Communications of the Association for Information Systems

Volume 8 Article 13

February 2002

Special Issue on the AMCIS 2001 Workshops: Business Information Systems Modeling with Computer Assisted Systems Engineering (Case) Software

Satya Prakash Saraswat
Bentley College, Ssaraswat@bentley.com

Recommended Citation

Saraswat, Satya Prakash (2002) "Special Issue on the AMCIS 2001 Workshops: Business Information Systems Modeling with Computer Assisted Systems Engineering (Case) Software," *Communications of the Association for Information Systems*: Vol. 8 , Article 13.
DOI: 10.17705/1CAIS.00813
Available at: https://aisel.aisnet.org/cais/vol8/iss1/13

Communications of the Association for Information Systems

SPECIAL ISSUE ON THE AMCIS 2001 WORKSHOPS: BUSINESS INFORMATION SYSTEMS MODELING WITH COMPUTER ASSISTED SYSTEMS ENGINEERING (CASE) SOFTWARE

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

ABSTRACT

This paper describes and critiques a workshop in business information systems modeling with Computer Assisted Systems Engineering (CASE) tools held at the 2001 Americas Conference on Information Systems (AMCIS). The workshop demonstrated the effectiveness of Visible Analyst Workbench (VAW) as a systems modeling tool that generates comprehensive Process Models with multi-level Data Flow Diagrams, Data Models with Entity Relationship Diagrams, Structure Charts for automatic code generation, and Functional Decomposition Diagrams as system planning tools. A simulated teaching case with four structured exercises integrated the conceptual foundations of the Systems Development Life Cycle (SDLC) with the practical difficulties of building complex business information systems models. The workshop also discussed how simulated teaching cases and CASE software could be used as pedagogical tools for foundational courses in systems analysis and design.

Keywords: CASE, Systems Analysis and Design, Systems modeling, Visible Analyst.

I. INTRODUCTION

The Computer Information Systems (CIS) department at Bentley College offers two courses in its undergraduate program for CIS majors. One of these, “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 course requires a full-semester and the wide range of topics covered 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. 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 three years with a simulated teaching case, this workshop demonstrated how a CASE tool can be used effectively to create data flow diagrams, entity relationship diagrams, functional decomposition diagrams, structure charts, and unstructured project diagrams. In addition, the workshop also explained how a CASE tool is used to identify and correct syntactical and logical errors of information flow in the data and process models, and check the consistency of the system model.

II. THE SITUATION CONTEXT FOR THE WORKSHOP

The workshop was conducted using a simulated teaching case developed as a half-semester project for undergraduate classes and used successfully for 7 semesters. The case is based on the existing order processing operations of a small corporation in the 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 is 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. In the case of telephone orders, these payments can also be made by credit card. 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, and when shipments arrive, the inventory information is updated immediately. 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

- order processing
- accounts management
- inventory management.

The modularized design of the system makes it possible for Nandini to operate and maintain the system satisfactorily with the limited computer expertise currently available in the organization.

Order Processing. The Order Processing module contains two processes, Order Verification and Invoicing. 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 used to verify customer orders and customer credit. Relevant information about new and existing customers is continually added to this database. The Invoicing process prepares invoices and transmits them to customers, with shipments. This process also prepares and sends 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.

Accounts Management. The Accounts Management module of the enterprise information system also contains two processes: accounts receivable and accounts payable. 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.

Inventory Management. The Inventory Management module includes the Purchasing Process and the Inventory Reporting Process. The Purchasing Process, is responsible for sending orders 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, on the other hand, provides stock-status and other inventory reports to Nandini's management on a weekly or daily basis

III. WORKSHOP ASSIGNMENTS

Four assignments of gradually increasing complexity were presented during the workshop. Each assignment consists of a number of steps to be completed in the specified order. These assignments are described in the Appendix together with representative diagrams generated as a solution to each assignment. A brief description of the assignments with their relevance to the conceptual foundations of the System Development Life Cycle (SDLC) follows.

Workshop Assignment 1. Create a Functional Decomposition Diagram (FDD) as the starting point of the project. Functional decomposition diagrams, as simple hierarchy charts, are introduced as project planning tools. They show the overall structure of the system, containing its modules and their relationships as parents and children. Visible Analyst Workbench (VAW), used in this workshop, contains a function called "spawn" which generates these diagrams automatically and links them to the data flow diagrams through an option in the "explode" function. This assignment introduces the concepts of "top down" analysis and "functional decomposition" in the SDLC. The "top" of the system is modeled by the single function at the highest level in the FDD and the context diagram, which are equivalent. The context diagram shows only external entities, and "composite" data flows between the entities and the organizational information system. Additional details of the situation context are provided at this point before the students attempt the second assignment.

Workshop Assignment 2. Distinguish between two important types of models relevant to the SDLC – the process model and the data model. In the process model, the concept of top-down analysis is further clarified by "splitting" the composite data flows created in the first assignment into two or more secondary data flows. The model requires that all net data flows connected to any process in the system should be properly accounted for. Although local data flows can be added to the diagram as desired, net data flows belonging to external entities cannot be added at any other level except the context diagram. When the diagrams are "analyzed" for syntax, violations of these constraints are detected and appropriate error messages are generated.

The Entity Relationship Diagrams (ERD) drawn in this assignment distinguishes between fundamental, associative, and attributive entities. The diagrams also make the notion of cardinality of relationships clear to the students. Cardinality is explained as the degree of relationship between two entities and the arrows connecting the entities distinguish between "one-to-many," and "many-to-many" relationships. A proper understanding of these constructs is important for the systems analysts to identify the weaknesses of the existing system and to make recommendations for correcting the problems. To support the system development phase of the SDLC, VAW can generate schema descriptions in structured query language (SQL) from the ERD's.

Workshop Assignment 3. This assignment is associated with the database design aspects of the SDLC. The process of data normalization is demonstrated by introducing an associative entity between two fundamental entities. The "key synchronization" function of VAW automatically migrates the primary keys of the parents to the children. This operation creates one-to-many associations between the parents and children entities. The differences between the pre-synchronization and post-synchronization structures of the entities are used to reinforce this concept. The information related to ERD's is stored in the repository and is used in the development phase of the SDLC to create SQL descriptions for the entities.

Workshop Assignment 4. Create a structure chart for a single process selected from the data flow diagrams. In addition to the hierarchy of the superior (invoking) and subordinate (invoked) modules in the process, these charts show the parameters passed between the processes. The assignment explains

- the differences between the data and control couples,
- conditional invocation of modules, and
- active versus passive modules.

A data module is included in the assignment to explain the difference between the active and passive modules of the process. The structure charts are generally associated with the design and development phases of the SDLC, and VAW uses these charts to generate programming language code in the development phase of the SDLC. When the “analyze” function of VAW is invoked for structure charts, the system calculates the “structural complexity” of the diagram. This metric depends on the number of invocation lines and of modules in the chart. Systems designers can use this coefficient to simplify a proposed design.

IV. PERSPECTIVE FROM THE WORKSHOP

Fourteen participants, mostly faculty members interested in teaching systems analysis and design with a CASE tool, attended the workshop. Some participants indicated that they have already been using the tool in their classes. These participants brought their valuable experience into the discussion and gave constructive suggestions to address the difficulties encountered with VAW in the classroom. Since the software is often bundled with textbooks, these instructors are using VAW in their systems analysis courses for drawing diagrams using short cases at the end of the chapters in the textbooks [Marakas, 2001]. Other participants indicated that they have been teaching systems analysis and design without a tool and would like to include CASE in their repertoire of teaching methods with a project-based approach. All participants were provided with personal computers with the educational version of VAW Release 7.5 to draw the diagrams included in the appendix of the paper. From the exercises demonstrated in the workshop, comments made by participants familiar with the system, and the questions asked by others, the following points emerged related to the strengths and weaknesses of the educational version of VAW.

STRENGTHS OF VAW (EDUCATIONAL VERSION)

Conference participants indicated that the student version of VAW 7.5 has the following advantages as a system analysis and design tool:

1. VAW has a user-friendly graphical interface and short learning curve. Essential commands and operations can be easily learned with a structured lesson in less than two hours. These features and observations, corroborated by other studies of the software, make it useful as a classroom CASE tool [Williams, 1996].
2. The software allows creation of 12 different types of diagrams, integrated though a repository. In addition to the data-flow, entity-relationship, functional-decomposition, and structure diagrams created in the project, it includes activity, class, collaboration, and state transition diagrams commonly used in systems analysis and design projects.
3. VAW allows a choice of four methodologies: Yourdon/DeMarco, Gane and Sarson, SSADM, and Métrica. For the workshop, Gane and Sarson methodology was selected, since it is commonly used in the United States and its symbols are most familiar to the business information systems community.
4. As an integrated tool, VAW is very useful in explaining to students the concepts of functional decomposition, top-down analysis, and process modeling with their interrelationships from a practitioner’s point of view. Although data flow, entity relationship, and decomposition diagrams can be easily generated with ordinary graphics packages such as MS-Draw, it is not possible to verify the consistency and correctness of these diagrams. The “analyze” function of VAW provides the facility for checking the syntax and, to some extent, the logic of these diagrams. The “spawn” function available with Functional Decomposition Diagrams creates data flow

diagrams that can be linked to the context and second level diagrams of the project. This was found to be an easier way of creating initial data flow diagrams for the process model.
5. Information contained in the repository can be used to automatically generate database schema for the project in the student version. The student version of the software allows thirteen choices of database formats for schema generation, from Access and DB2 to Oracle and Sybase SQL Server. Btrieve, the database format recommended by software vendor Visible Systems Corporation, was selected for the workshop.
6. Although the professional version of the software allows code generation in C, COBOL and SQL, the student version only provides for database schema generation in Structured Query Language (SQL). In the workshop exercise, functional decomposition diagrams were connected with data flow diagrams to demonstrate the linkages between project planning, process model, and the hierarchical decomposition concepts discussed in systems analysis and design courses.
7. The concept of “top-down” analysis is demonstrated in these diagrams by showing the equivalency between various levels of data flow diagrams, splitting data flows, and 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 demonstrated the repository functions and explained how the CASE tool can detect violations of rules contained in the repository.

SHORTCOMINGS OF VAW (EDUCATIONAL VERSION)

The following shortcoming of the educational version of VAW as a system-modeling tool were identified in the workshop:

1. Students can work with only one project at a time in the educational version of the software. This limitation makes it impossible to transfer data and diagrams between two different projects.
2. Workshop participants observed that the versions of the software supplied with the manuals are often different, and also the “current” versions supplied to different academic institutions are not the same. But the most difficult problem is that the projects created in a more recent version of the software cannot be opened with older versions. This incompatibility is a major source of annoyance for students who start the projects in the on-campus labs equipped with more recent versions of the software. Students often try to open these projects on their personal computers with older versions of the software obtained from their friends who took the course in the previous semesters.
3. Sometimes it is difficult to delete projects from the workspace. The users have to delete unspecified files from the VAW software directory through Window’s Explorer.
4. Some users encountered an exasperating problem with the naming of data flows in the process model. If an extremely short unnamed data flow is created unintentionally, such as a dot, it is almost impossible to detect and correct it in the project. These data flows are not easily visible on the screen and the system keeps giving the error message “There are one unnamed data flow[s].”
5. The educational version of the software does not support a multi-user environment through Local Area Networks (LAN). The professional version for networks, however, provides for extensive multi-user facilities. The users can be designated as system managers, project managers, or simple users. The network version allows multiple users to work on the project concurrently and modify diagrams in a project one at a time.

These shortcomings are not considered severe for small projects assigned to students in the first course in CASE and Systems Analysis and Design. The strengths and shortcomings of the educational version of VAW are summarized in Table 1.

Table 1: Strengths and Shortcomings of VAW (Educational Version)

Strengths	Shortcomings
User friendly interface	Single project allowed in the workspace
Short learning curve	Incompatibility of versions
Range of diagrams	Difficulty of deleting projects
Choice of four methodologies	Invisibility of short unnamed data flows
Integrated tool	Lack of multi-user support
Extensive repository functions	
Choice of database formats	
Database schema generation	
Syntax analysis for diagrams	

AVAILABILITY OF HELP FOR VAW

Extensive help is available online for the software. The help section is organized alphabetically according to the first letter of the error message generated by the system. In addition, an index is provided for a wide range of topics relevant to the software. The help section explains the sources and possible causes of errors with diagrams which students find extremely helpful. The other excellent source of help on the software is the operations manual published by Visible Systems Corporation [Visible Systems Corp., 2000b]. The tutorial supplied with the student version of the software contains a sample project that analyzes the functions of a department of motor vehicles [Visible Systems Corp., 2000a]. Unfortunately, the tutorial alone cannot be used as a textbook for the entire course because of its extremely limited content.

VALUE OF THE WORKSHOP FOR TEACHING CASE

The workshop made a useful contribution to the conference in three respects.

- First, it exposed prospective instructors of systems analysis and design courses to a functional instrument that can be integrated in their teaching portfolio to provide students hands-on experience with the process of systems modeling.
- Second, it brought into focus the need for comprehensive cases that can be used throughout the semester as a systems-modeling project in the course.
- Third, the discussion in the workshop among participants with various degrees of experience with CASE tools and system modeling concepts brought into perspective the advantages and disadvantages of using a particular tool such as VAW in the class room. It reinforced the argument that conceptual understanding of the discipline is more important than the tool and the CASE tools cannot be used as substitutes for the systems concepts taught in the classroom.

At least three participants of the workshop indicated that they intend to use the case analyzed in the workshop as a project in their courses. Several participants visited the Visible Systems Corporation booth at the AMCIS conference for further information on the availability of the software for educational purposes. This outcome is an indication of the usefulness of the workshop as a teaching instrument. The simulated case is more comprehensive than short cases generally provided at the end of chapters in the textbooks on systems analysis and design. Students often spend an inordinate amount of time in data gathering for systems projects. The simulated case used in the workshop can avoid this need. Students can thus have more time for practice with the tool and conceptual understanding of the system modeling constructs.

V. CONCLUSION

The assignments and the representative diagrams (context diagram, second-level data flow diagram, entity-relationship diagram, and structure chart) completed in the workshop are included in the Appendix. Because of the limitation of time, only the most significant parts of the project assignments were completed in the three-hour period allocated to the workshop. The discussion generated in the workshop addressed the strengths and shortcomings of VAW as a CASE tool, and comprehensive class projects as a pedagogical tool.

The workshop thus provided the participants a useful, hands-on experience with Visible Analyst Workbench. It gave an invaluable opportunity to those who are already using the tool in their classes to share their experiences with other participants. This experience can be used for teaching abstract concepts of systems analysis and design concepts with practical realities of system modeling with CASE tools more effectively. At the end of the workshop, a number of participants indicated that they would like to use the simulated case as a class project. Since I had developed the case for purely academic purposes, any faculty member interested in using it in their classes can freely share it. The appendix contains the partial solution to the case and the full solution can be obtained from the author.

Editor’s Note: This article is an expansion of the author’s workshop presented at AMCIS 2001. It was received on August 31, 2001 and was accepted on October 22, 2001. The article was with the author for approximately 7 weeks for four revisions. The article was published on February 27, 2002 together with the other articles in the special issue on the AMCIS 2001 Workshops.

REFERENCES

EDITOR’S NOTE: The following reference list contains the address of World Wide Web pages. Readers who have the ability to access the Web directly from their computer or are reading the paper on the Web, can gain direct access to these references. Readers are warned, however, that
1. these links existed as of the date of publication but are not guaranteed to be working thereafter.
2. the contents of Web pages may change over time. Where version information is provided in the References, different versions may not contain the information or the conclusions referenced.
3. the authors of the Web pages, not CAIS, are responsible for the accuracy of their content.
4. the author of this article, not CAIS, is responsible for the accuracy of the URL and version information.

Marakas, G. (2001) *Systems Analysis and Design: An Active Approach*, Upper Saddle River, NJ: Prentice Hall.

Visible Systems Corp. (2000a) *Visible Analyst Tutorial,* Waltham, MA: Visible Systems Corporation.

Visible Systems Corp. (2000b) *Visible Analyst Operations Manual,* Waltham, MA: Visible Systems Corporation.

Schmidt, A. (1995) *Working with Visible Analyst Workbench for Windows,* Upper Saddle River, NJ: Prentice Hall.

Williams, J. (1996) “Visible Analyst Workbench 6.0”, *DBMS on-line*, http://www.dbmsmag.com/9605d09.html.

APPENDIX. DETAILED ASSIGNMENTS AND REPRESENTATIVE DIAGRAMS

ASSIGNMENT 1 (EIGHT SECTIONS):

[Note: The underlined words represent the specific functions performed by VAW.]
(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 write 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. For the process you are exploding, select the most appropriate existing diagram from the three diagrams spawned in step (e) above, and connect the selected diagram as the child diagram to the exploded process.
(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 were given additional details of the situation context. They were 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.

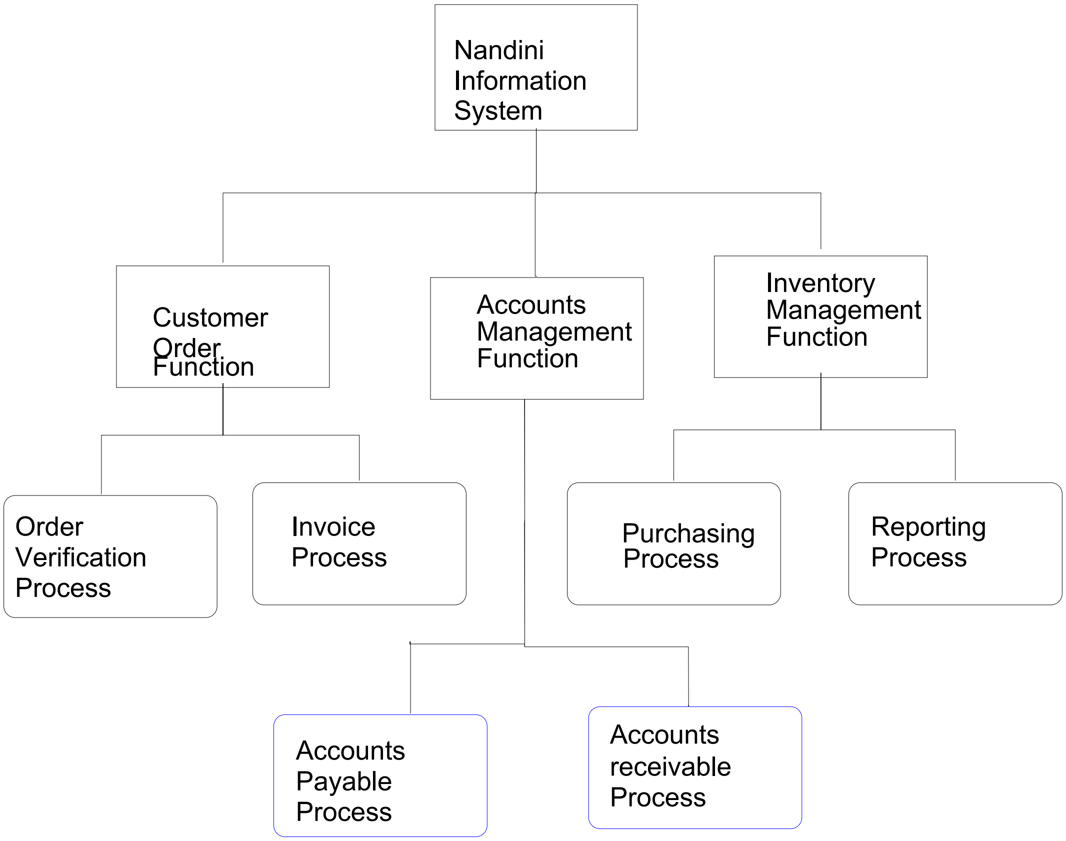

Nandini Information System
Customer Order Function
Accounts Management Function
Inventory Management Function
Order Verification Process
Invoice Process
Purchasing Process
Reporting Process
Accounts Payable Process
Accounts receivable Process

Figure A-1. The Functional Decomposition Diagram (for Assignment 1)

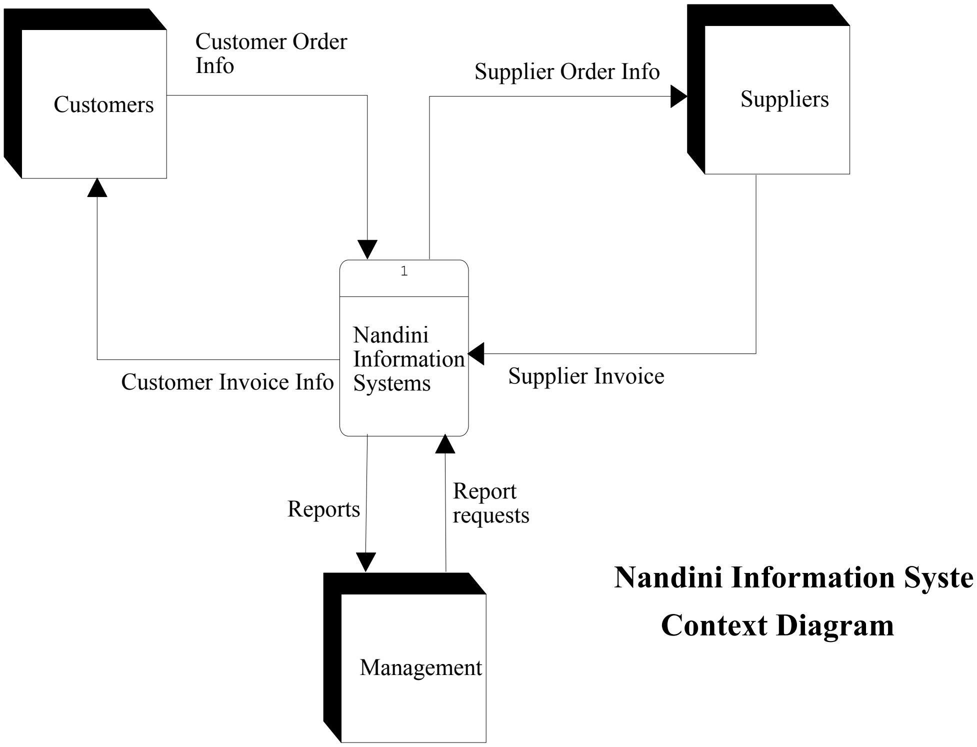
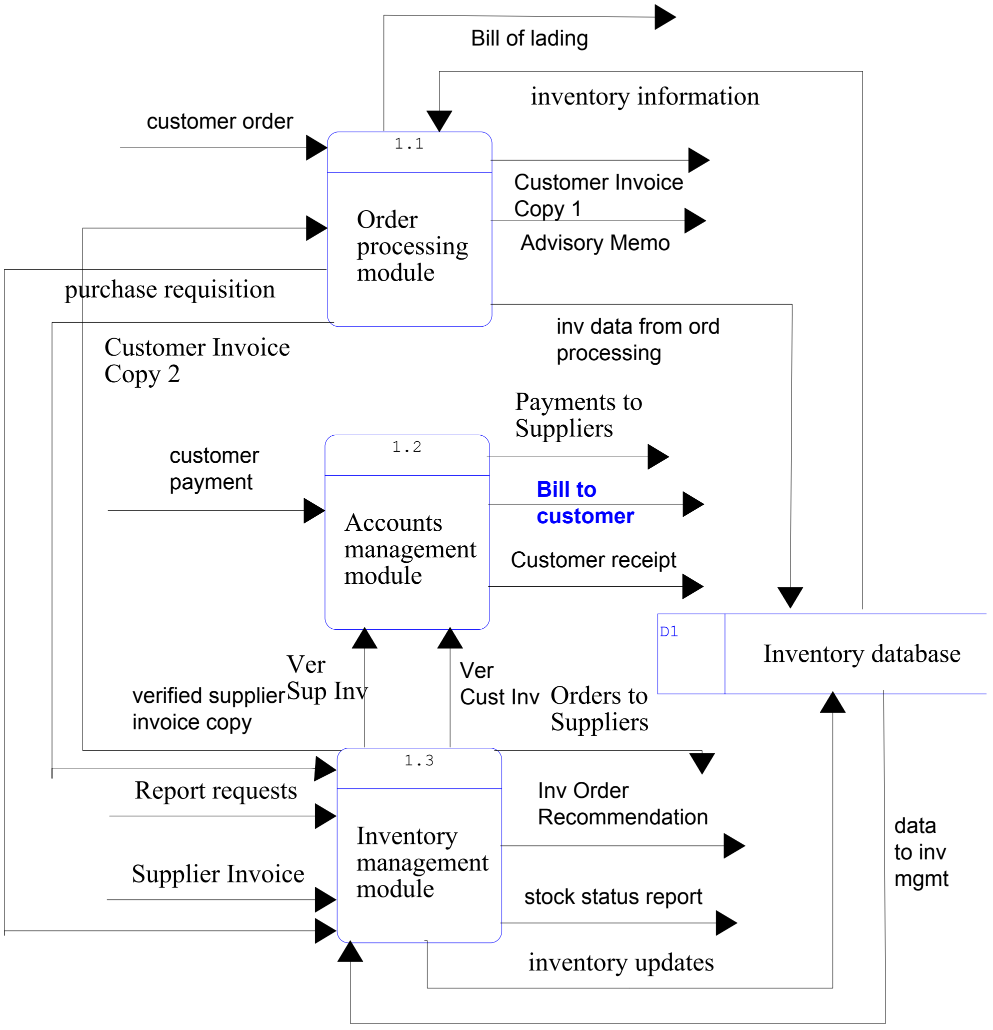


Figure A-2. The Context Diagram (Assignment 1)

ASSIGNMENT 2 (FOUR 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) 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.



NANDINI ENTERPRISES INFORMATION SYSTEM
LEVEL 2 DATA FLOW DIAGRAM (v2)

Figure A-3. The Second Level Data Flow Diagram (Assignment 2):

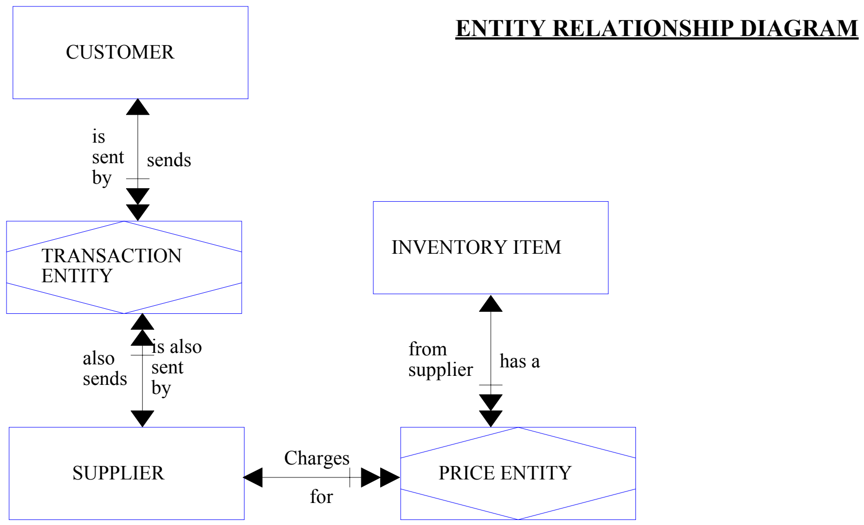


Figure A-4. The Normalized Entity Relationship Diagram (For Assignment 2)

ASSIGNMENT 3 (SIX 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) Define appropriate length and type (physical characteristics) for each data item in the model.

(f) 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.

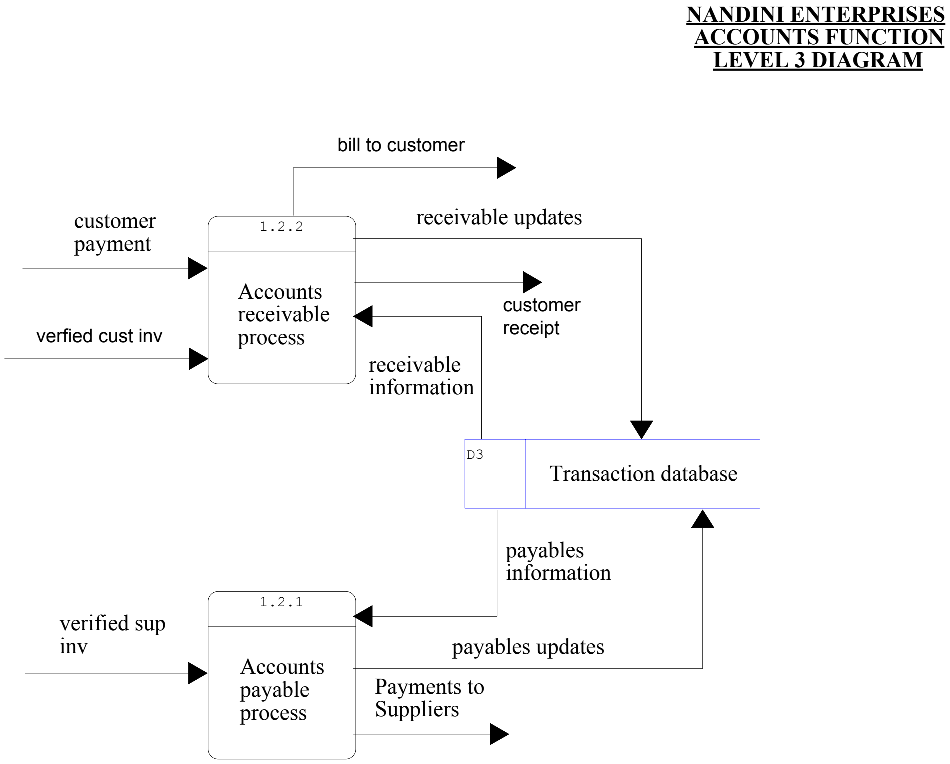


Figure A-5. A Representative Detailed Data Flow Diagram for the Accounts Management Process (For Assignment 3)

ASSIGNMENT 4 (TWO SECTIONS):

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

(b) Analyze the warning messages generated by VAW and explain the meaning of “structural complexity” of the diagram.

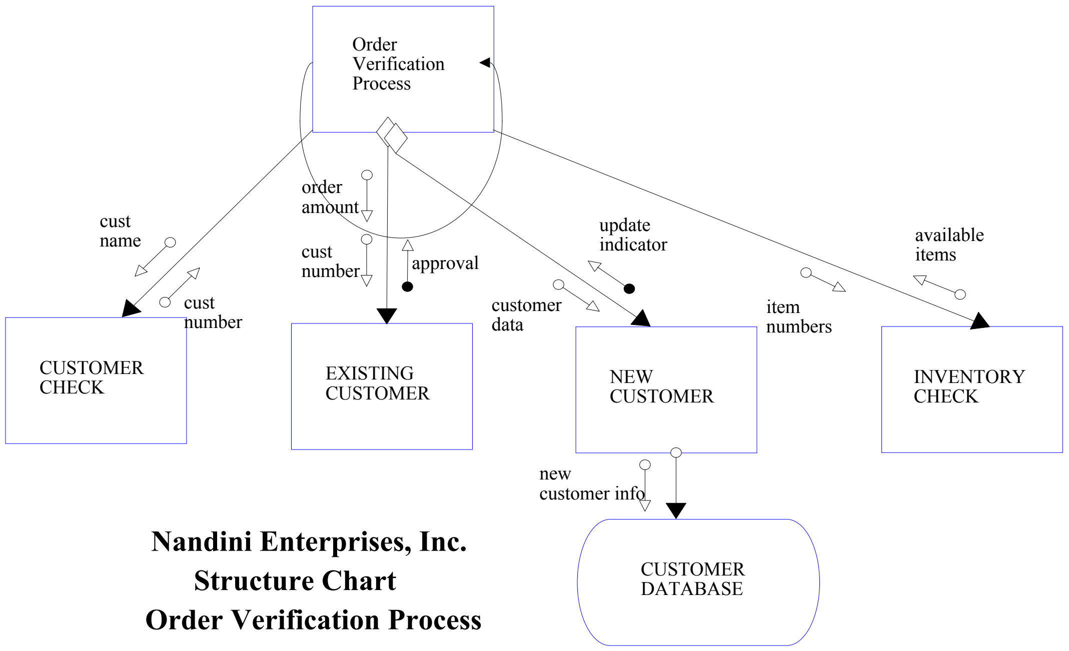


Figure A-6. The Structure Chart (For Assignment 4)

ABOUT THE AUTHOR

Satya Prakash Saraswat is professor of computer information systems at Bentley College in Waltham, MA, He holds Ph.D. and MBIS degrees from Georgia State University, an MBA from Murray State University, and BS (Chem. Eng.) from Harcourt Butler Technological Institute in India. Before joining Bentley, he taught information systems at San Diego State University and Georgia State University. His publications and areas of current research include global dimensions of information technology, information technology ethics, systems philosophy, and communications networks

Communications of the Association for Information Systems

ISSN: 1529-3181