December 2001

Business Information Systems Modeling with Computer Assisted Software Systems Engineering (CASE) Software

Satya Saraswat

Bentley College

Follow this and additional works at: http://aisel.aisnet.org/amcis2001

Recommended Citation
http://aisel.aisnet.org/amcis2001/429

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2001 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.
BUSINESS INFORMATION SYSTEMS MODELING WITH COMPUTER ASSISTED SYSTEMS ENGINEERING (CASE) SOFTWARE

Satya Prakash Saraswat
Computer Information Systems Department
Bentley College
Ssaraswat@bentley.edu

Abstract

Using Visible Analyst Workbench as a CASE tool for business information systems modeling, this workshop will demonstrate how to create: (a) Comprehensive process models with multi-level Data Flow Diagrams, (b) Data models with normalized Entity Relationship Diagrams, (c) Structure Charts for automatic code generation, and (d) Functional Decomposition Diagrams as system planning tools. In addition, the workshop will explain the effectiveness of CASE tools in identifying syntactical and logical errors of information flow in the models. Four structured exercises will integrate the conceptual foundations of the Systems Development Life Cycle with the practical difficulties of building complex business information systems models. The exercises used in the workshop will also explain how simulated projects and CASE software can be used as pedagogical tools for foundational courses in systems analysis and design. The workshop will also discuss the problems frequently encountered in teaching short CASE course as a co-requisite for the primary systems course in the CIS discipline.

Introduction

The Computer Information Systems (CIS) department at Bentley College offers two courses in its undergraduate program for CIS majors. One of these courses, “Analysis, Modeling, and Design,” introduces students to the theory of information systems analysis and design. In this course the students learn formal methodologies that constitute the foundation of information systems development practices. This is a full-semester course and the wide range of topics covered in this course makes it impossible to provide students any significant practical experience with the process of systems analysis and design with modern Computer Assisted Systems Engineering (CASE) tools. In order to address this difficulty, the faculty of the CIS department created a specialized half-semester course in 1997. This course, taken by students as a prerequisite or co-requisite for the analysis and modeling course, is being taught with Visible Analyst Workbench CASE software available from Visible Systems Corporation. Students in the class are given a comprehensive simulated exercise to be completed in groups of two or three in six weeks.

Based on the experience of teaching this course for four years, this workshop will demonstrate how a CASE tool can be effectively utilized with a simulated exercise to create data flow diagrams, entity relationship diagrams, functional decomposition diagrams, structure charts, and unstructured project diagrams. In addition to creating these diagrams, the workshop will demonstrate how a CASE tool can identify and correct syntactical and logical errors of information flow in the data and process models.

The Situation Context For The Workshop

The workshop is based on the existing order processing operations of a small corporation in computer and electronics industry, known as Nandini Enterprises, Inc. In its basic operations, Nandini orders personal computers and electronic equipment from its suppliers, stocks them in a warehouse, and ships them to its customers located primarily in New England and New York. Customers normally send their orders and related information by phone or fax, but they can also walk in Nandini’s office and place an order. The shipment of items currently in stock is guaranteed within 24 hours after a customer order has been received,
registered, and verified. If some items are not available in the stock, the customers are so advised and the unavailable items are ordered from the suppliers immediately. When the items are shipped to the customer, an invoice is sent with the shipment. In case of partial shipment of a customer order, an advisory memorandum is sent to the customer indicating that the items have been ordered by Nandini from its suppliers and will be shipped to the customer within a reasonable period of time. Upon receiving their shipments, customers generally send their payments by check. These payments can also be made by a credit card, in case of telephone orders. Cash or credit cards are accepted for orders directly placed in the office and a receipt is given to the customer. The receipts for the payments received in the mail are sent to the customers by mail. Nandini’s inventory is continuously monitored and appropriate levels of stock are maintained for items frequently ordered by customers. At the reorder point, orders are sent to the suppliers for various items, and when shipments arrive, the inventory information is immediately updated. Inventory adjustments are also made when the items are shipped to the customers from the warehouse.

Nandini’s enterprise information system is based on personal computers interconnected with a local area network. This system is implemented as a collection of software modules that perform three distinct functions in the organization. The modularized design of the system has made it possible for Nandini to operate and maintain the system satisfactorily with the limited computer expertise currently available in the organization. The Order Processing module contains two processes. The Order Verification process is responsible for registering the order and checking the credit status of the customer. Nandini maintains a comprehensive relational database containing information about customers, suppliers, orders, and inventory. This database is utilized for verification of customer orders and customer credit. Relevant information about new and existing customers is continuously added to this database. The Invoicing process prepares invoices and transmits them to customers, with shipments. This process also prepares and sends to the customer advisory memoranda about partially shipped orders and orders that cannot be shipped due to credit or other problems. Copies of customer invoices are transmitted to the accounts management function for further action. The Accounts Management module of the enterprise information system also contains two processes. The Accounts Receivable process manages the collection of payments from the customers, and the Accounts Payable process is responsible for making timely payments to the suppliers. These processes also update the associated customer and supplier records in the database. The third module of the enterprise information system is Inventory Management. One process in this function, the Purchasing Process, is responsible for sending orders to the suppliers of Nandini. These orders are prepared and sent to Nandini’s suppliers when the stock levels for various items fall below their reorder point, or when a customer orders special items that are generally not stocked in Nandini’s warehouse. This process also updates the database when shipments arrive from Nandini’s suppliers. The Inventory Reporting process of the inventory control module, on the other hand, provides stock-status and other inventory reports to the management of Nandini on a weekly or daily basis.

Workshop Assignments

The work consists of four assignments of gradually increasing complexity. The first assignment consists of drawing a functional decomposition diagram, as the starting point of the project, and a context diagram containing only ONE composite data flow from each external entity to the information systems process. In this assignment, data flow diagrams are “spawned” from the functional decomposition diagram, and the processes in the data flow diagrams are “exploded” to create more detailed diagram at the subsequent levels.

Assignment 1 (eight sections):

(a) Prepare a functional decomposition diagram (FDD) showing all the functions and processes of Nandini’s enterprise information system.

(b) Analyze this diagram and save the error or warning messages generated by VAW.

(c) Create the Context Diagram [Level 0 DFD] for the system showing only one process. Show only one input, and one output data flow connected to each external entity in the diagram and give appropriate names to these flows indicating their comprehensiveness.

(d) Explode the process in your top-level data flow diagram (DFD) to generate a second-level data flow diagram. Show three processes in this diagram but DO NOT CONNECT the net data flows to the process symbols. Analyze this diagram and print out the error messages.
(e) **Spawn** each of the three functions at level 2 in your functional decomposition diagram (FDD) to create new sets of data flow diagrams. Select your **second level** DFD created in the previous step as the **parent** of the data flow diagrams created in this step. Update your diagrams with each selection.

(f) Again **analyze** your functional decomposition diagram (FDD), and note down the errors or warnings generated by the system. There should be no errors or warnings if you have successfully completed these steps.

(g) **explode** each process shown in your level 2 **data flow diagram** as follows:

   Explode a process, select the most appropriate existing diagram from the three diagrams spawned in step (e) above, and use the selected diagram as the child diagram for the process you are exploding.

(h) **Decompose** the process shown in your top-level data flow diagram and create an unstructured process decomposition diagram.

---

After the completion of the first assignment, the workshop participants are given additional details of the situation context. They are expected to create the second and the third level data flow diagrams, showing additional details of the information flow with relevant data stores and split data flows to understand the meaning of top-down analysis. The students are also expected to create a data model with normalized relationships and appropriate descriptions of key and non-items in the model. The requirements of the second assignments are given below:

**Assignment 2 (Five sections):**

(a) With the additional information provided in the situation context, update your Level 2 (next to context level) data flow diagram created in assignment 1. Add appropriate data store(s) to this diagram, split the net data flows, and add appropriate local data flows to model the situation more realistically. Analyze your diagram for inconsistencies, and correct all errors before submitting the assignment.

(b) Modify all Level 3 (the last level for the project) data flow diagrams created in assignment 1 with the additional information in the situation context. Add appropriate data flows and data stores to your model in each of the three levels of diagrams. Analyze your diagrams for inconsistencies and correct all errors before submitting the assignment.

(c) Create an appropriate Entity Relationship Diagram showing all the entities and relationships between them. Analyze your diagram for syntax and normalization, and record the errors generated by VAW.

(d) Submit a printed copy of all data flow diagrams and the entity relationship diagram with your assignment.

(e) Create two repository reports as follows:

   Report 1: This report should contain a split flows listing from the data flow diagrams, for the entire project, showing all entry characteristics, with multiple entries per page.

   Report 2: This report should contain a detailed listing of only “Process” entities in the second level data flow diagram, in the alphabetical order, with multiple listings per page.

---

At this point the, additional details of the situation context are provided for the third and the final assignment. The third assignment completes the three levels of data flow diagrams and creates a structure chart. The process selected for creating the structure chart is “order verifications process.” The sub-modules invoked by this process are identified in the modified situation context, and the parameters to be exchanged in the invocation process are given to the students. The requirements of this assignment are described below:
**Assignment 3 (Seven sections):**

(a) Update your data flow diagrams and Entity Relationship diagrams with the additional information provided to you.

(b) Define all your databases in the repository and enter the data items for each database in the composition field of the repository. Enter the name, type, length, and null value possibilities for key fields only.

(c) Perform key analysis to check the consistency of your design, and perform key synchronization to migrate all the foreign keys to appropriate databases.

(d) Create a repository report from your entity relationship diagram, containing a detailed listing, including all entity types from your main entity relationship diagram (a single diagram), showing all entry characteristics, sorted by entity type, with multiple entries per page. Attach your report with the assignment.

(e) Create a structure chart from the description provided in the modified situation context and analyze this diagram to ensure its correctness.

(f) Define appropriate length and type (physical characteristics) for each data item in the model.

(g) From the repository menu, run the model balancing option and modify your diagrams to ensure that the model is correctly balanced. The primary keys and other composition of the data stores should match the entities in the ERD’s. Set your ERD balancing rules to specify that all fundamental elements must be used on a DFD, and every entity must correspond to a data store.

**Conclusion**

The representative context diagram, second-level data flow diagram, entity-relationship diagram, and the structure chart for these three assignments are included in the appendix to this paper. The concept of “top-down” analysis is demonstrated in these diagrams by showing the equivalency between various levels of data flow diagrams, and by demonstrating the differences between local and net data flows at each level. The CASE tools can easily detect the violations of data flow rules and maintain all the required information in the repository. The workshop also demonstrates the repository functions and explains how the CASE tool can detect violations of rules contained in the repository. The workshop thus provides the participants a hands-on experience with a CASE tool. The experience gained from the workshop can be utilized for teaching systems analysis and design concepts more effectively.
APPENDIX

Representative Diagrams Created in the Workshop

(a) The context diagram:

(b) The second level data flow diagram:
(c) An Entity Relationship Diagram:

- CUSTOMER
- INVENTORY ITEM
- TRANSACTION ENTITY
- SUPPLIER
- PRICE ENTITY

Dashed arrows indicate relationships:
- CUSTOMER is sent by TRANSACTION ENTITY
- TRANSACTION ENTITY also sends SUPPLIER
- INVENTORY ITEM from supplier has a
- CUSTOMER charges for PRICE ENTITY

(d) A structure chart:

Nandini Enterprises, Inc.
Structure Chart
Order Verification Process