and Chang, E. "Managing Change in Computer-Aided Design Databases." *Proceedings International Conference on VLDB*, pp. 455-462, 1987.

Lander, E. S., Langridge, R., and Saccocio, D. M., "Mapping and Interpreting Biological Information." *Communications of the ACM*, 34(1): 33-39, 1991.

Shoshani, A., Olken, F., and Wong, H. K. T., "Characteristics of Scientific Databases." *Proceedings 10th International Conference on VLDB*, pp. 147-160, 1984.

Shoshani, A. and Kawagoe, K., "Temporal Data Management." *Proceedings 12th International Conference on VLDB*, pp. 79-88, 1986.

Zhoa, L. and Roberts, S. A., "An Object-Oriented Data Model for Database Modelling, Implementation and Access." *The Computer Journal*, 31(2): 116-124, 1988.

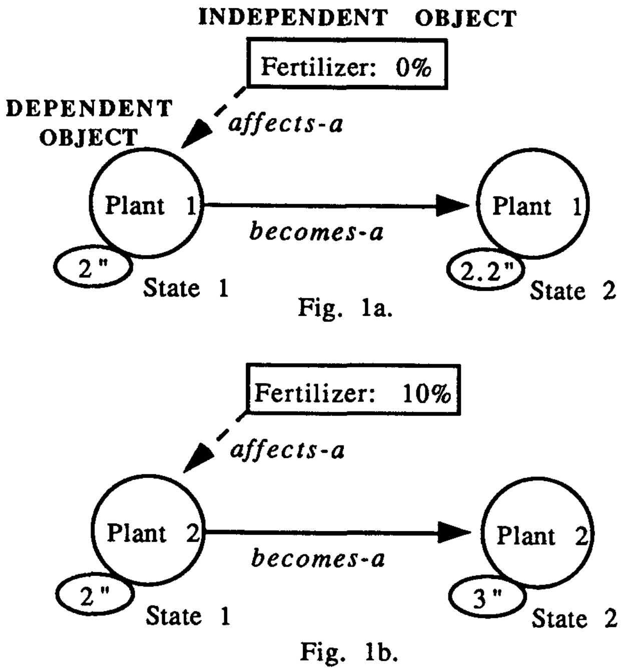

Fig. 1a.

Fig. 1b.

Figure 1. Two fertilizer concentrations (experimental independent objects) affecting state transitions (becomes-a relationships) of two plants (dependent objects).

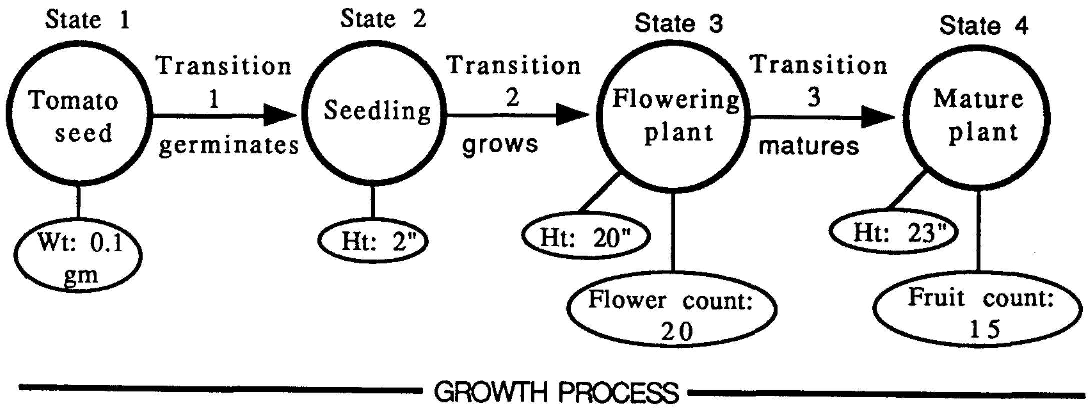

Figure 2. A series of state transitions modeling growth of a tomato plant. Tomato seed germinates into a seedling , which grows into a flowering plant, and then into a fruit-bearing plant.

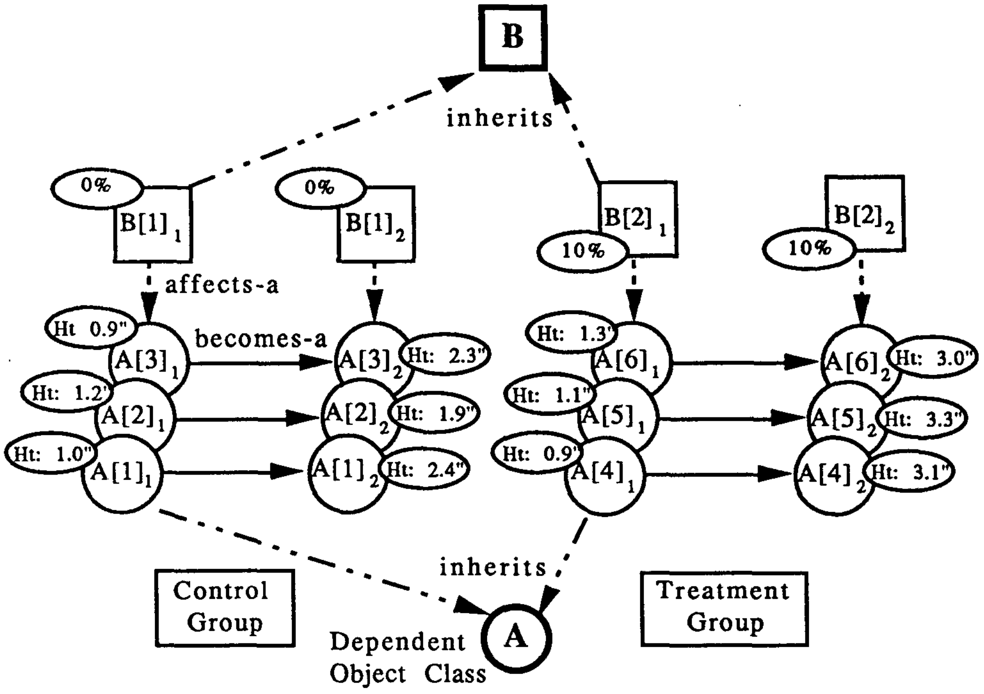

Figure 3. Process-oriented model applied to an experiment on tomato plant growth, showing two experimental groups (Control and Treatment), two object classes (A and B), replicates (A[1] .. A[6] and B[1] .. B[2]), versions (e.g., $A[1]_1$ and $A[1]_2$ .. $A[6]_1$ and $A[6]_2$), and the relationships child-of (inherits), becomes-a, and affects-a. Class objects (fertilizer and tomato) appear as bold figures, independent objects as squares, dependent objects as plain circles, and attributes as ovals. Solid arrows (→) depict becomes-a relationships; plain broken arrows (— →), child-of relationships; bola broken arrows (— →), affects-a relationships.

A. Tomato type

Exp No.	DVar	Variety	Age	Source
1	Tomato	BigRed	1	Seeds Inc

B. Tomato seedlings

Exp No.	Rep No.	Date	Ht cm	No. Leaves	Stem Width
1	1	121791	1.0	2	0.1
1	1	122091	2.4	4	0.1
1	2	121791	1.2	2	0.1
1	2	122091	1.9	4	0.1
1	3	121791	0.9	2	0.1
1	3	122091	2.3	4	0.1
1	4	121791	0.9	2	0.1
1	4	122091	3.1	4	0.1
1	5	121791	1.1	2	0.1
1	5	122091	3.3	4	0.1
1	6	121791	1.3	2	0.1
1	6	122091	3.0	4	0.1

C. Flowering tomato plants

Exp No.	Rep No.	Date	Ht cm	No. Leaves	No. Flowers
1	1	021592	5.5	12	6
1	1	021892	5.6	14	8
1	2	021592	5.3	10	4
1	2	021892	5.4	13	5
1	3	021592	5.7	11	5
1	3	021892	5.8	12	7
1	4	021592	8.6	14	10
1	4	021892	8.7	16	14
1	5	021592	9.1	15	8
1	5	021892	9.3	16	12
1	6	021592	10.0	13	9
1	6	021892	10.2	15	12

D. Fruiting tomato plants

Exp No.	Rep No.	Date	Ht	No. Leaves	No. Fruit
1	1	031592	9.5	19	10
1	1	031892	10.0	19	10
1	2	031592	8.0	18	8
1	2	031892	8.6	19	9
1	3	031592	9.4	17	11
1	3	031892	9.8	18	11
1	4	031592	11.7	24	15
1	4	031892	12.1	25	16
1	5	031592	12.3	25	13
1	5	031892	12.8	25	14
1	6	031592	12.7	23	16
1	6	031892	13.1	24	16

E. Fertilizer type

Exp No.	IVar	Source	Formula	Cost	Name
1	Fertilizer	Acme	N28P2K2	10.00	GrowFast

F. Fertilizer treatments

Exp No.	Rep No.	Date	Conc %
1	1	121791	0.0
1	1	122091	0.0
1	2	121791	0.0
1	2	122091	0.0
1	3	121791	0.0
1	3	122091	0.0
1	4	121791	10.0
1	4	122091	10.0
1	5	121791	10.0
1	5	122091	10.0
1	6	121791	10.0
1	6	122091	10.0

G. Other independent variables

Exp No.	Rep No.	Date	Water ml	Light	pH	Temp °C
1	1	121791	50	100	6.0	25
1	1	122091	50	100	6.1	24
1	2	121791	50	100	5.9	25
1	2	122091	50	100	6.0	24
1	3	121791	50	100	6.0	25
1	3	122091	50	100	5.9	24
1	4	121791	50	100	5.9	25
1	4	122091	50	100	6.0	24
1	5	121791	50	100	6.0	25
1	5	122091	50	100	6.1	24
1	6	121791	50	100	6.0	25
1	6	122091	50	100	5.9	24

Table 1. Hypothetical data summarizing effect of fertilizer on plant growth. Table A shows class data about the dependent variable. Tables B, C, and D contain data describing the dependent variable. Tables E, F, and G store data describing the independent variables.