Process Modeling

Prepared by Salvatore T. March
For Classroom Use Only

Introduction

In information system development, two different types of processes are modeled, *business processes* and *information system processes*. Business processes are modeled using Workflow diagrams focusing on behavior (events and activities). Such diagrams show the actors or roles within an organization and represent the actions performed and the interactions among those actors and the flow of work, including physical materials, required to accomplish a business task. Information system processes are modeled using Data Flow Diagrams (DFDs). These show the input of data into information system processes from their appropriate sources (external entities), the transformation and storage of this data, and the output of processed data to its appropriate target (external entities). Both types of process models are discussed below.

Business processes are modeled to identify bottlenecks and inefficiencies in the business processes and opportunities to address these via the reorganization or work and the application of information technology. Business process modeling is at the heart of business system reengineering. Information system processes are modeled to define the role that the information system plays within business processes, to specify its interfaces for capturing data, its major functions and the rules within them for transforming data, the contents of its persistent data and its interfaces for producing information outputs.

Business Process Modeling

Business processes are represented using workflow diagrams. These document what activities must be performed in response to events that occur and what responses result from those activities. The purpose is to identify bottlenecks and inefficiencies so they can be reengineered to make them more effective and more efficient, typically by applying information technology in some manner. It is critical, when modeling business processes, to focus on identifying how information technology can be used to reengineer the processes rather than identifying tasks within the current process that can be automated. That is, the goal of business process modeling is to identify opportunities for information technology to enable the business organization to be more productive, to enhance its ability to service its customers, or in some way improve the way in which it does business.

The basic constructs used to model business processes are icons that represent "agents" in the business and in its environment that:

1. cause activities to be performed, or
2. impose rules on the way in which the business performs activities, or
3. require being informed of the occurrence or results of activities, or
4. in any other way interact with

the business process under consideration. These include employees, managers, other information and mechanical systems, other organizational units, stakeholders within the business, customers, government, stockholders, etc.

Business process modeling is closely related to Behavior modeling which focuses on identifying the events to which the business must respond, activities performed in order to responses to those events and conditions governing those activities and responses. A business process often begins when an external "agent" initiates an event to which the business must respond. The business is informed of the event and takes appropriate action. Business process modeling traces the actions performed in order to respond to that event.

Consider, for example, the event, Sales Order Placed. It is initiated by an "agent" external to the business, commonly referred to as a Customer. A business process model, therefore, begins with an icon, a stick figure, representing the Customer and a flow representing the order (Figure B.1). Modeling the business process performed to respond to this event simply traces what occurs in the business, beginning with the best case scenario, when "nothing goes wrong" and ending with the worst case scenario when "everything goes wrong." UML refers to such scenarios as Use Cases.

Crucial to developing such a representation is identifying people within the organization who know what occurs and what can go wrong and who have the authority and capability to modify those business processes. Just as crucial is the ability of the analyst to work together with those people to determine how information technology can be used most effectively in this business situation.

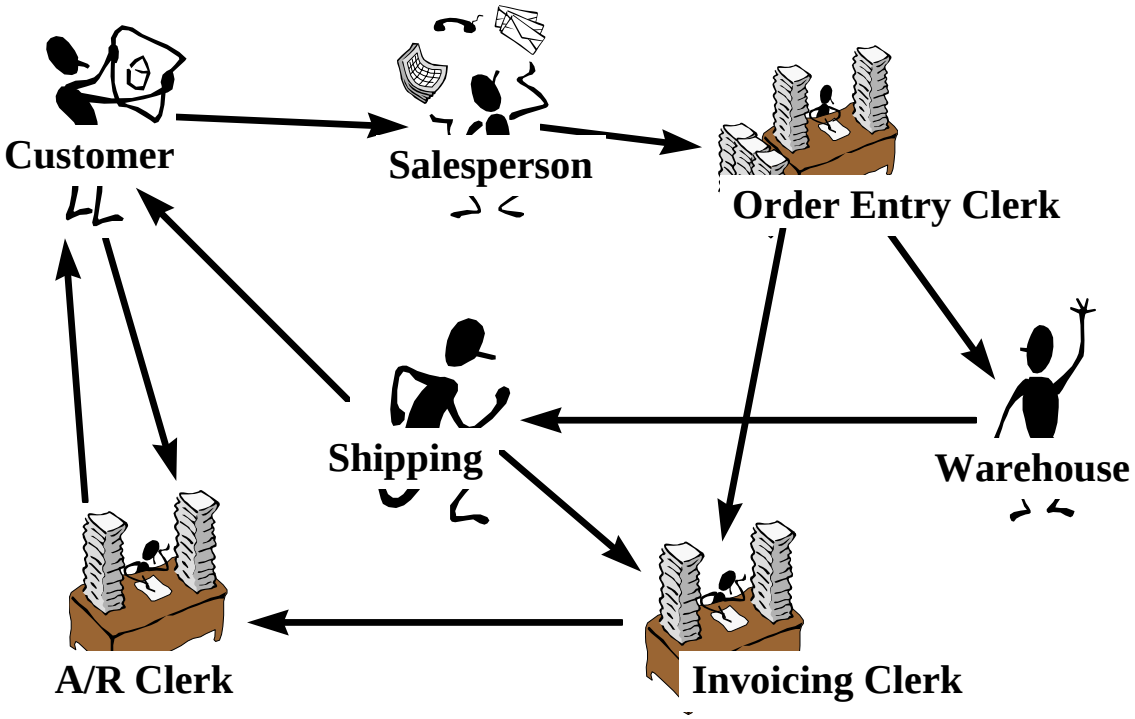

Figure B.1 A Business Process Model of Sales Order Processing

Figure B.1 illustrates the case when "nothing goes wrong" for a distribution company that utilizes its own sales force, ships from inventory and makes its own deliveries. A Customer

interacts with Salesperson who determines the appropriate products to meet the Customer's need and negotiates appropriate pricing and a delivery date. The Salesperson informs the Order Entry Clerk who enters the order into the current system and sends a picking ticket to the Warehouse. The Warehouse personnel pick the order and place it on the shipping dock with the picking ticket filled in with the quantities picked. The Shipping personnel load the order onto a truck and deliver it to the customer, giving a copy of the picking ticket, now referred to as a shipping ticket, to the Customer and returning a signed copy of the picking ticket to the Invoicing Clerk. The Invoicing Clerk produces an invoice, using the negotiated prices as specified on the original order and the quantities shipped as specified on the returned shipping ticket. The invoice is given to the A/R Clerk who sends a copy to the Customer and retains a copy for accounting purposes. When the Customer sends a payment, the A/R Clerk applies it to outstanding invoices, keeping a "open item" accounting system. Periodically, the A/R Clerk sends statements to Customers showing the status of their accounts.

Beginning with a business process representation of this type enables the analyst to work with people within the organization to assess the potential benefits of information technology for that business process. A framework for such an analysis, adapted from [Wetherbe, 1991] is presented below. The framework contains a series of questions that an analyst can use to determine if and how information technology can add value to the organization. Answers to these questions provide valuable insight into determining the data and processing requirements for an information system to support this business process.

Functional Analysis Questions
For each icon in the diagram, ask the following questions (some will be more relevant than others for any given icon):

1. what is the *mission*?
2. what *tasks* are being performed?
3. what *problems* are encountered or have recently been encountered in accomplishing those tasks? What are potential *solutions* to those problems?
4. what *decisions* are made?
5. what are the *critical success factors* for those tasks?
6. what measures are used to evaluate how well those tasks are being performed?
7. who else must be informed of the execution or results of those tasks?

Given the answers to these questions, the analyst can then ask, "What *information* is needed:"

1. to accomplish the mission?
2. to perform those tasks?
3. to solve those problems or implement those solutions?
4. to make those decisions?
5. to evaluate those identified critical success factors?
6. to measure performance?
7. by those who need to be informed?

The information content identified must then be decomposed into base *data values* that need to be acquired and the *processing rules* by which to transform it into information. Finally, appropriate sources and data acquisition techniques must be identified. Once these are

accomplished, the analyst can begin to re-engineer the processes with the stakeholders and recommend information system capabilities needed to support these processes.

Consider applying the above framework to the Sales Order Processing model of Figure B.1. Depending on the type of business and products sold, it may or may not be feasible to address these questions directly to Customers. If not, a surrogate must be identified to whom the questions can be addressed. Likely this would be a person from the Marketing function with a good understanding of the Customer perspective. Furthermore, the analyst must determine if all Customers can be considered to be homogeneous for purposes of the analysis or if different Customer segments must be analyzed separately.

From the Customer perspective, the mission is to acquire the most appropriate products for their business situation at a good price, to be delivered in an appropriate timeframe. Tasks necessary to accomplish that mission are: determine what products are on the market, their characteristics, their appropriateness, their cost, available suppliers, their reliability, their delivery and service capability, their reputation, the potential to purchase from a customer or establish a strategic alliance with the supplier.

Problems that might have been encountered include poor quality or overpriced purchases, late or improper deliveries, inadequate supplier payment terms or response for returns and allowances. Solutions include finding a high quality supplier and negotiating appropriate pricing and terms. The decisions are what to purchase and from whom. The Customer's critical success factors include acquiring appropriate products at a fair price, measured by how well the products perform and the availability of comparable products at a lower cost. Answers to these questions begin to overlap. This is expected and appropriate. They are designed to provide different perspectives on the information system requirements for a business process. What one person might view as a problem, for example, another views as a decision. The point is to crystallize the information system requirements within the minds of the people within the business and to assure that these are as complete as possible.

Following an analysis of this type, a supplier of industrial insulation determined that their customers had difficulty determining what products to purchase and as a result often returned high quality, but inappropriately ordered products. To make this decision, their customers needed to know, the surface temperature, heat loss, product and installation costs and heating cost savings for various insulation products for a given project. Many of their customers lacked the sophistication and staff to perform these calculations. Hence, they determined that they should develop a computer application that salespeople could take directly to their customers, run an analysis that provided this information and generate an order directly from the results. Thus, what began as an analysis of sales order processing resulted in an application of information technology that added value to their products and gave them a competitive advantage.

Information System Process Modeling

Given a well specified business process and the information requirements to support that process, Data Flow Diagrams (DFDs) are a means to represent the data acquisition,

transformation and storage and the information delivery processes within an information system. DFSs have four major constructs: External Entities, Data Flows, Processes, and Data Stores. Each is discussed below.

External Entities represent the people or external systems with which the current (or proposed) system must interact to acquire its input data or to which the current system must deliver its output data. External Entities typically represent the "responsible party" for the data flow. For example, while a payroll clerk may actually enter the employee hours worked into a payroll system, the employee is the *responsible party* for the data and would be a better choice for the External Entity from which this data flow occurs.

Data Flows represent the data content acquired from an External Entity, output to an External Entity, required as input to an information system Process or produced as output from an information system Process. Data cannot be transformed in any way within a Data Flow. A Data Flow is simply a "pipeline" of data. For example, the data on an employee's timecard could be represented as a Data Flow named, Timecard. The data elements included in each data flow must be documented in a data dictionary associated with the data flow. The timecard data flow, for example, likely contains data elements such as: employee number, employee name, period ending, and hours worked.

Processes transform data according to business rules. Processes in a DFD are strictly computer implemented processes. The business rules and processing logic must be documented for each process. All outputs produced from a Process must be calculated from its inputs according to its processing logic, ultimately expressed in a computer programming language, but initially expressed using a process narrative, Structured English, decision tables or decision trees. Processes that are too complicated to be easily described in this way are decomposed or *exploded* into 5 to 9 processes at the next lower level of detail. Explosions continue until all the logic of all processes can be described using Structured English, Decision tables, or Decision trees. Processes transform data, they do not create it. A Process must have sufficient input to calculate each of its outputs. Furthermore, the description of its processing logic must be sufficient to develop a procedure in a computer programming language to implement it.

A Process used to calculate FICA withholding taxes, for example might be described by the following processing logic:

```
IF YtdGrossPay + CurrentGrossPay <= MaxFICAWages
THEN
    ficaTax = CurrentGrossPay * FICATaxRate
ELSE
    IF YtdGrossPay <= MaxFICAWages
    THEN
        ficaTax = (MaxFICAWages - YtdGrossPay) * FICATaxRate
    ELSE
        ficaTax = 0
    END IF
END IF
```

This processing logic implies that the process must have as input: YtdGrossPay, CurrentGrossPay, MaxFICAWages, and FICATaxRate. Its output is ficaTax. Processes should be named with an action phrase, e.g., Calculate FICA Tax.

Data Stores are Data Flows "at rest." Data Stores can be viewed as an "inventory" of data that lives within the system when no processes are active. Often they are used to "decouple" processes over time. For example, if employee fill out and submit timecards each day, but payroll is produced only once per week, the information system may show a process to enter and validate timecard data which outputs the data into a Timecard Data Store. Another process inputs data from that data store to produce paychecks, update payroll records, and produce required payroll reports. Data Stores do not transform data, they are simply repositories. Exactly the same data must flow out of a Data Store as flows into it. As with Data Flows, the contents of each Data Store must be documented in a data dictionary.

DFDs are "leveled" to show increasing detail of these transformation processes. The top few DFD levels typically represent the top few menus in the information system application. Both are designed to identify the organization of functions within the system. There exist numerous CASE tools to assist in producing and analyzing DFDs. These are often combined with code generators to produce menus, screens and algorithms for the system implementation.

At the highest level, termed a *Context* diagram, the information system is represented by a single process. The Context diagram is a *black box* view of the system focusing on its interactions with the environment, that is, its users, represented as External Entities. The Context diagram defines what data will be input to the system and what data will be output. It represents a contract with the client. It essentially says, "I will develop software to capture the inputs and transform them into the outputs shown. The system will not capture any inputs or produce any outputs not shown." It is the analyst's job to be sure that the outputs shown on the Context Diagram are sufficient to meet the needs of the client and that the inputs shown on the Context Diagram are sufficient to produce them.

Figure 1 shows a Context (or Level 0) Data Flow Diagram for a simplified Payroll System. The process is named, Produce Payroll. Two External Entities are specified, Employee and Payroll Manger. Six data flows are specified, Timecards, Pay Checks, W2 Forms, Employee Fixed Data, Tax Tables and Parameters, and Payroll Reports. Employee provides Timecards and receives Pay Checks and W2-Forms. Payroll Manager provides Employee Fixed Data and Tax Tables and Parameters and receives Payroll Reports.

The contents of each data flow must be defined in a data dictionary. Figure 2 shows the data dictionary for the six data flows included in the Context Diagram. Employee Fixed Data includes employee number, name, social security number, and wageRate. Although the source of data such as name and social security number actually come from the employee, the Payroll Manager is deemed to be the "responsible party" for this data. Similarly, the Federal Government is the source of Tax Tables and Parameters (e.g., MaxFICAWages, FICARate), however, again, the Payroll Manager is deemed to be the "responsible party." Note that at this stage in the analysis, the contents of the data flow, Payroll Reports, has not been defined. This informs the

development team that analysis must be done to determine the exact requirements. General areas for investigation such as General Ledger and Federal and State reporting requirements are noted.

If the analyst can describe the processing logic and business rules to transform each input into each output, then the process analysis is complete. Typically, however, the context process should be "exploded" into five to nine subprocesses, each of which accomplishes some part of the transformation. Two techniques are typically used to explode a process, **Transaction Analysis** and **Functional Decomposition**. In Transaction Analysis, each output data flow is analyzed as follows. First a process is created to produce that output. The input data flows needed to produce that output are determined and then a source must be determined for each of these inputs. If the source is not an External Entity, then a process must be created to produce that data flow and the process is repeated. At this point data stores are introduced to hold data that "lives" within the system, beyond being used by a process. In Functional Decomposition, five to nine major subfunctions are identified and tied together using data flows. Frequently Transaction Analysis is used as a means to validate a Functional Decomposition.

Figure 3 illustrates an explosion of Process Payroll into five subprocesses. Enter Timecards, Produce Pay Checks, Produce W-2s, Maintain Tax Data, Produce Payroll Reports, and Update Employee Fixed Data. Typically the first explosion represents the menu choices for an application. That is, the major functions the user can initiate. From a Transaction Analysis perspective, the output Data Flow, Pay Checks is produced by the process, Produce Pay Checks. To produce that output requires input data flows containing Timecard data, Employee Check Data, and Tax Tables and Parameters. Each input data flow must have a source. In this case each comes from a data store, since it is deemed to be "long lived." The input data needed by a process is determined by developing the logic and business rules needed to produce the output, as illustrated in the processing logic above to calculate FICA Tax, one of the data items in the Pay Checks Data Flow. If the processing logic is too complex, that process must be exploded.

Data does not magically appear in data stores, however, so each data store must have an input. Timecards are entered into the Timecard Data Store by the process Enter Timecards which gets its inputs from the External Entity, Employee and from the data store Employee. Enter Timecards must insure that only Valid Timecards are entered, hence, it must validate the Employee Number on the Timecard against the Employee Number and Status maintained in the Employee Data Store. Data is input into the Employee Data Store by the process Update Employee Fixed Data which obtains its input from the External Entity, Payroll Manager. Similarly, Tax Tables and Parameters are obtained from the Tax tables Data Store, maintained by the process, Maintain Tax Data, which obtains its input from the External Entity, Payroll Manager.

The Process, Produce Pay Checks produces two additional outputs, YTD for the Employee Data Store, to update year to date gross pay and tax withholdings, and Pay Checks to update Pay Check History. Presumably the Data Flow, Payroll Reports requires a record of historical paychecks.

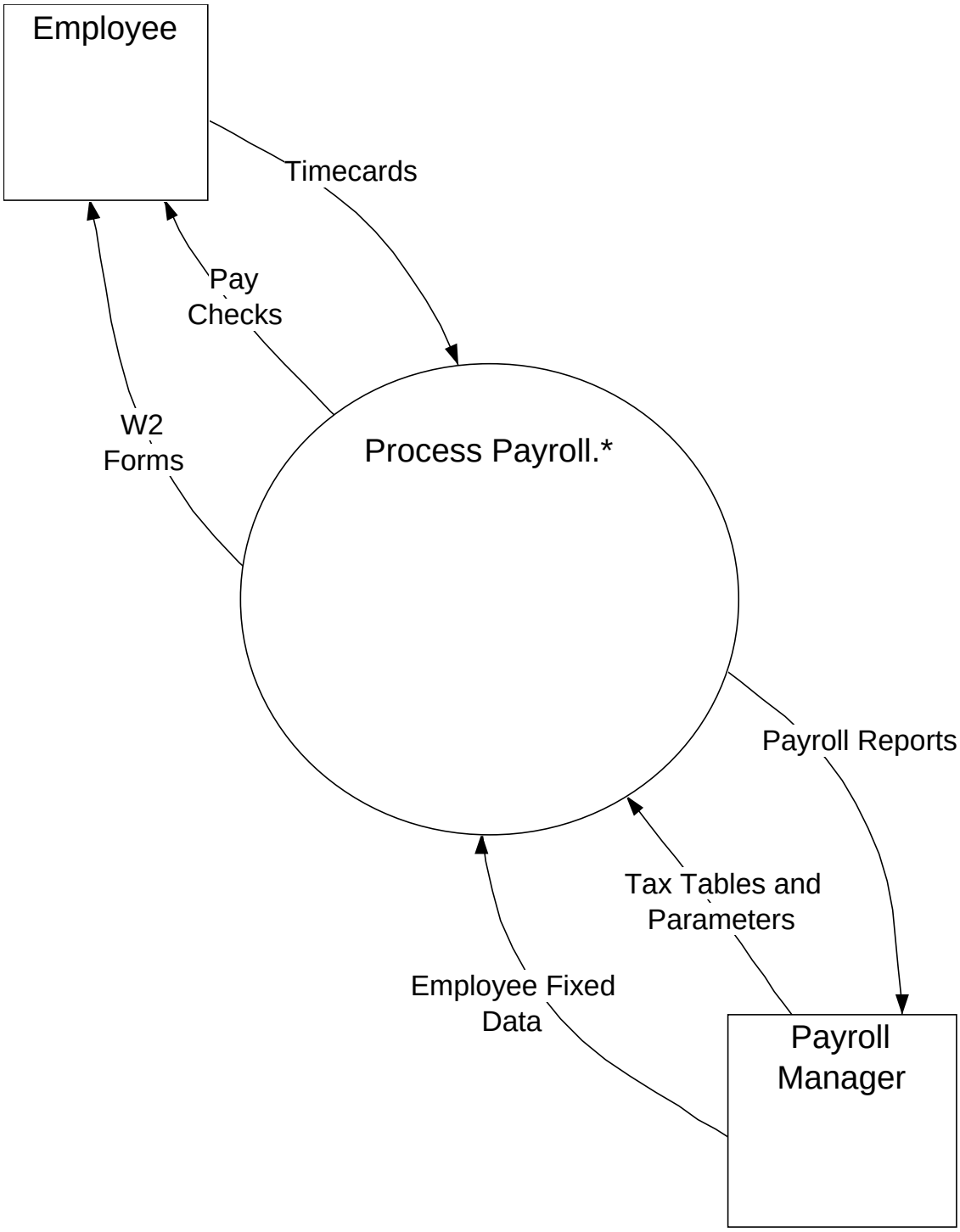

Figure 1. Context Diagram for Payroll System

Data Flow: Timecards
Contents: EmpNo, Hours Worked, Date Worked

Data Flow: Pay Checks
Contents: Check Number, Check Date, EmpNo, EmpName, SSN, Address, Tax State, Gross Pay, Fit Tax, FICA Tax, State Tax, FUTA Tax, Net Pay, YTD Gross Pay, YTD Fit Tax, YTD FICA Tax, YTD State Tax, YTD FUTA Tax, YTD Net Pay

Data Flow: W2 Forms
Contents: Control Number, EmpNo, EmpName, SSN, Address, Tax State, YTD Gross Pay, YTD FICA Gross Pay, YTD Fit Tax, YTD FICA Tax, YTD State Tax, YTD FUTA Tax, YTD Net Pay, SUBTOTAL Gross Pay, SUBTOTAL Fit Tax, SUBTOTAL FICA Tax, SUBTOTAL State Tax, SUBTOTAL FUTA Tax, SUBTOTAL Net Pay

Data Flow: Employee Fixed Data
Contents: EmpNo, EmpName, SSN, Address, Tax State, Pay Rate, Federal Exemptions, State Exemptions, Extra Fit Withholding, Extra State Withholding, Marital Status

Data Flow: Tax Tables and Parameters
Contents: MaxFICAWages, FICARate, FitPerExemption, FitRate, MaxFUTAWages, FUTARate, State, StatePerExemption, StateRate

Data Flow: Payroll Reports
Contents: To be determined. Need to obtain specifications for General Ledger interface, Federal and State reporting requirements, and Management Reports. These may require additional input parameters. At least W2 Form data must be available.

Figure 2. Data Dictionary for Context Data Flows

Analyze the Data Flows, Data Stores, and Processes in this DFD as follows. All data items in the input Data Flows to a Process should be used in the logic of that Process. All data items in the output Data Flows from a Process must be calculated from the data items in the input Data Flows. Only data items in the input Data Flows of a Data Store may be included in its output Data Flows. An obvious violation of these rules occurs in the Employee Data Store. It has an output data item called Status, however, there is no such input data item. If needed by the processing logic of Enter Timecards, then this data item must be input to the Data Store by the Process, Update Employee Fixed Data, which must obtain it from the External Entity Payroll Manager.

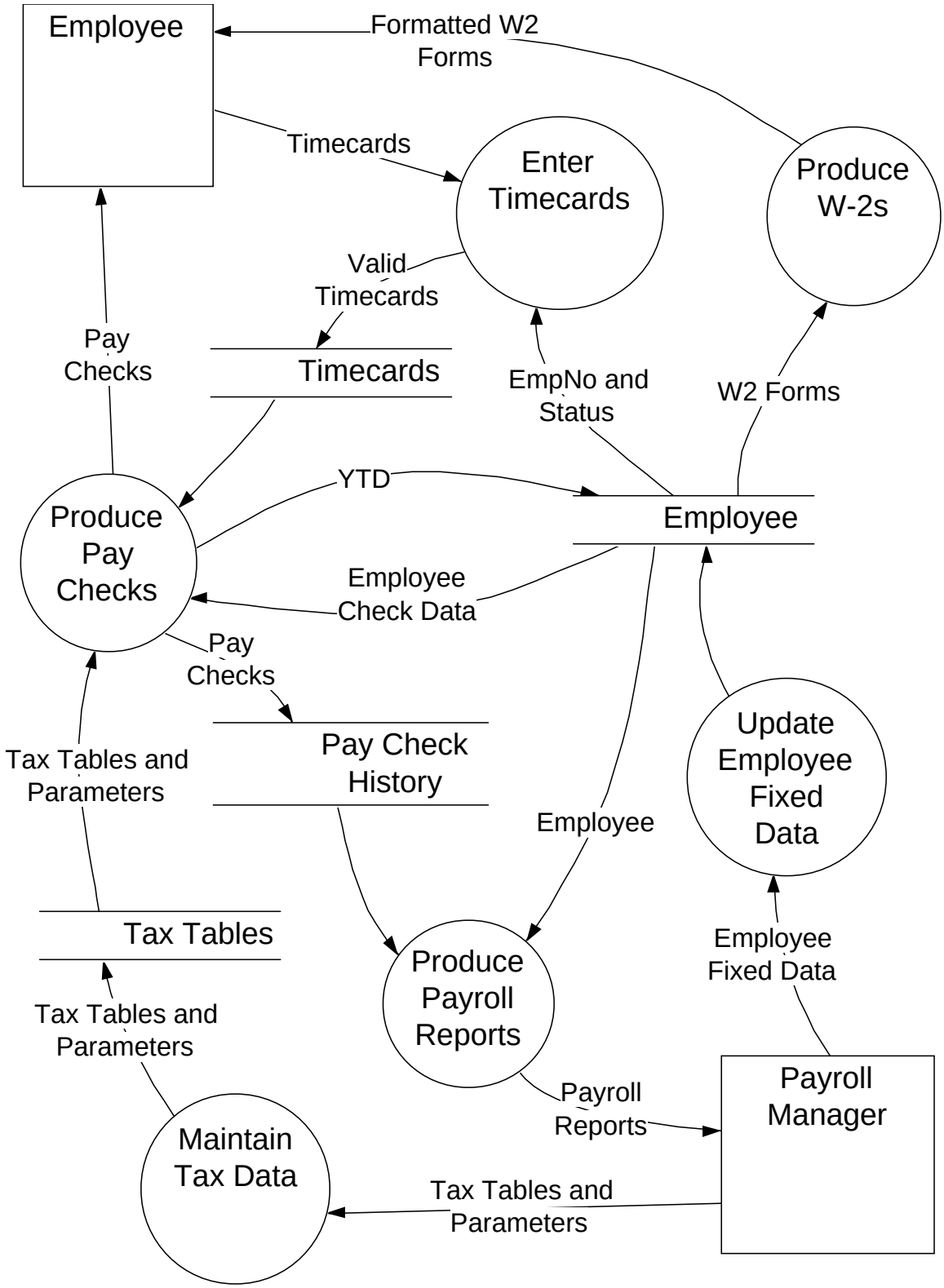

Figure 3. Explosion DFD for Process Payroll

Some Processes, such as Maintain Tax Data, are extremely simple and require only a brief description such as, “Allow the user to enter values for the federal tax parameters, MaxFICAWages, FICARate, FitPerExemption, FitRate, MaxFUTAWages, FUTARate. Be sure that all values are positive numbers and that FICARate and FUTARate are each less than 20% and that FitRate is less than 50%. Allow the user to choose or enter a new Tax State and enter its StatePerExemption, StateRate. Each must be positive and StateRate must be less than 10%.”

Other Processes, such as Produce Pay Checks, are complex and may require several levels of explosion before succinct process descriptions and business rules can be written. There are no rules for the number of levels for a DFD. Continue exploding until the process can be easily described. Processes that produce reports can be described using SQL to eliminate the need to provide detailed algorithms for grouping and sorting. Report formats should be described separately. Frequently it is easier to provide a prototype report rather than a description.