Technical Report on

NAME: SANTU JANA
UNIVERSITY ROLL: 10700121127
COLLEGE ROLL:CSE/21/L-146
TITLE OF THE REPORT:

Trading-house Automation System

Subject: Software EngineeringCode: ESC-501

Table of Contents

Table of Contents

1. Introduction

a. Purpose:

The purpose of the document is to collect and analyze all assorted ideas that have come up to define the system, its requirements with respect to consumers. Also, we shall predict and sort out how we hope this product will be used in order to gain a better understanding of the project, outline concepts that may be developed later, and document ideas that are being considered, but may be discarded as the product develops. In short, the purpose of this SRS document is to provide a detailed overview of our software product, its parameters and goals. This document describes the project's target audience and its user interface, hardware and software requirements. It defines how our client, team and audience see the product and its functionality. Nonetheless, it helps any designer and developer to assist in software delivery lifecycle (SDLC) processes.

b. Overview:

The remaining sections of this document provide a general description, including characteristics of the users of this project, the product's hardware, and the functional and data requirements of the product. General description of the project is discussed in section 2 of this document. Section 3 gives the functional requirements, data requirements and constraints and assumptions made while designing the Trading House Automation System. Section 4 and 5 are for supporting information.

c. Environmental characteristics:

This section should briefly outline the environment (hardware and other software) with which the software will interact.

i. **Hardware:** CPU, Monitor.

ii. **Peripheral:** No.

iii. **People:** The report is done by people Admin, staff authority,Customers.

2. Functional Requirements

1. TRADING HOUSE AUTOMATION SYSTEM:

1.1 Process the order.

1.2 Handle query using sale statistics.

1.3 Handle indent request.

1.4 Accept the order from the customer with checking his previous history.

2. MANAGER:

2.1 Send the customer query to the Trading house automation system.

2.2 Get the statistics from Trading house automation system.

3. CUSTOMER:

3.1 Order will be sent to the Trading house automation system.

3.2 Takes the response from Trading house automation system.

4. PURCHASE DEPARTMENT:

4.1 Generate the indent.

3. Case Study

Problem Statement

<< Draw the context Diagram and level-1 Data Flow Diagram. >>

Solution

The context diagram and Level-1 DFD for the trading house automation problem is shown in below.

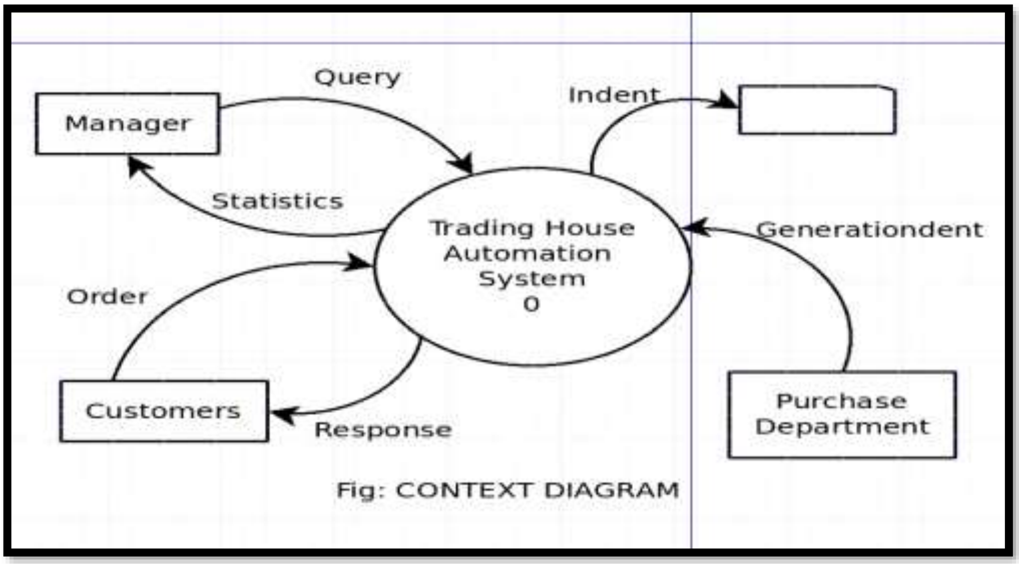

Fig: CONTEXT DIAGRAM

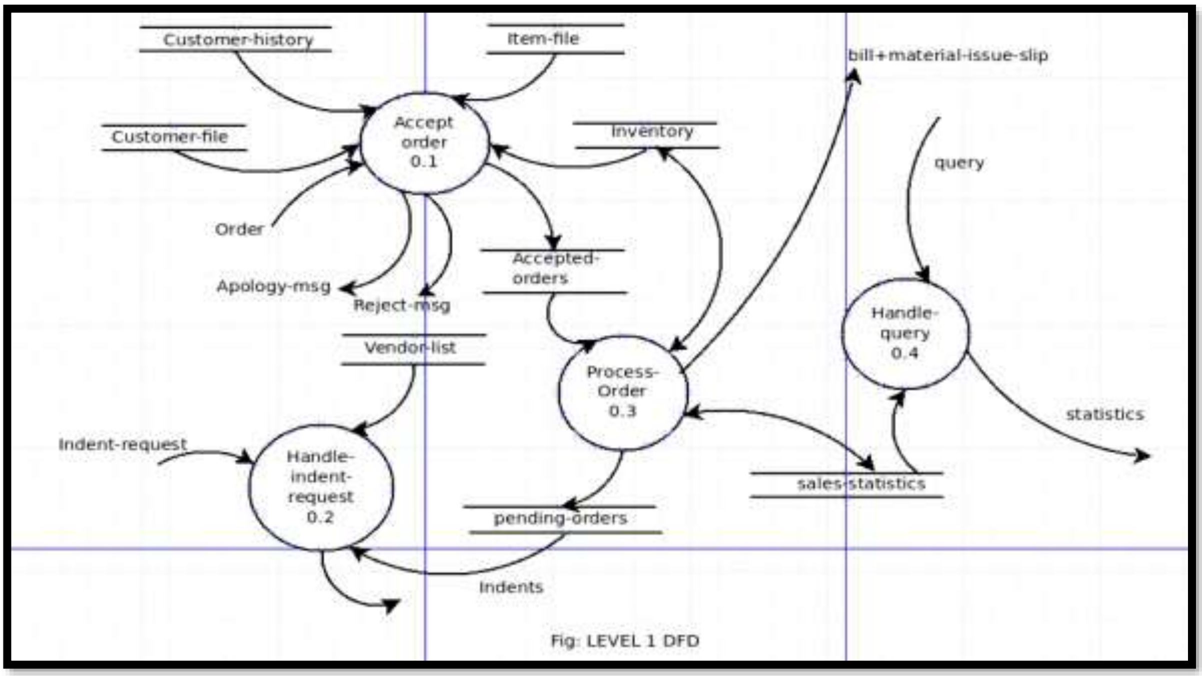

Fig: LEVEL 1 DFD

4. STRUCTURE CHART

By observing the level 1 DFD of TAS, we can see that the data input to the diagram are handled by different bubbles and therefore transaction analysis is applicable to this DFD. Input data to this DFD are handled in three different ways (accept-order, accept- indent-request, and handle-query), we have three different transactions corresponding to these as shown in below figure.

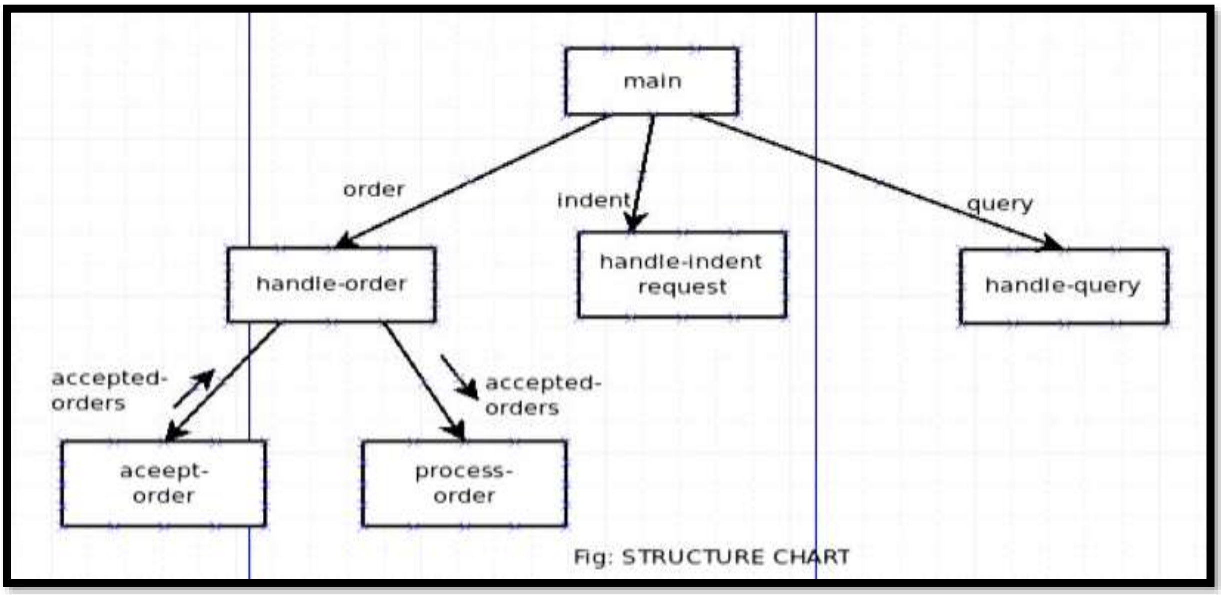

Fig: STRUCTURE CHART

5. Conclusion

The following observations can be made from DFD diagram of Trading-house Automation System.

1. In a DFD, if two data stores deal with different types of data, e.g. one type of data is invariant with time whereas another varies with time, (e.g. vendor address, and inventory data) it is a good idea to represent them as separate data stores. The inventory data changes each time supply arrives and the inventory is updated or an item is sold, whereas the vendor data remains unchanged.
2. If we are developing the DFD model of a process which is already being manually carried out, then the names of the registers being maintained in the manual process would appear as data stores in the DFD model. For example, if TAS is currently being manually carried out, then normally there would registers corresponding to accepted orders, pending orders, vendor list, etc.
3. We can observe that DFDs enable a software developer to develop the data domain and functional domain model of the system at the same time. As the DFD is refined into greater levels of detail, the analyst performs an implicit functional decomposition. At the same time, the DFD

refinement automatically results in refinement of corresponding data items.

4. The data that are maintained in physical registers in manual processing, become data stores in the DFD representation. Therefore, to determine which data should be represented as a data store, it is useful to try to imagine whether a set of data items would be maintained in a register in a manual system.

6. References

[1] Fundamentals of Software Engineering by Rajib Mall