

## **Encoded Data Systems Ltd.**

Job Planing and Inception Software  
(JPIS)

Design Specification  
DS001

Author: T. Adam (Design Engineer)

Date of Creation: 15<sup>th</sup> January 2005

Revision: 0.2 (draft)

**Revision History**

<b>Revision</b>	<b>Created</b>	<b>Date</b>	<b>Comments</b>
0.0	T. Adam	<b>15/01/2005</b>	<b>First Draft</b>
0.1	T. Adam	<b>17/01/2003</b>	<b>Update of Revision 0.0 (changed function specs 1.1)</b>
0.2	T. Adam	<b>21/01/2003</b>	<b>Draft</b>

**Table Of Contents**

<b>1. Introduction:</b>	<b>Page 4</b>
<b>2. Architectural Design:</b>	<b>Page 5</b>
<b>3. Function Specifications:</b>	<b>Page 6</b>
<b>Appendix A:</b>	<b>Page 9</b>
<b>Appendix B:</b>	<b>Page 18</b>
<b>Appendix C:</b>	<b>Page 20</b>

## 1. Introduction

This document represents the design process specified in the encoded data systems (PPFA/JP1/01/RS).

This specification is in response to the modeling of the encoded data system's inception and job planning computer only. As such, the system comprises of the following:

- Encoding of Data
- Scheduling of Events
- Generating Reports
- Batch production of Products

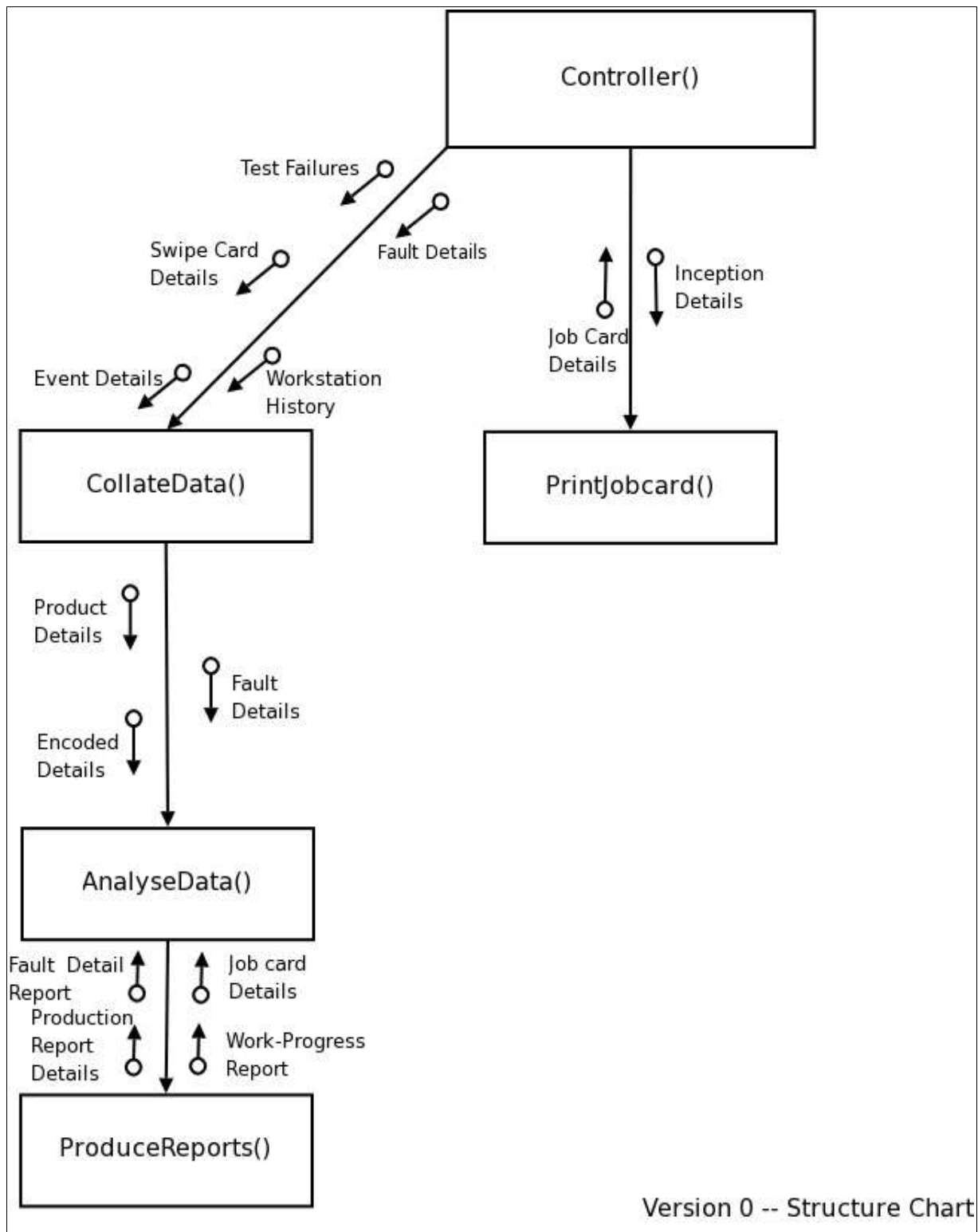
This specification shows:

- Architectural design of the software components
- Detailed design for each component

Behavioural models and all other models have been appendicised .

## 2. Architectural Design

The following diagrams show the modules which comprise the software system:



### 3. Function Specifications

The following section details the component design specifications:

Function: controller()	Function ID: 0
Author: T. Adam	Date: 20/01/2005
Version: 0.0	
Parameters: faultDetails, testfailures, eventdetails, workstationHistory, inceptiondetails	
Returns: Void	
Description: <b>Controller function for the overall Inception computer. When data is received to it, calls the appropriate functions based on the data.</b>	
Data Structures: None	
Detailed Design:	
<pre> WHILE there is data   DO     IF Data is faultdetails OR testfailures OR eventdetails OR workstationhistory       CALL collateData()     ELSE IF Data is InceptionDetails       CALL printJobCard()   END WHILE         </pre>	

Function: collateData()	Function ID: 2
Author: T. Adam	Date: 20/01/2005
Version: 0.0	
Parameters: testfailures, faultdetails, eventdetails, workstationhistory, swipecarddetails	
Returns: encodedData, batch details, product details	
Description: <b>Collatedata() is responsible for collecting the information passed to it from controller() and then acting upon that data. The collatedata() function calls AnalyseData(). Although this function calls two lower-level functions, DecodeEvent() and EncodeProduct ().</b>	
Data Structures: ProductBatch file store.	

Detailed Design:

WHILE there is data

DO

IF Data is eventdatails or workstationhistory

CALL DecodeEvent()

ELSE IF Data is swipecarddetails, testfailures, eventDetails

CALL encodeData()

END WHILE

Function: AnalyseData()	Function ID: 3
Author: T. Adam	Date: 20/01/2005
Version: 0.0	
Parameters: EncodedData, BatchDetails, ProductDetails	
Returns: JobCardDetails, work-progress details, ProductionReportDetails, FaultDetail Report	
Description: <b>AnalyseData() is responsible for producing reports based on the data presented to it.</b>	
Data Structures: ProductBatch file store.	
Detailed Design:	
READ FROM ProductBatch file store	
Get EncodedData, BatchDetails, ProductDetails	
ProduceReport()	
END READ	

Function: PrintJobCard()	Function ID: 1
Author: T. Adam	Date: 20/01/2005
Version: 0.0	
Parameters: Inception Details (10-bit integer)	
Returns: JobCardDetails	
Description: <b>PrintJobCard() is responsible for printing the job card for that particular batch</b>	
Data Structures: InceptionDetails file store.	

Detailed Design:

READ FROM InceptionDetails file store

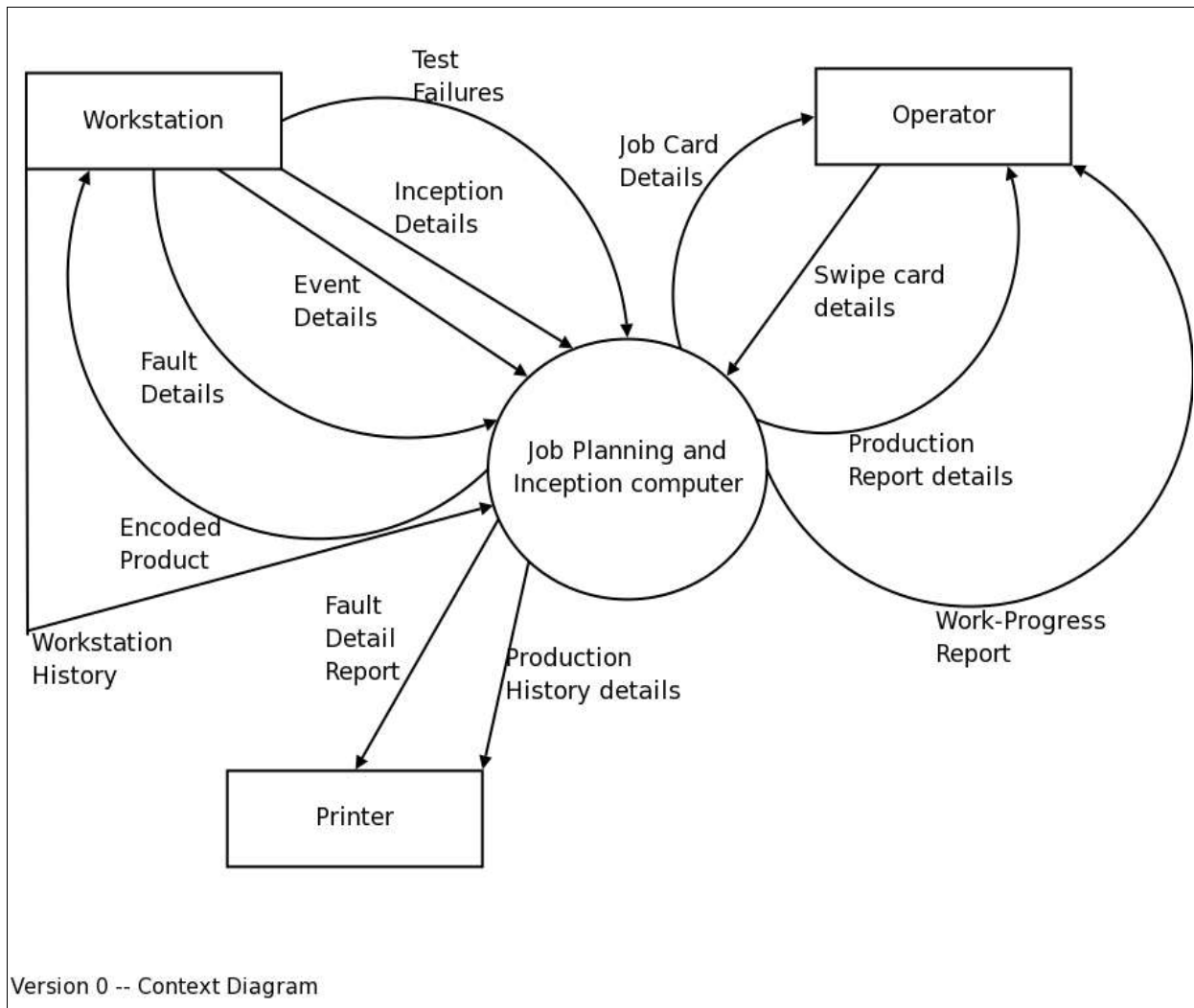
    Get InceptionDetails

    Produce JobcardDetails

END READ

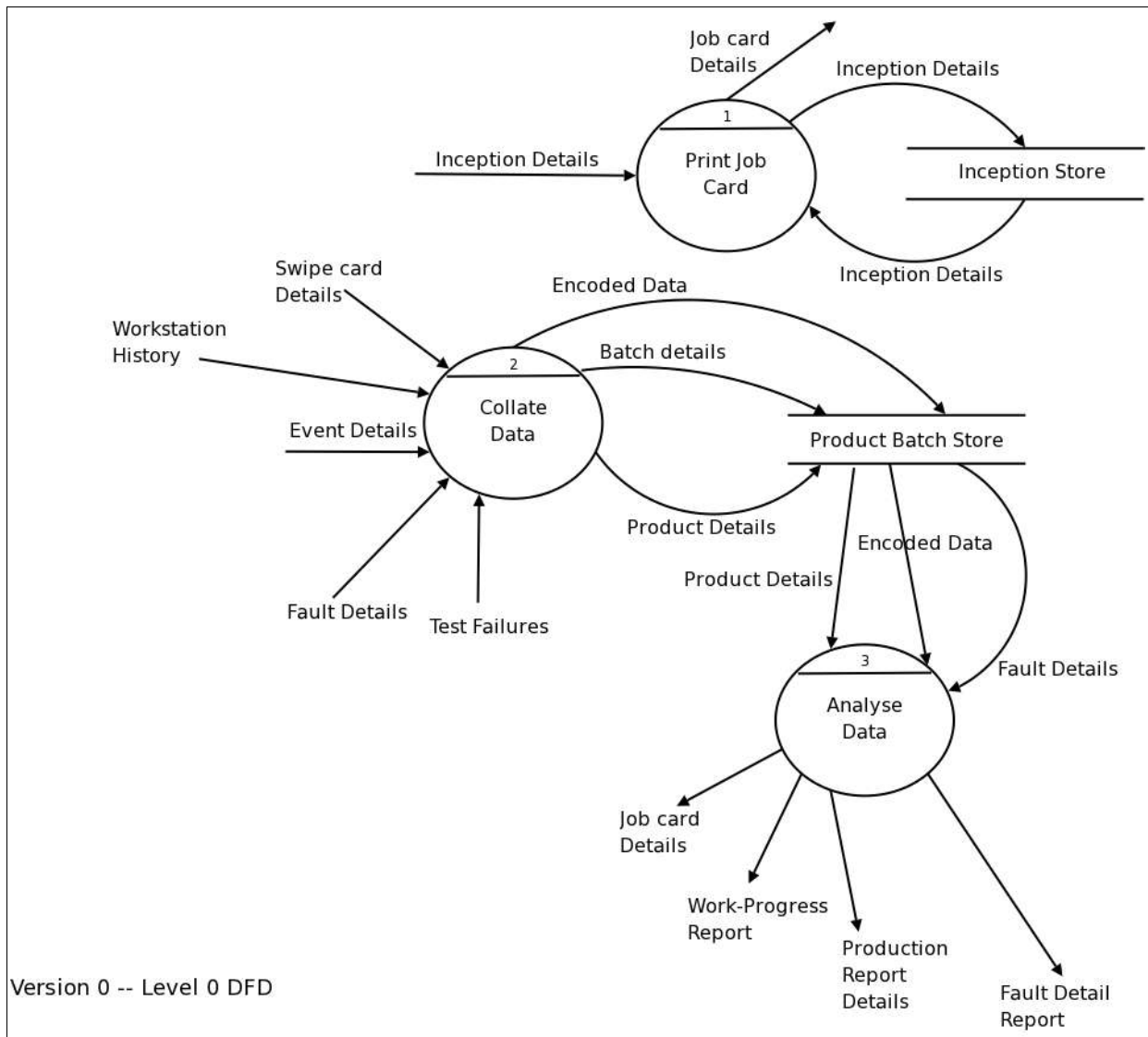
**APPENDIX A (ANALYSIS AND DESIGN MODEL)**

**Context Diagram**



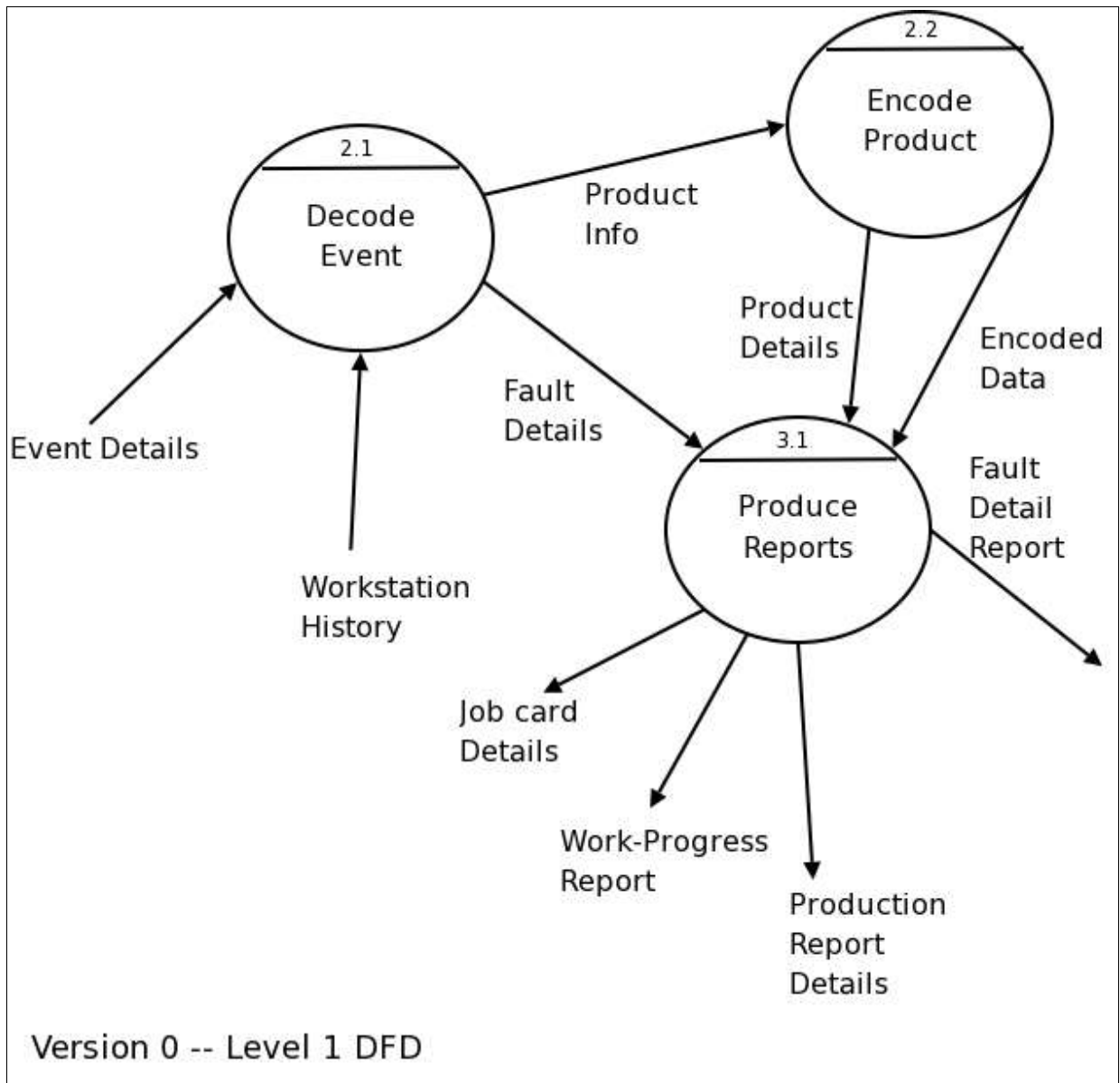
**APPENDIX A (ANALYSIS AND DESIGN MODEL)**

**Data Flow Diagram Level 0 (DFD #0)**



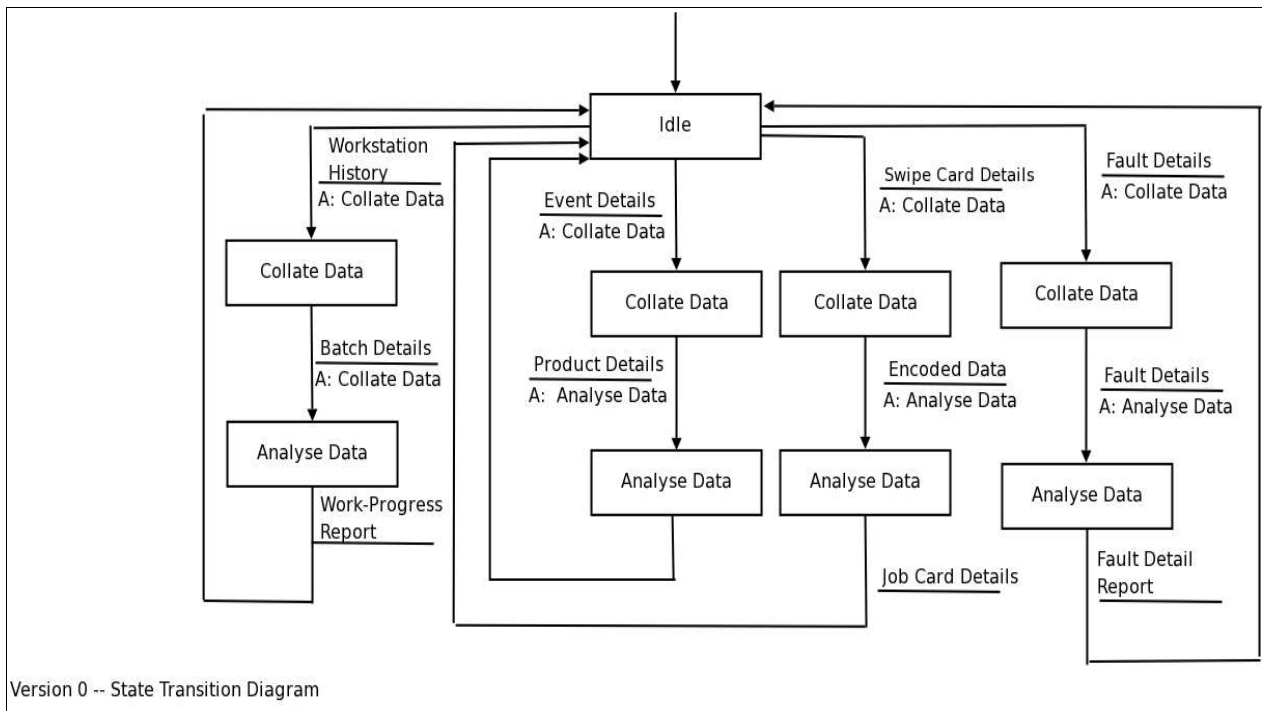
**APPENDIX A (ANALYSIS AND DESIGN MODEL)**

**Data Flow Diagram Level 1 (DFD #1)**



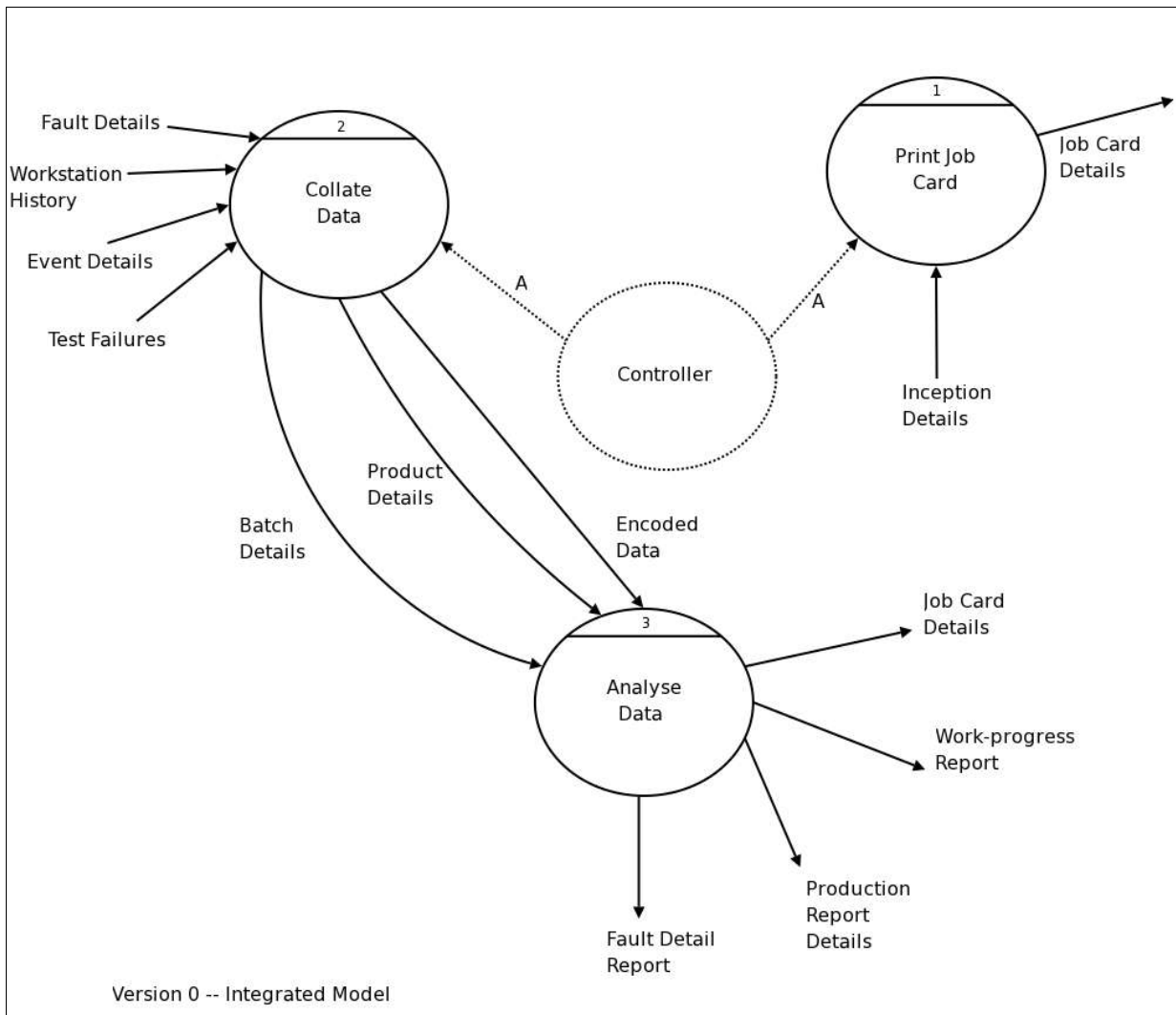
**APPENDIX A (ANALYSIS AND DESIGN MODEL)**

**State Transition Diagram**



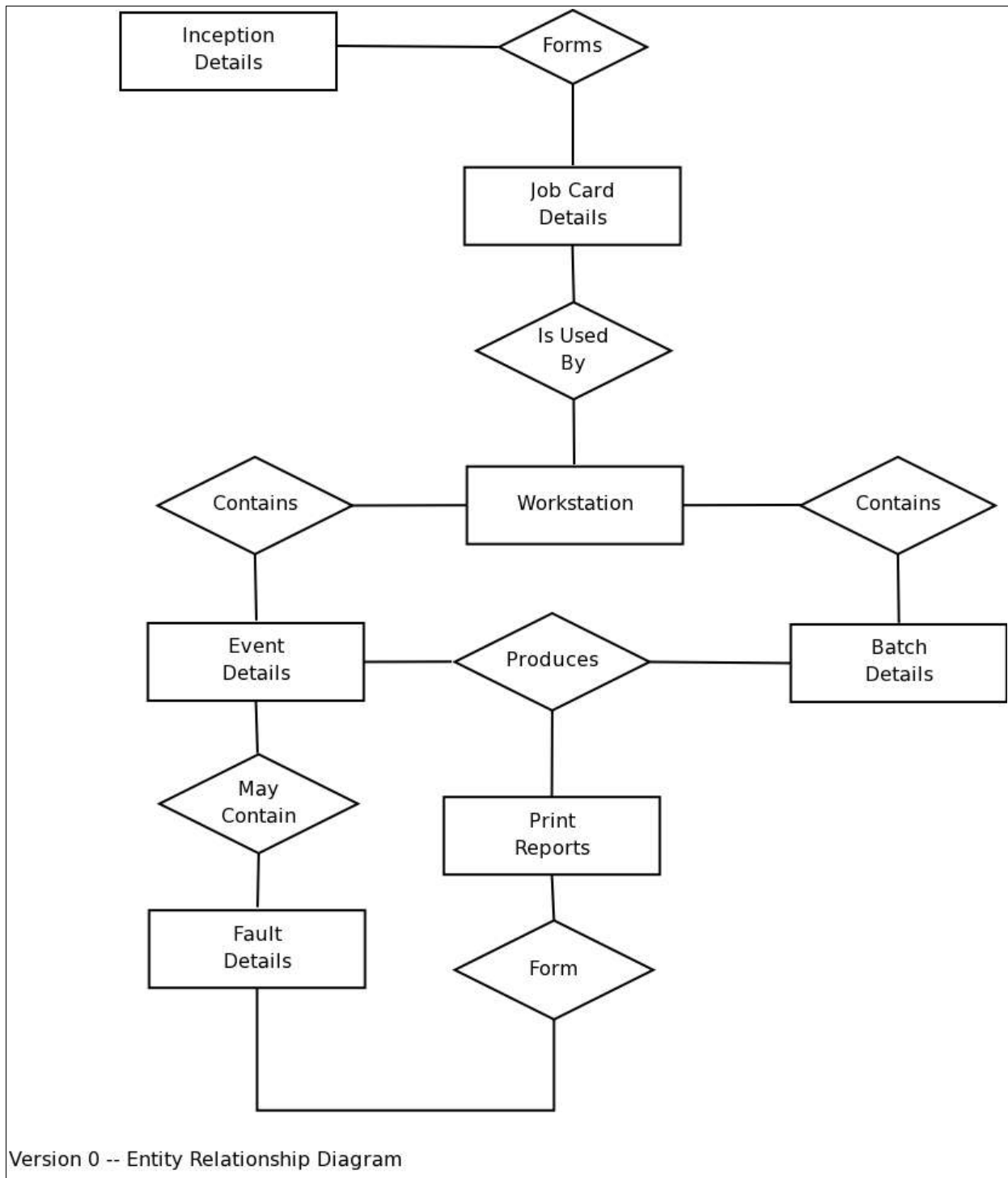
### APPENDIX A (ANALYSIS AND DESIGN MODEL)

#### Integrated Model



**APPENDIX A (ANALYSIS AND DESIGN MODEL)**

**Entity Relationship Diagram**



**APPENDIX A (ANALYSIS AND DESIGN MODEL)****PROCESS SPECIFICATION 1 (PSPEC #1) – PRINT JOB CARD**

@IN = inception details  
@OUT = Job Card Details

@PSPEC 0.1

READ FROM InceptionDetails file store

    Get InceptionDetails

    Produce JobcardDetails

**END READ**

**PROCESS SPECIFICATION 2 (PSPEC #2) – COLLATE DATA**

@IN = test failures  
@IN = fault details.  
@IN = event details  
@IN = workstation history  
@IN = swipecard details

@OUT = encoded data  
@OUT = batch details  
@OUT = product details

@PSPEC 0.2

WHILE there is data

    DO

        IF Data is faultdetails OR testfailures OR eventdetails OR workstationhistory

            CALL collateData()

        ELSE IF Data is InceptionDetails

            CALL printJobCard()

END WHILE

**PROCESS SPECIFICATION 3 (PSPEC #3) – ANALYSE DATA**

@IN = encoded data  
@IN = batch details  
@IN = product details

@OUT = Job card details

@OUT = work progress details

@OUT = production report details

@OUT = fault detail report

**APPENDIX A (ANALYSIS AND DESIGN MODEL)****Data Dictionary**

<i>Entity Name</i>	<i>Type</i>	<i>Description</i>	<i>BNF</i>
Inception Details	Data Flow	Contains Customer reference (eight digits), and batch size of two digits	
Batch Details	Data Flow	12 digit number to identify a batch	
Event Details	Data Flow	Comprises the start and end times for events that take place at a given workstation. The data comprises of start and end times for batches and products	Date/ Time
Workstaton History	Data Flow	Contains the manufactueing process identifier, workstation identifier, product batch identifier, start end times for the product batch as well as that of the workstation.	
Encoded Data	Data Flow	The data passed from a workstation after the product has been tested for.	
Fault Details	Data Flow	Collation of event and batch details comprised with any fault analysis used.	
Test Failures	Data Flow	Comprises the workstation history and batch details	
Workstation	Entity	The physical terminal an operator works at	
Operator	Entity	The person entering data in at the terminal	

<i>Entity Name</i>	<i>Type</i>	<i>Description</i>	<i>BNF</i>
Printer	Entity	The device used to print production history reports.	

**APPENDIX B (TRACEABILITY INFORMATION)****Traceability Table**

<i>Requirements Specification</i>	<i>Model Entity</i>	<i>On Page</i>	<i>Design Entity</i>	<i>On Page</i>
2.1.2.1, 2.1.5, 2.1.6	Transform 1 print job card	DFD#0 pp10 PSPEC#1 pp15	PrintJobCard()	Structure Chart pp 5 Function Spec pp 7
2.1.4, 2.2.1.1, 2.2.1.2, 2.2.1.3, 2.2.1.4, 2.3.2, 2.3.3	Transform 2 Collate Data	DFD #0 pp10 PSPEC#2 pp15	CollateData()	Structure Chart pp 5 Function Spec pp 6
2.3.5, 2.3.6, 2.4.5, 2.5.1, 2.5.6, 2.6.1	Transform 3 Analyse Data	DFD#0 pp10 PSPEC#3 pp15	AnalyseData()	Structure Chart pp 5 Function Spec pp 7

**APPENDIX C (CODE AND TEST TABLE)**

C code

The following details the code for the printJobcard() function:

```
#include <stdio.h>
#include <stdlib.h>

#define MAXLENGTH 12

int PrintJobCard(int inceptionDetails)
{
    /* initially, this will arrive as an integer -- that's fine
    * just make sure it is of the right length
    */

    if( (sizeof(inceptionDetails) == MAXLENGTH))
    {
        /* yep -- this worked */
        jobCardDetails = inceptionDetails;
    } else {
        return 0;
    }
}
```

**Test Table**

<i>Test No</i>	<i>Summary</i>	<i>Input</i>	<i>Expected Output</i>	<i>Actual Output</i>	<i>Comments</i>
1	Test for input length	11	0	0	Worked
2	Test for input length	269761762111	269761762111	269761762111	Worked