VICTORIA UNIVERSITY OF TECHNOLOGY

DEPARTMENT OF COMPUTER AND MATHEMATICAL SCIENCES

Data Management in Support of Measuring Operational Availability of Complex Systems

Anthony John Shelley and Peter Cerone

(74 COMP 28)

May, 1996

(AMS : 68 P15, 90B25)

TECHNICAL REPORT

VICTORIA UNIVERSITY OF TECHNOLOGY
P O BOX 14428
MELBOURNE CITY MAIL CENTRE
MELBOURNE, VICTORIA, 8001
AUSTRALIA

TELEPHONE (03) 9688 4492
FACSIMILE (03) 9688 4050

Footscray Campus

DATA MANAGEMENT IN SUPPORT OF MEASURING OPERATIONAL AVAILABILITY OF COMPLEX SYSTEMS

An INVITED PAPER
in the programme for
IFORS 96
Vancouver, Canada 1996

Anthony John Shelley

Peter Cerone

Victoria University of Technology
Department of Computer and Mathematical Sciences
March 1996

COPY RESTRICTED

- 1996 Australia

Transfield Defence Systems Pty Ltd

This document contains information that is proprietary to Transfield Defence Systems. This work is copyright. Other than as permitted by law, no part of it may be reproduced, stored in a retrieval system or transmitted in any form or by any process without prior written permission.

Transfield Defence Systems Pty Ltd

Nelson House, Nelson Place

Williamstown, Victoria 3016

Australia

ABSTRACT

A database model and supporting utilities are developed to enable efficient data loading, storage, manipulation and retrieval of Baseline and Event Data. Baseline Data will be supplied by Transfield Defence Systems. Event Data is collected and supplied by the Royal Australian Navy. Event Data has been split into two categories; Primary and Supplementary. Primary Event Data is required for the calculation of Achieved Operational Availability of complex systems. Supplementary Event Data is supplied on demand to support engineering and reliability, maintainability and availability investigations. Yourdon's Essential Model, which is based on the structured analysis approach for developing system models, is utilised. The Operational Availability Recording and Reporting System (OARRS) is intended to be used, throughout the service life of the Australian ANZAC Ship Class, as the tool to support the determination of logistic support effectiveness.

1. INTRODUCTION

Transfield Defence Systems Pty Ltd (TDS) was awarded a contract to build and deliver 10 ANZAC Class frigates for the Royal Australian Navy (RAN) and the Royal New Zealand Navy (RNZN) on the 10th of November 1989. Eight frigates were ordered by the RAN and two by the RNZN. The ANZAC frigate is a 3500 ton combatant based on the Blohm + Voss designed MEKO 200 PN (see Figure 1).

Figure 1 ANZAC Class frigate

A significant part of the ANZAC Ship Project (ASP) contract is the Integrated Logistic Support (ILS) package that details support requirements. Logistic support is the set of activities, products and services necessary to assure the effective and economical support of a system throughout its programmed life cycle. Blanchard (1992) defines logistic support as

> *an integral part of all aspects of system planning, design and development, test and evaluation, production and/or construction, consumer use and system retirement. The elements of support must be developed on an integrated basis with all other segments of the system.*

The principal logistic support activities and resulting products and services required by the ASP contract are:

- development of maintenance documentation incorporating on-board, shore-base and shore-depot maintenance requirements,
- identification and procurement of spare parts to support operational and maintenance requirements,
- provision of technical documentation to support operation and maintenance at the on-board, shore-base and shore-depot levels, and
- develop and conduct training courses for the first two ships' crews on equipment operation and maintenance.

The products associated with the above requirements are hereinafter referred to as ILS deliverables or simply deliverables.

The Department of Defence (DOD) defines ILS as

> *a disciplined, unified, and iterative approach to the management and technical activities necessary to:*
>
> - *integrate support considerations into system and equipment design,*
> - *develop support requirements that are related consistently to readiness objectives, to design, and to each other,*
> - *acquire the required support and*
> - *provide the required support during the operational phase at minimum cost* [DOD 1983].

A major ILS objective is to assure the compatibility and integration of all ILS deliverables.

Top level characteristics of the ILS specification of the ASP contract are:

- fixed price,
- Test, Evaluation and Validation (TE&V) programme for ILS products,
- Operational Availability Assessment Period (OAAP) for data collection and analysis subsequent to ship delivery,
- ILS deliverables warranty is based on Operational Availability[1] (A_o) threshold limits.

These characteristics combine to provide incentives for TDS to identify and procure an optimum set of spare parts and support components. The advantage to the Client is through-life cost savings realised from the reduced cost of spare parts inventory management over the three decades of in-service operations of the ANZAC Class. The methodology used to identify an optimised set of spare parts is based on the following:

---

1 Operational Availability is the probability that a system or equipment, when used under stated conditions in an actual operational environment, will operate satisfactorily when called upon [Blanchard (1992)].

- Reliability, Maintainability and Availability (RM&A) models (i.e. A_o model),
- assessment of maintenance requirements,
- operational experience.

TDS is required to establish procedures for the measuring of A_o achievements in specified critical systems[2]. The A_o assessment programme extends over a planned 10 ship operating years[3] or from delivery of ANZAC 01 to one year after delivery of ANZAC 04 (hereinafter termed the Operational Availability Assessment Period (OAAP)). The A_o threshold for the overall combatant capability is specified in the ASP contract as 0.8. A threshold of 0.9 is specified for selected critical systems.

During the OAAP, TDS is required to demonstrate that the logistic package is adequate to sustain an A_o equal to or greater than the threshold. If it is found that the prescribed thresholds have not been achieved, or there are accepted deficiencies in the non-critical systems, TDS is required to implement remedial action to rectify deficiencies in the ILS deliverables. Additionally, other logistic support performance indicators (hereinafter referred to as Measures of Effectiveness (MOEs)) for both critical and non-critical systems are to be calculated using data collected during the OAAP.

1.1 Spares Optimisation

TDS is required to procure the first set of spare parts for each ship and the shore-based warehouse facilities. An A_o model[4] incorporating all levels[5] of spares holdings, failure rate, logistic delay time and cost has been developed to predict the adequacy of sparing levels. The fundamental equation used to predict A_o during the analysis stage is:

$$A_o = \frac{k * MTBF}{k * MTBF + MDT}, \quad (1)$$

where A_o = Operational Availability during the analysis stage,

$MTBF$ = Mean Time Between Failure,

k = the ratio of calendar hours to annual operating hours,

MDT = Mean Down Time = $(MTTR + MADT + MLDT)$,

---

2 A "Critical System" is a group of equipment specified to be critical by the ASP contract.

3

ANZAC 01	-	4 ship years	planned delivery March 1996
ANZAC 02	-	3 ship years	planned delivery March 1997
ANZAC 03	-	2 ship years	planned delivery November 1997
ANZAC 04	-	1 ship year	planned delivery November 1998
Total		**10 ship years**	

4 The "A_o Model" was developed by Scientific Management Associates (SMA) Australia.

5 The "Levels" of spare part holdings are; Onboard, Base, Depot, Rotatable Pool, Major Insurance and Shore Installations.

with *MTTR* = Mean Time To Repair, constitutes the total elapsed time required (when the system is not operational) to repair and restore a system to full operating status,

MADT = Mean Administrative Delay Time, is that portion of downtime during which maintenance is delayed for reasons of an administrative nature; personnel assignment priority, gather technical manuals, test equipment and so on,

MLDT = Mean Logistic Delay Time, is the maintenance downtime that is expended as a result of waiting for a spare part to become available.

The dominant parameters in the calculation of A_0 are MTBF, a reliability design term and MDT, a supportability term. First priority for improving A_0 of an equipment or system during design and development would be given to increasing MTBF. Most ANZAC systems are mature in design, therefore MDT becomes the significant parameter. MDT is the sum of MTTR, MADT and MLDT. During the analysis stage, both MTTR and MADT are based on contractual default values, leaving MLDT as the independent variable in the A_0 equation. MLDT is a function of the level of sparing, that is, whether the needed spare is held on-board the ship or ashore at a central warehouse. Consequently, the supply effectiveness[6] is a vital aspect of supportability with respect to A_0 improvement.

1.2 Calculation of Achieved A_0 during the OAAP

The equation used to calculate achieved A_0 during the OAAP is:

$$A_o = \frac{Uptime}{Uptime + Downtime}, \qquad (2)$$

where A_o = achieved Operational Availability during the OAAP,

Uptime = total calendar time less downtime and time for Selected Restricted Availabilities (SRAs)[7],

Downtime = the sum of all recorded down time.

TDS is required to present the downtime data schematically, as shown in Figure 2.

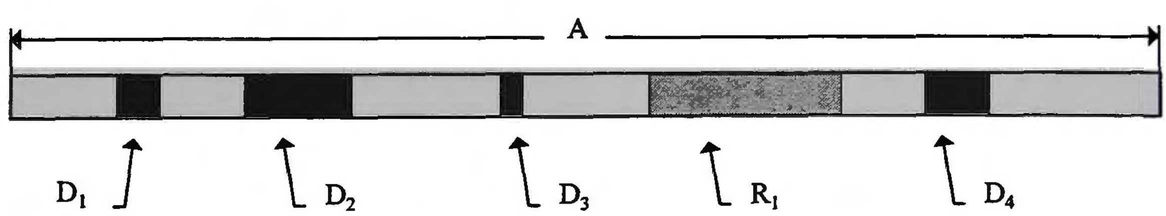


Figure 2 Schematic Representation of Downtime.

6 "Supply effectiveness" is the probability that a spare part will be available on-board when needed.

7 A "SRA" is a planned time period in which major works are performed on the ship, usually in dry dock.

From Figure 2 the parameters required to calculate achieved A_o are defined as:

$$Downtime = \sum_{i=1}^{m} D_i \text{ where } D_i \text{ are critical failure downtimes,}$$

$$Uptime = A - (Downtime - \sum_{i=1}^{n} R_i) \text{ where } R_i \text{ are refit or SRA periods,}$$

where $m = 4$, $n = 1$ is illustrated.

For TDS to determine achieved A_o and the other MOEs required during the OAAP, the following steps are taken:

- define the data elements that reside in the ANZAC Maintenance and Planning System (AMPS), required for calculation of achieved A_o using Equation (2),
- develop a methodology to collect and transfer the data (the methodology for systematically collecting information to produce reliability\availability data is generally referred to as *Event Data Collection* [Bello (1985)]),
- develop an *Event Data* database to hold the collected data,
- use the Event Data database to calculate and report the achieved A_o and other MOEs,
- define and implement trigger points to initiate corrective action,
- develop a reporting methodology for delivery of achieved A_o and other MOEs to the Client.

2. Event Data

2.1 ANZAC ship Maintenance and Planning System (AMPS)

Prior to the ANZAC Class, the Client managed Class and on-board maintenance according to procedures described in the RAN Ship Maintenance Administration Manual, ABR5230[8]. AMPS, it is suggested, heralds a new era for the Client in Class and ship support capabilities. AMPS is a computerised logistic support system developed to receive and manage all logistic support data produced for the ANZAC Class. AMPS comprises three main modules, namely:

1. Facilities Maintenance Management System (FMMS)[9] that provides the on-board and ANZAC class maintenance management functions,
2. Procurement Automation (PA) that provides the on-board and ANZAC Class spare part and inventory management functions and,
3. Documentation configuration/management software.

According to Balestreri (1986) to deliver (to the Client and TDS) good quality reliability data a centralised archive of all data is essential. Furthermore, the functional break-down of systems into

---

8 ABR5230 details manual procedures for the recording and reporting of planned and un-planned (breakdown) maintenance.

9 The Facilities Maintenance Management System is proprietary software developed by KDR Creative Software Pty Ltd Melbourne.

various homogeneous equipments enables data relating to the particular system to be collated and analysed more effectively. The RAN developed Common Management Code[10] (CMC) was chosen as the primary key, to codify the ship's systems and equipments within FMMS.

AMPS, in particular FMMS, will be the centralised data storage point for the ANZAC Class and the CMC provides the mechanism, within FMMS, to relate spare part transactions, maintenance activities and technical documentation to a particular system/equipment. AMPS will be the primary source of reliability data, for both TDS and the Client, during and subsequent to the Operational Availability Assessment Period.

2.2 An Event

The fundamental object of data collection will be an "Event". An Event in this context is:

> *a maintenance occurrence that indicates that a system or piece of equipment is not capable of performing its task as defined in the ASP Contract.*

An Event can either be planned or un-planned maintenance that may or may not cause system failure (or downtime). An Event is captured within FMMS via a "job" object and is characterised by the following attributes:

- CMC of the equipment the job relates to,
- event start and end times,
- criticality (whether the job relates to a critical or non-critical system),
- activity cause (whether the job is planned or un-planned maintenance),
- the job's Time To Repair (TTR) and Logistics Delay Time (LDT),
- text detailing failure cause and affect on equipment,
- associated spare parts usage.

2.3 Event Allocation

FMMS automatically associates planned maintenance jobs with the appropriate equipment, however, un-planned maintenance jobs require the maintainer to manually establish this job/equipment association. The functionality of each critical system/equipment, from the A_0 measurement point of view, is defined by its associated Availability Block Diagram (ABD). The ABD is a logic diagram that models the impact of equipment failure on the availability of the top-level system. Each rectangle (or block) within the ABD is identified by CMC and represents an equipment (or in some cases another system or component). The ABD logic models equipment dependencies (series connections) and redundancies (parallel connections). Critical system ABDs are developed by TDS with their functionality approved by the RAN.

---

10 The CMC is an 11 character, alpha-numeric code that identifies a piece of equipment within a system, and relates stores support and documentation to that equipment. The hierarchical code is divided into five areas of management interest, with four levels of breakdown.

For example, the Horizon Reference Set critical equipment ABD (see Figure 3) comprises two Ship's Inertial Navigation System (SINS) (identified by CMC CD-N-SM-A-MW001), one Control Indicator (identified by CMC SU-B-ZF-E-IN002), one Electronics Component Assembly (identified by CMC SU-B-ZF-E-KD000) and one Indicator Stabilisation Data (identified by CMC SU-B-ZF-E-IN001). The two SINSs form a parallel connection that in turn is part of a series connection including the Electronics Component Assembly and Indicator Stabilisation Data. The Control Indicator has no effect on A_o calculations, however is included in the critical equipment ABD for completeness.

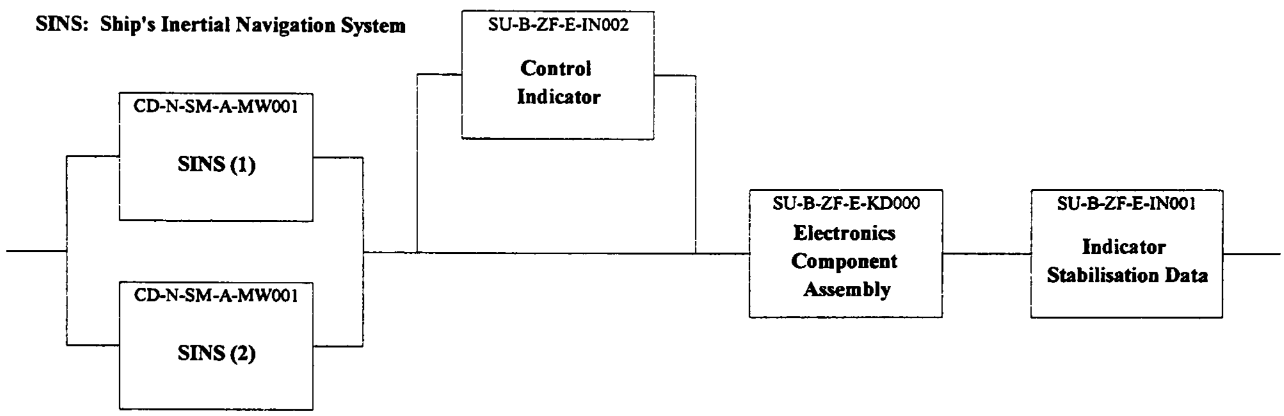


Figure 3 Horizon Reference Set Critical Equipment ABD

The logic of Figure 3 dictates that if SINS(1) fails (i.e. SINS(1) is down), the Horizon Reference Set critical equipment remains operational (up), however if both SINS are down, the Horizon Reference Set is considered to be down. Further, if one of the SINS and the Electronics Component Assembly are down at the same time, the Horizon Reference Set is down.

The CMC provides the correlation between a block in a critical system's/equipment ABD and an event that has been recorded in FMMS. Therefore, it is of paramount importance for the maintainer to allocate the FMMS job (i.e. Event) to the correct piece of equipment. The requirement to manually associate an Event with an equipment has been identified as a potential source of inaccuracies in the determination of achieved A_o. Hence the need to validate event allocation.

2.4 Baseline Data

Baseline data is a broad class of data used to support the calculation of MOEs and the generation of RM&A reports. Baseline data is sourced from TDS, see Figure 4, and incorporates the following components:

- a hierarchical list of all the CMCs (i.e. equipments) on-board the ship,
- configuration data for each critical and non-critical system/equipment,
- for spare parts, the recommended allowances to be held on-board and at each level in the RAN supply system (i.e. Base and Depot),

- Planned Maintenance Documentation (PMD) data that includes estimated time to complete the PM and initial equipment default downtimes,
- and for critical systems, availability goals and thresholds, ABD configuration data, projected annual operating hours and default values for Administrative Delay Time (ADT), TTR and LDT.

2.5 Primary Event Data

Primary Event data comprises the minimum data required to establish details of a completed planned or unplanned maintenance event for the purpose of calculating MOEs and in particular the achieved A_o (Contract) MOE. It is the Client's responsibility to collect and archive such data and to make it available to TDS as required. Figure 4 shows Primary Event data originating from AMPS (on-board the ship) and being delivered to OARRS via the ANZAC Class Logistics Office (ACLO).

Primary event data incorporates the following components:

- event history data (which equates to a completed FMMS "job") that includes; duration of the job, actual time to repair, actual logistics delay time and the associated affected equipment,
- event materials data that identifies the quantities of spare parts consumed by an event,
- equipment and ship operating conditions, cycles and hours.

2.6 Supplementary Event Data

Supplementary event data comprises all additional data that may be required to be accessed by TDS to allocate ambiguous events that cause downtime, validate spares allowances, analyse events contributing to unusual failures, excessive downtime or spares usage, or for RM&A reporting. These data may be sourced from the ship's Control and Monitoring (C&M) system, defect reports attached to an FMMS "job", or from manually maintained ship's logs and records (see Figure 4). It is the Client's responsibility to collect and archive such data and to make it available to TDS as required. Supplementary Event data is also delivered via ACLO.

For unplanned maintenance events, supplementary data may include:

- circumstances of the failure (system history, operational parameters, external conditions, etc),
- textual description of the event,
- cause of the failure (e.g. wear, defective materials, human error, etc),
- nature of the repair made (repair, replacement from stock, etc),
- special (unanticipated) equipment, skills or other resource required for making the repair,
- ship and equipment operating cycles, hours and parameters.

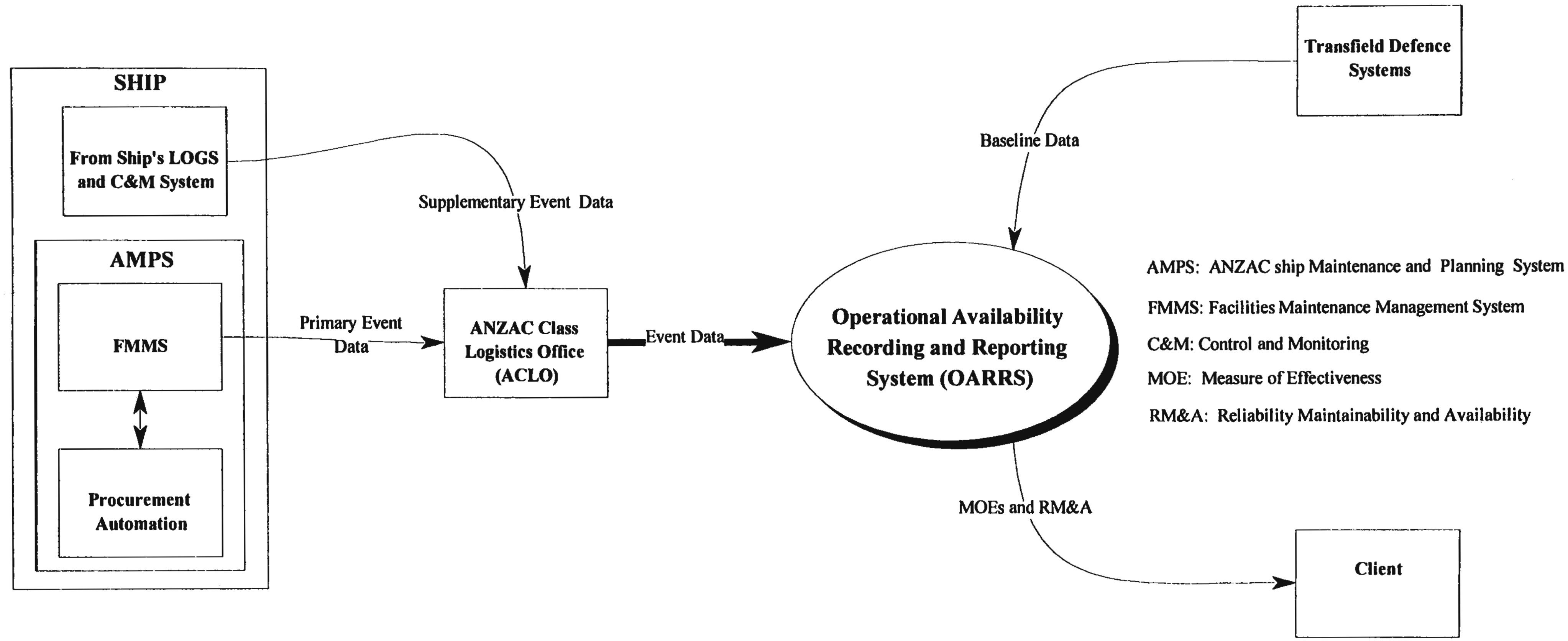


Figure 4 Identification of Baseline and Event Data Sources

2.7 Measures Of Effectiveness

The identification and definition of MOEs is an on-going, collaborative effort between TDS ILS and Client personnel. At the time this paper was submitted, eight MOEs had been identified and defined. However, only the Achieved A_o (Contract) MOE is documented here. The definition of Achieved A_o (Contract) has been adapted from Hall and Shelley (1995).

2.8 Definitions

t_s Start Date of the measurement period covered by the report.

t_e End Date of the measurement period covered by the report.

T Total Time of the measurement period $\equiv T = t_e - t_s$.

T_{sra} Duration of any Selected Restricted Availabilities (SRA) during the measurement period as determined from the primary data.

T_o Ship's Operational Time used for calculating MOEs.

$$T_o = T - \sum T_{sra}$$

2.9 MOE_c - Achieved A_o (Contract)

MOE_c is the achieved operational availability for comparison with the contractually established availability goals and thresholds. This MOE measures the effectiveness of the ILS Package with respect to range, depth and level of spare parts. MOE_c is expressed as

$$MOE_c = \frac{T_o - \left(\sum_{i=1}^{n} (ADT_d | ADT_a + TTR_d | TTR_a + LDT_d | LDT_a)_i \right)_k}{T_o} \qquad (3)$$

where MOE_c = Achieved Operational Availability (Contractual) of critical system k,

T_o = Ship's Operational Time,

ADT_d = Administrative Delay Time (default),

ADT_a = Administrative Delay Time (actual),

$ADT_d|ADT_a$ = the lesser of the two values,

TTR_d = Time To Repair (default),

TTR_a = Time To Repair (actual),

$TTR_d|TTR_a$ = the lesser of the two values,

LDT_d = Logistic Delay Time (default),

LDT_a = Logistic Delay Time (actual),

$LDT_d|LDT_a$ = the lesser of the two values,

k = critical system under consideration,

i = un-planned maintenance event allocated to critical system k,

n = total number of un-planned maintenance events allocated to critical system k.

2.10 Data Collection

The specification of Event data elements to be collected and who is to collect this data are the two key responsibilities in the Event data collection exercise. During the OAAP the Client is solely responsible for the collection and archiving of Primary and Supplementary Event data, whilst TDS provides the Baseline data. It is the responsibility of TDS to specify the Event data elements to be collected. TDS is also required to specify formal data transfer protocols, for the transfer of Primary and Supplementary Event data between the Client and TDS.

Bendell (1988) stresses the importance of clearly identifying the primary purpose of a collection exercise. In developing the Event data element specification, TDS was guided by a number of key criteria, namely:

- the data elements required to calculate the various MOEs, with particular attention being given to MOE_c, Achieved Operational Availability (Contract), refer Equation (2),
- given the nature of the calculations that are to be performed on Primary Event data, collecting Primary Event data on a failure-count (or failure-rate) basis is not adequate, hence the need (for each Event) to capture total time, actual TTR, spare part consumption and the location from where the spare part was sourced. The capturing of such data allows for the derivation of other important data, such as the LDT and ADT associated with each Event,
- AMPS, in particular FMMS, should be the main source of Primary Event data,
- to minimise any extra effort required by the crew in capturing accurate Event data.

3. SYSTEM ANALYSIS & DESIGN

The concept of an Operational Availability Recording and Reporting System (OARRS) is not new, however, there is no manual process in place or computer system currently available that is able to calculate achieved A_o in the manner required and provide the necessary reports specified in the ASP Contract.

Yourdon's Essential model was favoured as the OARRS analysis and design methodology for the following main reasons:

1. the Essential Model is very much suited to the analysis and design of NEW applications,

2. TDS's existing application development methodology closely matches Yourdon's Essential Model.

The second point mentioned above is the main reason behind the decision to use Yourdon's Essential Model methodology for the analysis and design of OARRS.

3.1 The Essential Model

Yourdon (1989) defines the Essential Model as:

> *a model of what the system must do in order to satisfy the user's requirements, with as little as possible said about how the system will be implemented. The essential system model is seen as the essence of the system.*

The Essential Model is equivalent to the new logical model in the classical structured analysis methodology. The Essential Model consists of two major components:

1. Environmental model,
2. Behavioural model.

The *environmental model* defines the boundary between the system and the rest of the world and consists of a brief description of the purpose of the system, a context diagram, an event list, external data dictionary and external data model.

The *behavioural model* describes the way in which the system is to interact with the outside environment and consists of the process model, Entity Relationship Diagram (ERD), data dictionary, process specifications and data model.

3.2 OARRS Environmental Model

3.2.1 Statement of Purpose

The purpose of the Operational Availability Recording and Reporting System is to calculate and report the achieved A_0 of critical systems and equipment. The A_0 performance of selected systems and/or equipment is considered satisfactory if specified thresholds are achieved. OARRS will be the TDS centralised archive for all reliability data needed to support the raising of RM&A reports required by the ASP Contract.

3.2.2 Context Diagram

The context diagram is a special form of Dataflow Diagram (DFD) in which a single bubble represents the entire system. The context in which OARRS exists is defined through the following characteristics:

- the people, organisations and/or systems with which OARRS communicates (referred to as terminators),
- the data OARRS receives from the outside world,

- the data produced by OARRS and sent to the outside world,
- data stores that are shared between OARRS and the terminators,
- the boundary between OARRS and the rest of the world.

Figure 5 shows the OARRS context diagram.

3.2.3 Event List

The Event List, is simply a textual list of events that occur in the environment to which OARRS must respond.

The OARRS Event List follows:

1. Provide Baseline Data,
2. Provide Primary Event Data,
3. MOEs and RM&A Reports,
4. Provide Supplementary Event Data,
5. Provide Spares Change Data.

All Events are flow-oriented events; that is, the system becomes aware that the event has occurred when a piece of data arrives. These flow-oriented events correspond to a dataflow on the context diagram.

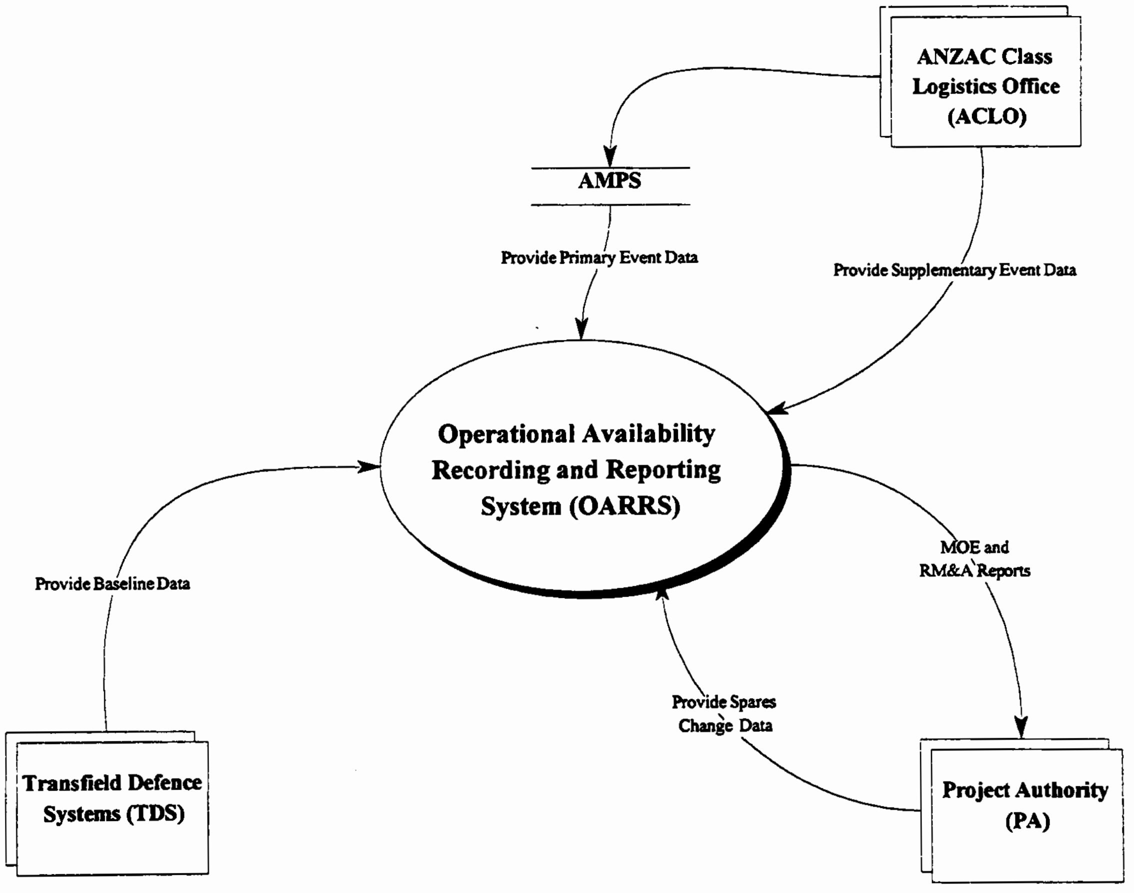


Figure 5 OARRS Context Diagram

3.2.4 External Data Dictionary

The external data store identified as "AMPS" in Figure 5 represents a number of tables that are part of the maintenance management module (i.e. FMMS) of AMPS. It is from these tables that OARRS will source the majority of primary event data required to generate the MOEs and RM&A reports.

The external data dictionary defines the tables and data elements (i.e. columns) that are the interface between FMMS and OARRS. The external data dictionary (see Appendix A. Table 2)[11] comprises the following columns:

1. **TABLE** — indicates the name of the FMMS table,
2. **KEY** — indicates the row(s) of the table's Primary Key,
3. **COLUMN** — column name of a field in the table indicated in the **TABLE** column,
4. **DATA TYPE** — data type of the field were:
   - X(8) = text field 8 characters in length,
   - F = float,
   - D = date,
5. **DESCRIPTION** — textual description of the data that populates the field.

3.2.5 External Data Model

The external data model details the relationships between the external interface tables identified in the External Data Dictionary. The external data model comprises three tables, *job_history*, *job_materials* and *facility_readings*. The **Primary Key fields** are in Bold typeface and the relationship between the *job_history* and *job_materials* tables is *"one or many" to "zero, one or many"* that is, a job can have zero, one or many materials and a material can belong to one or many job(s). Appendix A. Figure 15 shows the external data model.

3.3 OARRS Behavioural Model

3.3.1 The Process Model

The first step in developing the process model involves drawing the first-cut Dataflow Diagram (DFD), with one process (i.e. bubble) corresponding to the system's response to each event defined in the Event List. Data stores are then added to model the data that must persist between asynchronous events. At this point, event partitioning is applied to the first-cut DFD to produce the final process model.

---

11 Permission to reproduce the information contained in Appendix A has been granted to the author by Kevin Ramsey of KDR Creative Software Pty Ltd, the developer of FMMS.

Event partitioning divides a process into sub-processes, each sub-process is further partitioned into sub-processes and so on, until the level of an "atomic" process (i.e. a process that requires no further partitioning) is reached. Figure 6 details the OARRS first-cut DFD.

Processes 1, 2, 3, 4 and 5 (see Figure 6) model the OARRS top-level response to the five events detailed in the OARRS Event List. The in-coming and out-going dataflows are identified by arrows that point into and out of a process respectively. Data stores are represented by two parallel lines.

The following sections detail the event partitioning of process 3 (Generate MOEs and RM&A Reports) into a set of levelled DFDs. A similar method is used for the partitioning of the other four processes, however, the detail is not presented here.

3.3.1.1 Partitioning of the Generate MOEs and RM&A Reports Process

Figure 7 illustrates the downward partitioning of process 3 (see Figure 6) into Process 3.1 Generate Critical System MOEs and process 3.2 Generate RM&A Reports.

3.3.1.2 Partitioning of the Generate Critical System MOEs Process

Figure 8 illustrates the downward partitioning of Process 3.1 (see Figure 7). Table 1 describes the eight processes (identified in Figure 8) that correspond to the eight MOEs. Process 3.1.1 is partitioned further and discussed in Section 3.3.1.3.

Process Number	Process Description
3.1.1	Generates the achieved A_o (Contract) MOE.
3.1.2	Generates the achieved A_o (Actual) MOE.
3.1.3	Generates the achieved MTBF MOE
3.1.4	Generates the elapsed time ratio for planned maintenance MOE
3.1.5	Generates the TTR ratio for un-planned maintenance MOE
3.1.6	Generates the achieved logistic delay ratio MOE
3.1.7	Generates the achieved administrative delay ratio MOE
3.1.8	Generates the operating hours ratio MOE

Table 1 Descriptions of Processes 3.1.1 through 3.1.8

3.3.1.3 Partitioning of the Generate Achieved A_o (Contract) process

Figure 9 illustrates the downward partitioning of Process 3.1.1 (see Figure 8). All "bubbles" in Figure 9 represent Atomic processes and therefore are not partitioned any further.

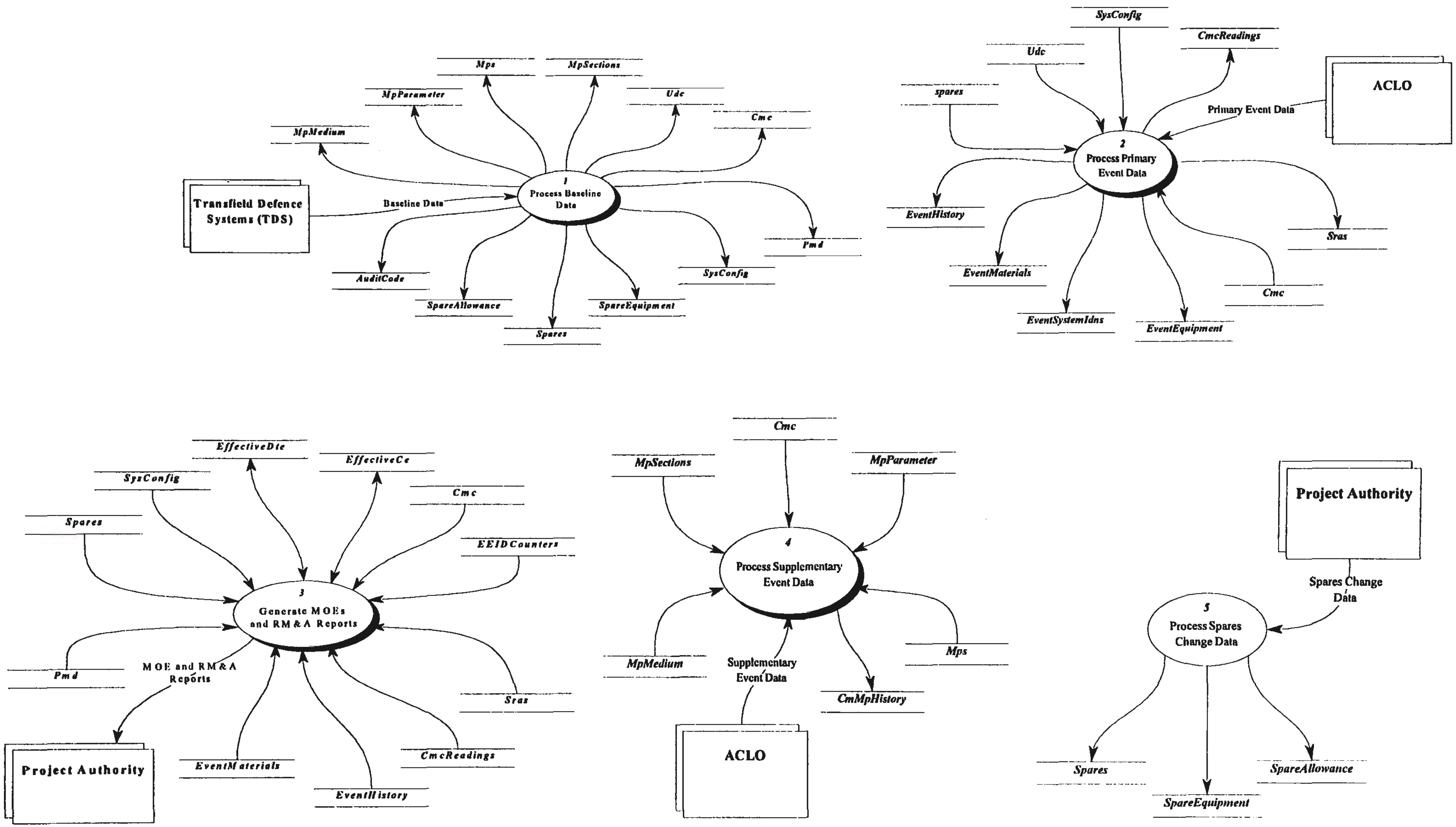


Figure 6 OARRS first-cut DFD

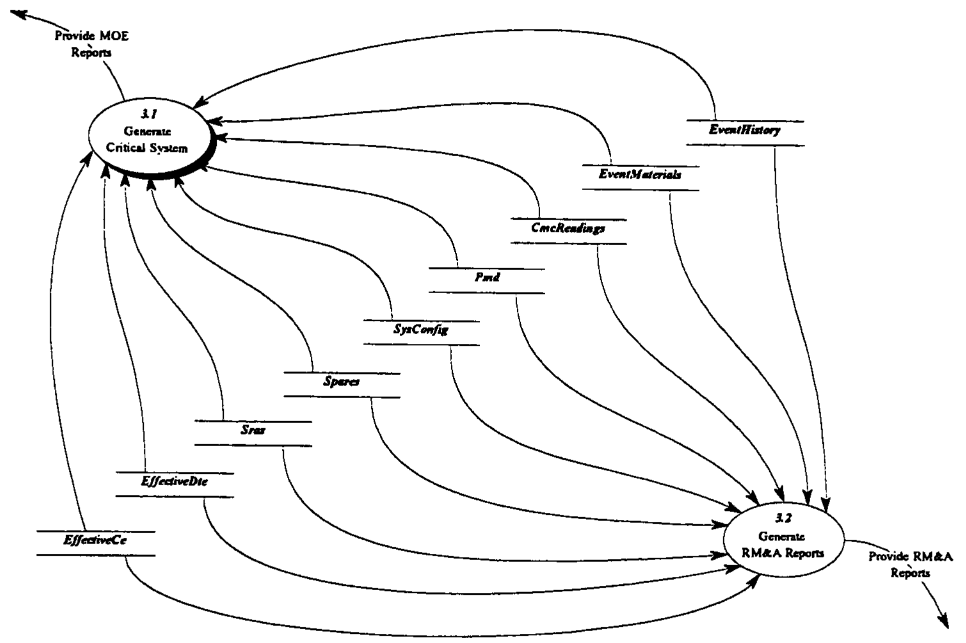


Figure 7 Partitioning of Process 3 in Figure 6

Process 3.1.1.1 filters Event Data obtained from the *EventHistory* data store, by **cmc**, **shutdown_cmc** (if data is present event caused equipment to be down) and **activity_cause** = "BKDN" indicating the Event is un-planned maintenance. The filtered data populates a temporary data store which is sorted by **contractual_end_date** in **date_act_start**.

Process 3.1.1.2 is performed on each "base-level block"[12] in an ABD and sources filtered Event Data form the *Temporary* data store, resolves any overlapping Events within the base-level block and populates the *EffectiveDte* and *EffectiveCe* data stores with Effective Events. The concept of an overlapping Event and an Effective Event are defined in Section 4.3, together with the method for resolving overlapping Events for base-level blocks.

Process 3.1.1.3 is performed on redundant blocks as defined by the critical system/equipment ABD. Referring to Figure 3, the SINS(1) and SINS(2) blocks are a redundant configuration (i.e. in parallel). Both SINS are base-level blocks, therefore Effective Events are sourced from the *EffectiveDte* data store, grouped and sorted by **ee_downtime_end** in **ee_downtime _start**. Effective Events are derived from the grouped data and stored in *EffectiveDte,* with contributing Effective Events stored in *EffectiveCe*. The method for resolving overlapping Events in redundant blocks is also described in Section 4.3.

---

12 Blocks in an ABD may contain within them ABDs of a lower level. A "base-level block" in an ABD is a block that does not contain a lower level ABD. A base-level block in an ABD may contain one or more CMCs.

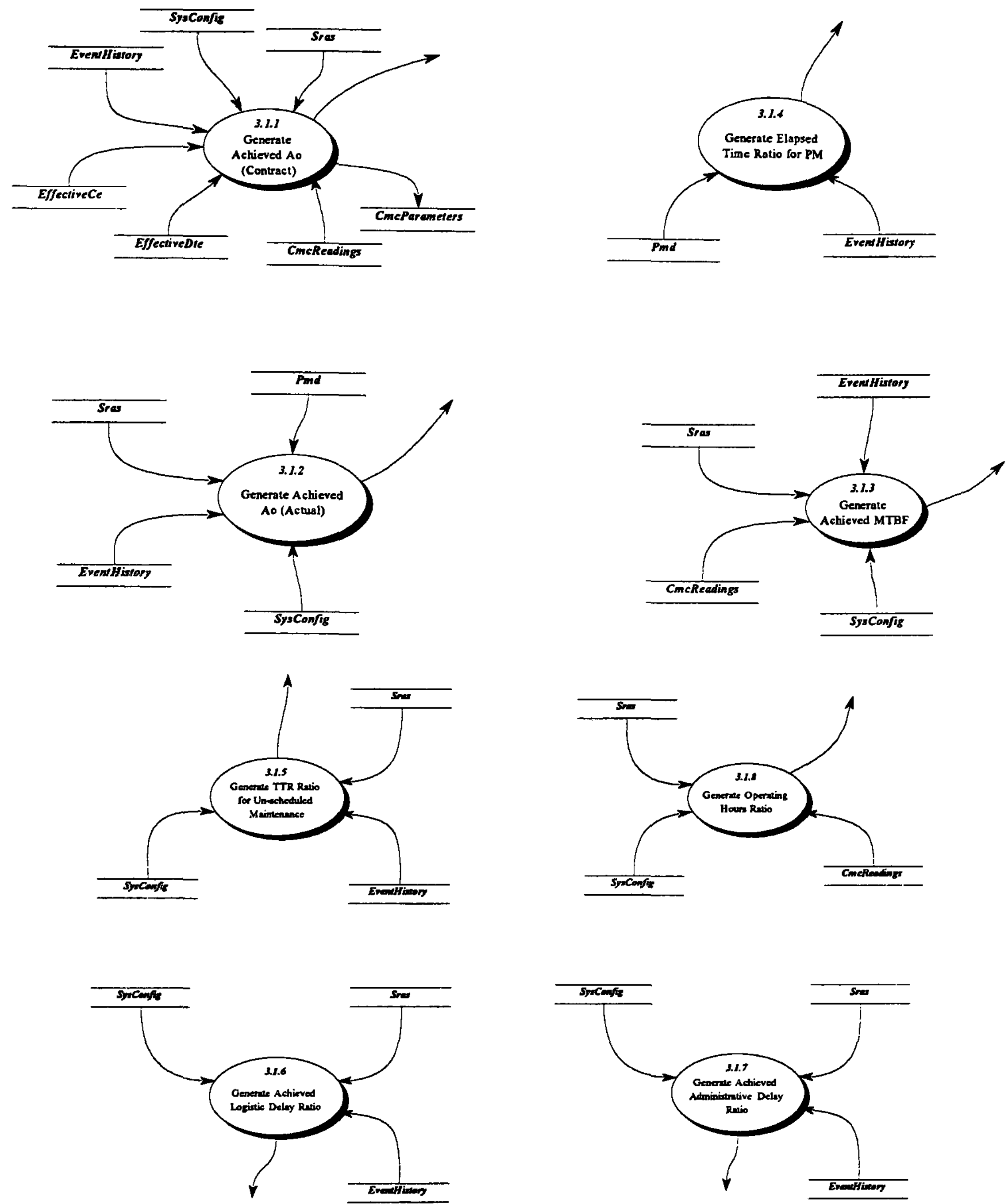


Figure 8 Partitioning of Process 3.1 in Figure 7

Process 3.1.1.4 sources Effective Events produced by process 3.1.1.3 and resolves any overlapping Effective Events. Overlaps here are of the same type as for process 3.1.1.2 and are resolved in the same manner.

Prior to process 3.1.1.5 running, all redundant configurations have been reduced to an effective block (by process 3.1.1.4) that has Effective Events, which are stored in the *EffectiveDte* data store. Therefore, process 3.1.1.5 deals with an "effective" ABD structure that includes only dependant (i.e. series) blocks. Process 3.1.1.5 sources Event data from the *EffectiveDte* data store; and by using a similar method to that of process 3.1.1.2, populates the *EffectiveDte* data store with Effective (downtime) Events and the *EffectiveCe* data store with the contributing Effective Events.

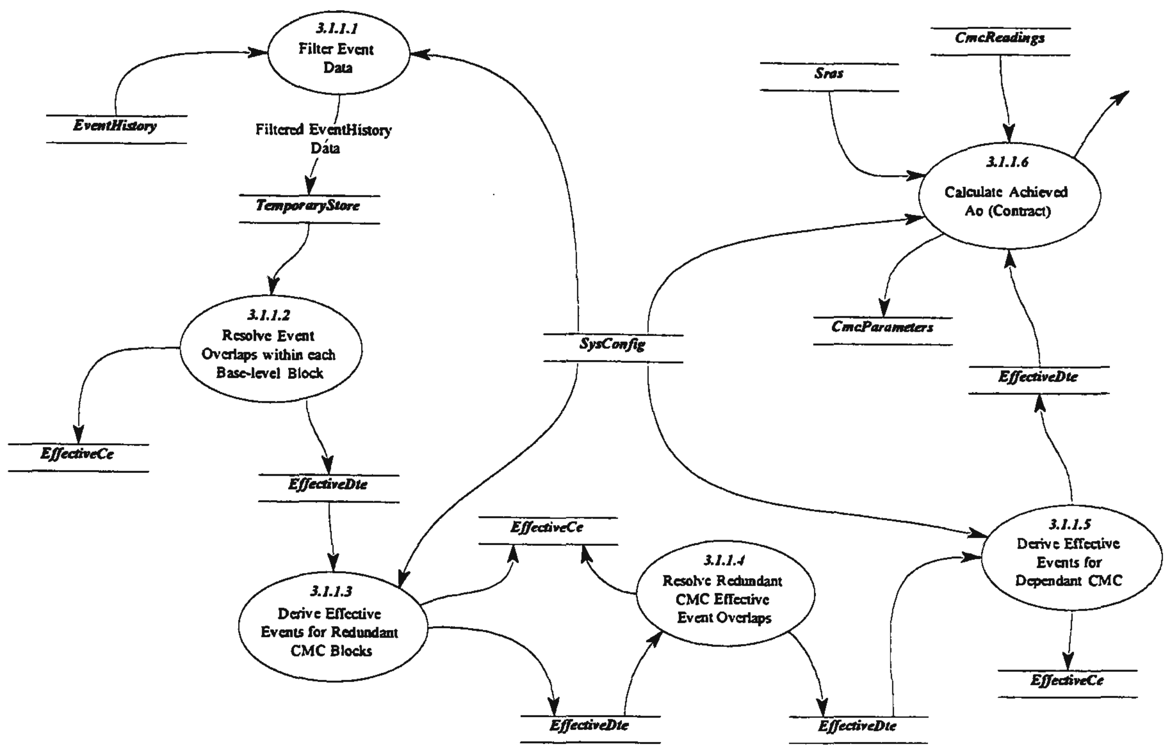


Figure 9 Partitioning of Process 3.1.1 in Figure 8

Process 3.1.1.6 accepts data from the *EffectiveDte* data store and calculates the achieved A_o (Contract) MOE using Equation (3). The calculated achieved A_o (Contract) MOE figure is stored in the *CmcParameters* data store.

3.3.2 The Entity Relationship Diagram (ERD)

The ERD is a model that describes the **stored data layout** of a system at a high level of abstraction and is quite different from the Process Model which models the **functions** performed by a system. Every data store on a DFD must correspond to an object type or a relationship on the ERD [Yourdon 1989]. Object names (ERD) and data stores (DFD) must match. The ERD is not detailed in this paper, however, the normalised Data Model, which is derived from the ERD, is detailed in a following Section and Appendix B.

3.3.3 The Data Dictionary

The Data Dictionary is an organised listing of all data elements that are pertinent to the system and entries must apply to both the DFD and ERD model. The OARRS Data Dictionary has been excluded from the paper.

3.3.4 The Process Specifications

Each "Atomic" process (in the Process Model) is described by a Process Specification. A Process Specification defines what must be done in order to transform atomic process inputs into outputs. OARRS Process Specifications have been written using structured English. The OARRS Process Specifications have also been excluded from the paper.

3.3.5 The Data Model

The data model, which is derived from the ERD, is a high-level view of the system that defines the organisation of data elements within tables and the relationships between tables identified during the database design phase. It models the relationship types between tables and identifies primary keys and indices. The OARRS data model is shown in Appendix B.

A table is denoted by a rectangle. Data elements are encapsulated within their respective tables with primary key data element(s) underlined. Relationships are indicated by the various lines linking tables with the relationship type denoted by the line terminator (see Figure 10 for line terminator descriptions).

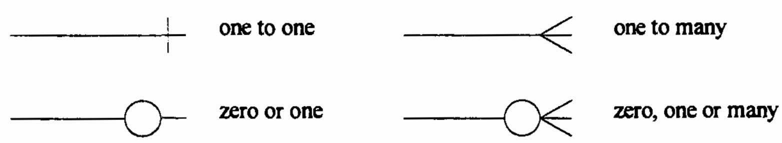


Figure 10 Line Terminator Descriptions

Presentation of the data model to the reader in an un-cluttered manner aids in the understanding of the model. To this end, a number of relationships have been deliberately omitted from the data model diagram, namely:

- the relationship between the *tbludc* data store and all other data stores that contain the udc data element; in all cases this is a 'zero or one' to 'one or many' relationship and
- the relationship between the *tblcmc* data store and all other data stores that contain the cmc data element; in all cases this is a 'zero or one' to 'one or many' relationship.

4. IMPLEMENTATION

4.1 OARRS Functional Overview

During the OAAP, TDS is required to calculate and report (to the Client) Achieved A_0 (Contract) and the other seven MOEs every four months. Hence, the Client's obligation to provide (to TDS) Primary Event Data at four monthly intervals. The first delivery of Primary Event Data is due 1 July 1996. One of the fundamental requirements of OARRS is to be able to calculate and report the eight MOEs for a given system/equipment by Ship or on a Class basis.

The physical organisation of OARRS data is directly affected by a number of aspects, namely:

- system and equipment configuration for Ships 01 and 03 is different to that of Ships 02 and 04 (i.e. Ships 01 and 03 are being built for the RAN and Ships 02 and 04 for the RNZN),
- system and equipment configuration changes are also possible between Ships 01 and 03 and/or 02 and 04 (conceivably due to production and/or engineering changes),
- the calculation of achieved A_0 is directly related to a critical system's associated ABDs, that are directly related to the system/equipment configuration.

The high level implementation of OARRS considers these three aspects together with the fundamental requirement of OARRS to be able to calculate and report MOEs for a given system/equipment by Ship or on a Class basis.

4.2 The Two-Component Database Implementation

The two-component database structure (see Figure 11) separates the tables, table relationships and data (referred to as the DATA component) from the forms, queries, macros, Access Basic modules and other application specific components (referred to as the APPLICATION component); the two components together forming the database.

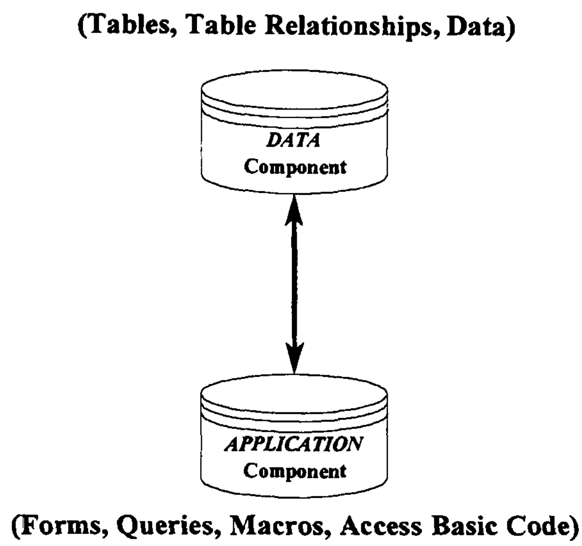


Figure 11 Two-component Database Structure

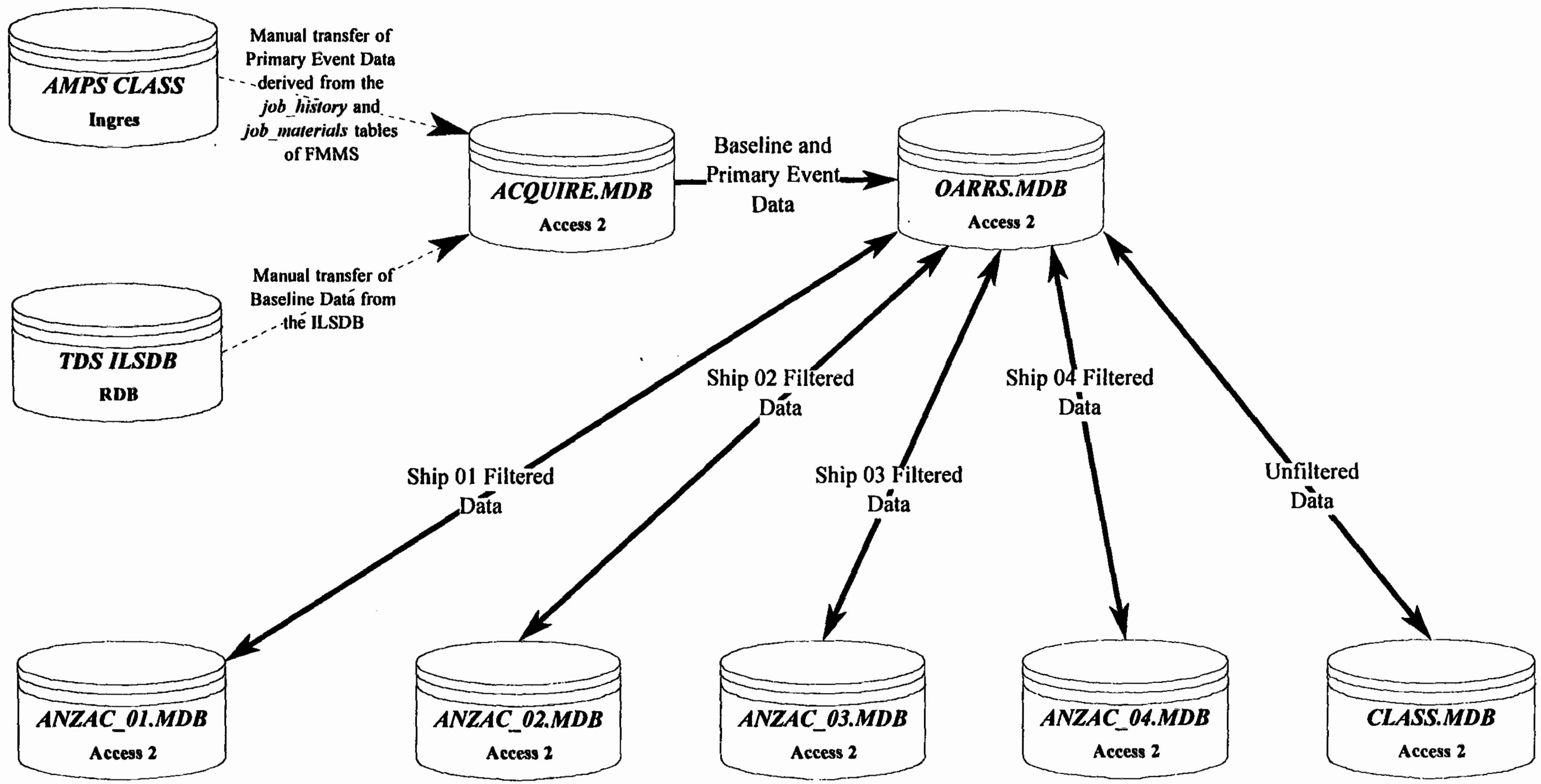
The OARRS Functional Overview (see Figure 12) utilises the two-component database concept. The OARRS database implementation will consist of a DATA component (OARRS.MDB[13]) containing all tables, table relationships and data (Baseline and Primary Event Data) and six APPLICATION databases; one for each Ship (ANZAC_01.MDB - ANZAC_04.MDB), one for data acquisition (ACQUIRE.MDB) and one for the ANZAC class (CLASS.MDB).

The two-component database implementation has a number of advantages over a single-component database implementation (i.e. table structures, data and application components encapsulated in the one MDB file), namely:

- *application component development/updates/additions can be performed whilst not affecting the data component or other application components.* Given the OARRS implementation strategy and the system/equipment/ABD configuration differences between ships, the

[13] Microsoft Access database filename extension.

OARRS DATA Component (Tables and Data)



OARRS APPLICATION Components (Forms, Queries, Macros, Access Basic Code)

Figure 12 OARRS Functional Overview

- loosely coupled nature of the application components is a significant advantage during the coding phase.
- *sharing of baseline data and primary event data between application databases is achieved.*
- *access to the data component of the database, by other outside databases is easier to implement and control.* For example, another database application developed by TDS requires access to Primary Event Data held in OARRS.MDB. The access required can be provided without disruption to any other application component. Also, the table relationships defined in OARRS.MDB do not have to be redefined in the application component.
- *network performance is increased.* According to Dunning (1995), network performance can be improved by splitting the database and running a locally installed application component whilst the data component remains on the network. With a major portion of the database installed locally (i.e. the application component) a performance increase results from the substantially decreased network traffic.

Microsoft (MS) ACCESS 2.0 will be the OARRS Relational Database Management System (RDBMS) development tool and MS Windows for Workgroups the implementation platform. The OARRS database will be single user and implemented on a stand alone IBM compatible Personal Computer (PC). ACCESS provides to the user an intuitive and easy to use Graphical User Interface (GUI). The programmer also benefits through using ACCESS's objects based, GUI development environment.

4.3 Calculation of Achieved A_o (Contract)

The equation used to calculate MOE_c (see Equation 3) is relatively simple. However, the manipulation of Primary Event data into a format suitable for Equation 3 to be applied, is complex. The overall process (see Figure 9) for the calculation of MOE_c is discussed in Section 3.3.1.3. This Section defines an overlapping Event and the process that derives Effective Events from overlapping Events.

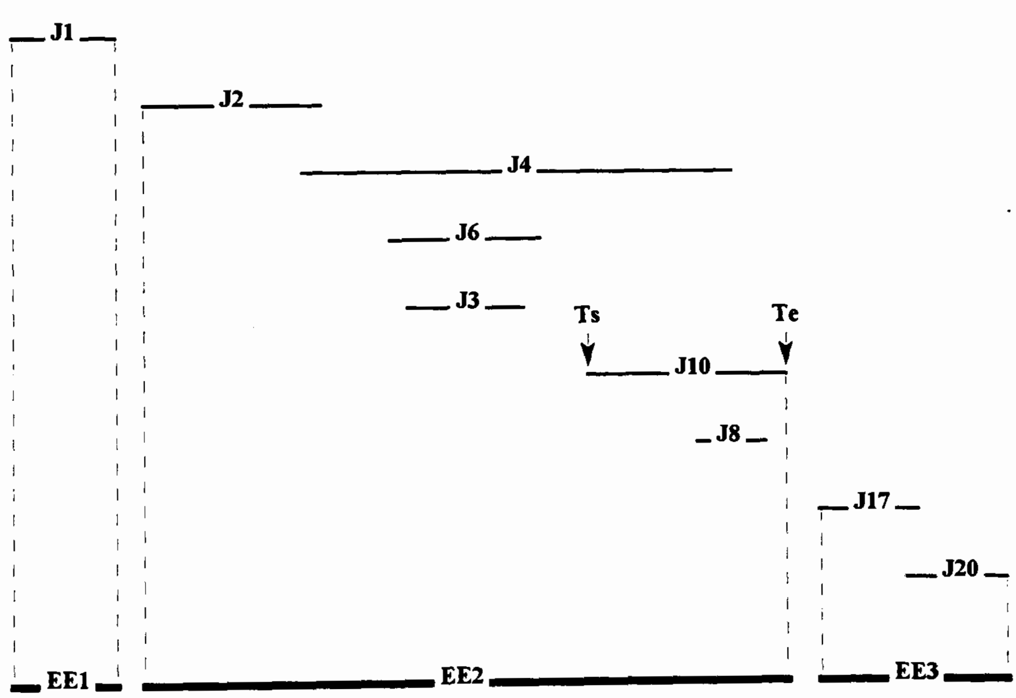
The calculation process begins with the derivation of Effective Events from overlapping Events for Base-level blocks[14]. Once Effective Events have been derived for all base-level blocks, Effective Events for blocks in a redundant configuration are then processed to form an effective block with associated derived Effective Events. Finally, a set of Effective Events are derived for blocks and effective blocks that are in series with each other. Equation 3 is then applied to Effective Events (derived from blocks in series) to produce the MOE_c figure. This process "rolls up" into the critical system ABD to finally produce a MOE_c figure for the critical system.

---

14 Blocks in an ABD may contain within them ABDs of a lower level. A "base-level block" in an ABD is a block that does not contain a lower level ABD. A base-level block in an ABD may contain one or more CMCs.

Figure 13 shows base-level block Events (i.e. FMMS un-planned maintenance jobs) which have been sorted by **contractual_end_date in date_act_start** and arranged graphically to illustrate Event overlap and the derived Effective Events. Consider the following scenario:

- Events J17 and J20 (see Figure 13) are allocated to CMC1 and CMC2 respectively. CMC1 and CMC2 are contained within the Ship's Inertial Navigation System (1) base-level block (see Figure 3). Remembering that base-level blocks contain one or more CMCs.
- Figure 13 clearly shows the overlap that exists between Events J17 and J20, that is, Event J20 starts before Event J17 ends. Hence J17 and J20 are said to be overlapping Events.
- The downtime for both J17 and J20 is 20 mins with a 5 min overlap. Because J17 and J20 overlap by 5 mins the effective downtime resulting from Events J17 and J20 is 35 mins, not 40 mins.
- Therefore, an Effective Event (i.e. EE3) is derived starting at **J17.date_act_start** and ending at **J20.contractual_end**_date. EE3 is stored in the *tblEffectiveDte* data store while the Events contributing to EE3 are stored in the *tblEffectiveCe* data store (i.e. J17 and J20). Similarly, the Effective Event EE2 results from the overlapping of Events J2, J4, J6, J3, J10 and J8.
- However, Event J1 has no associated overlap and is therefore written directly to the *tblEffectiveDte* data store as EE1, while J1 is stored in the *tblEffectiveCe* data store.



Ts = Job Start Date/Time (**tblEventHistory.date_act_start**)
Te = Job End Date/Time (**tblEventHistory.contractual_end_date**)
J = Job
Job Downtime = Te - Ts
EE = Effective Event
▬ = Graphical representation of an Effective Event
___ = Graphical representation of an un-scheduled maintenance Event (ie

Figure 13 Resolution of overlapping Events in a Base-Level Block

TDS must provide traceability for all Events used in the calculation of MOE_c, hence the requirement to hold jobs that have contributed to an Effective Event. Further, if an RM&A investigation[15] were triggered, TDS would require information from the Effective Event audit trail to conduct the investigation. The above example demonstrates the process of deriving Effective Events for base-level blocks.

The next step in the calculation of MOE_c, is deriving Effective Events for redundant configurations. Figure 14 shows four base-level blocks (A, B, C and D) in a redundant configuration (i.e. parallel). Effective Events for the four blocks are grouped and arranged graphically to illustrate overlap and the resultant Effective Events. Figure 14 demonstrates the different processing required to produce Effective Events from overlapping Effective Events associated with redundant configurations.
In this example, the redundant configuration requires two out of the four blocks to be working for the system to be considered operational (up), therefore three or four Events must overlap to produce an Effective (downtime) Event. Consider the following scenario:

- Effective Events AJ2, CJ6 and BJ8 overlap resulting in Effective Event One (EE1) with a starting date of **BJ8.date_act_start** and an end date of **AJ2.contractual_end_date**. EE1 is held in the *tblEffectiveDte* data store while the Effective Events contributing to EE1 are stored in the *tblEffectiveCe* data store (i.e. AJ2, CJ6 and BJ8).
- Similarly, EE2 is derived from the overlapping of Effective Events CJ6, BJ8 and DJ1. Note also the period of time between the end of EE1 and the start of EE2. During this period only two Events overlap (i.e. CJ6 and BJ8), therefore the system remains up and no Effective Event is created.

The last step in the calculation process deals with effective blocks and blocks that are in series with each other. The process resolves overlapping Effective Events using the same method described in the first scenario (see Figure 13). Derived Effective Events for series blocks are held in the *tblEffectiveDte* data store and the contributing Effective Events are held in the *tblEffectiveCe* data store. Equation 3 is applied to the Effective Events held in the *tblEffectiveDte* data store to produce the MOE_c figure.

4.4 OARRS Implementation Schedule

The OARRS implementation schedule has been divided into three phases: **Phase I** which began 11 December 1995 and is due to conclude 31 March 1996, **Phase II** which is due to start 1 April 1996 and conclude 1 June 1996 and **Phase III** which at the time this paper was submitted had not been defined in terms of a start and completion date. The OARRS implementation was divided in this manner to facilitate the completion of the contractually required aspects of OARRS [i.e. calculation and reporting of Achieved A_o (Contract)] two months prior to the first delivery of Primary Event data. The early

---

15 An RM&A investigation would be triggered if the calculated achieved A_o failed to meet the specified A_o threshold value.

completion of Phase I allows testing of the OARRS APPLICATION component to be performed using live FMMS Event data, thereby reducing possible data import and calculation errors when using live data. The three phases are described in more detail below.

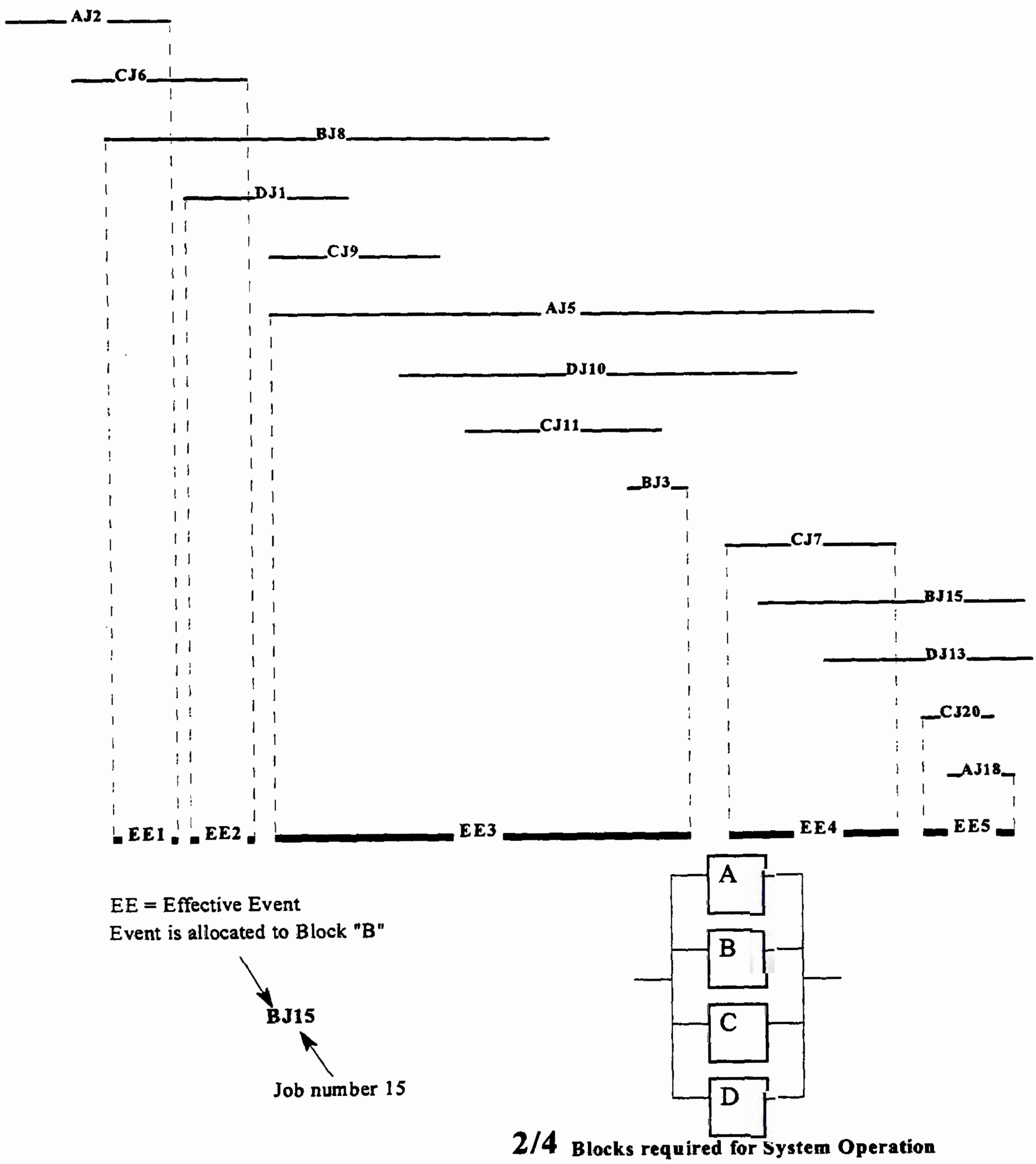


Figure 14 Deriving Effective Events for Redundant Configurations

Phase I **- Development of OARRS.MDB and Selected Modules of ANZAC_01.MDB and ACQUIRE.MDB** and includes the following components:

- Import of Baseline and Event data,
- Processing of Event data,
- Calculation of achieved A_o (Contract), achieved A_o (Actual) and MTBF,
- Produce MOE reports for achieved A_o (Contract), achieved A_o (Actual) and MTBF,

- Data Entry and Data View Forms,
- Application structure (i.e. menus etc),
- System and Acceptance testing,
- Implementation.

Phase II - **Complete the development of ANZAC_01.MDB and ACQUIRE.MDB** and includes the following components:

- Phase II system design,
- Database enhancements,
- Import of Supplementary data (format not known),
- Calculation of Elapsed Time ratio for PM, TTR ratio, LDT ratio, ADT ratio and Equipment Operating Hours ratio,
- Produce MOE reports for Elapsed Time ratio for PM, TTR ratio, LDT ratio, ADT ratio and Equipment Operating Hours ratio,
- Data Entry and Data View Forms,
- Security,
- System and Acceptance testing,
- Implementation.

Phase III - **Development of ANZAC_02.MDB to ANZAC_04.MDB and CLASS.MDB**

- Phase III commencement and completion dates and scope of work are yet to be decided.

5. CONCLUSION

One of the main objectives of the paper was to demonstrate for a very complex system how to define, capture and store information to produce reports on the evaluation of Measures of Effectiveness. The effort (in data element definition) focused on the contractually required MOE_c. The analysis of the data required to calculate MOE_c found it beneficial to divide the data into three categories based on the source and the relevance of the data in determining MOE_c (i.e. Baseline Data, Primary and Supplementary Event Data).

At this time the seven other MOEs, whilst utilising the same Primary Event data used to calculate MOE_c, have not been fully analysed. The analysis of the seven other MOEs is a priority task that is to be performed during Phase II of the OARRS implementation schedule.

Another major objective was the design of a database application that will hold the collected data and be able to calculate the achieved A_o (Contract) MOE, the seven other MOEs and generate hard-copy MOE reports. Yourdon's Essential model has been adopted as the OARRS analysis and design methodology. The OARRS Phase I analysis and design has been completed leaving Phase II as an area for future work.

The implementation of Phase I with respect to the "coding" of the OARRS APPLICATION and DATA components and development of a Graphical User Interface is progressing.

Analysis of the collected Event data with a view to developing meaningful representations/presentations and extrapolations of the data is an area that would complement the work reported thus far.

REFERENCES

Australian Government Department of Defence ANZAC Ship Project (ASP) Contract C218318.

Balestreri S. (1985), *The Component Event Data Bank and Its Uses*, Reliability Data Bases, Proceedings of the ISPRA Course held at the Joint Research Centre, Ispra, Italy, 21-25 October 1985 in collaboration with EuReDatA. Edited by Aniello Amendola and Alfred Z. Keller, D. Reidel Publishing Company.

Bello G. C. (1985), *Data Sources and Collection Procedures*, Reliability Data Bases, Proceedings of the ISPRA Course held at the Joint Research Centre, Ispra, Italy, 21-25 October 1985 in collaboration with EuReDatA. Edited by Aniello Amendola and Alfred Z. Keller, D. Reidel Publishing Company.

Blanchard B. S. (1992), *Logistics Engineering and Management 4th Edition*, Prentice-Hall International Editions.

Bendell T. (1988), *An Overview of Collection, Analysis, and Application of Reliability Data in the Process Industries*, IEEE Transactions on Reliability, Vol. 37, No. 2, 1988 June.

CMC Project, *RAN Common Management Code Handbook*, CMC Project, PO Box E33, Queen Victoria Tce, Canberra ACT 2600.

DOD, Nov (1983), *Acquisition and Management of Integrated Logistic Support for Systems and Equipment*, DOD Directive 5000.39, Washington, D.C.

Dunning M.D. (1995), *Splitting Databases*, Access/Visual Basic Advisor - June 1995, Advisor Publications Inc.

Hall B. and Shelley A. J. (1995), *ILS Data Recording and Reporting Procedures*, AC11.052 Revision 2, Transfield Defence Systems, Williamstown, Australia.

Yourdon E. (1989), *Modern Structured Analysis*, Prentice-Hall International Editions.

APPENDIX A. EXTERNAL INTERFACE COMPONENTS

TABLE	KEY	COLUMN	DATA TYPE	DESCRIPTION
job_history	PK	job_no	X(8)	FMMS Job Number.
	PK	sequence_no	9(10)	A number to identify the specific job activity where multiple activities exist for the job.
		std_activity_no	X(8)	FMMS Standard Activity Number.
		job_title	X(80)	Event Description.
		facility_code	X(30)	CMC to which the Job has been allocated.
		shop_id	X(4)	A valid shop_id as held in the AMPS shop table.
		activity_type	X(4)	Job is Mandatory or Non-mandatory.
		activity_cause	X(4)	Job is Scheduled or Un-scheduled Maintenance.
		job_status	X(2)	An alphanumeric code that identifies the status of this event (should be "C" for completed).
		originator	X(8)	A valid originator code as held in the AMPS originator table.
		date_raised	D	FMMS generated, date/time the event was raised.
		date_est_start	D	USER entered, date/time the event is estimated to start.
		date_est_end	D	USER entered, date/time the event is estimated to end.
		date_act_start	D	Date/time the Job was Actually Started.
		date_act_end	D	Date/time the Job was Actually Finished.
		elapsed_hours_est	9(10)V99	The estimated time to perform the planned maintenance activity; copied from the standard activity.
		elapsed_hours_act	9(10)V99	Actual Time To Repair (TTR).
		job_result	X(4)	Job Result (should be "Completed").
		job_reference	X(16)	If activity_cause = "SCHEDULED" then holds the TDS MRC Code else NULL.
		job_priority	9(2)	A number between 1 and 10 that represents the relative priority of this job. **NOTE: Priority 1 and 2 indicate an URDEF is associated with this job.**
		serial_number	X(24)	Serial number associated with the CMC allocated to this job.

TABLE	KEY	COLUMN	DATA TYPE	DESCRIPTION
		shutdown_facility	X(30)	If NOT NULL equipment is down.
		trigger_reading	9(10)V99	Trigger value (ie operating hours) at the time the event occurred.
job_materials	K1	job_no	X(8)	The job_id of the event that used this spare part/material; job_id must exist in the *job_history* table
	K1	sequence_no	9(10)	A number to identify the specific job activity where multiple activities exist for the job.
		catlog_code	X(12)	NATO Stock Number (NSN).
		warehouse_id	X(8)	UDC of the NAVY Ship, Base or Depot that supplied the spare part/material.
		timestamp	D	FMMS generated, date/time the stores request was raised.
		units_consumed	9(10)V99	The number of units of this spare part/material consumed by the event identified by job_id.
		part_number	X(30)	Manufacturer's part number.
		status	X(2)	An alphanumeric code that identifies the status of this stores request (should be "C" for completed).
		date_consumed	D	The date/time this item was consumed.
		reqsn_reference	X(20)	Reference into the purchasing module (Procurement Automation) of AMPS.
facility_readings	PK	facility_code	X(30)	CMC to which the Job has been allocated.
	PK	trigger_code	X(4)	A code that identifies a valid triggering mechanism.
	PK	location	X(12)	Location of the measuring point on the equipment.
		total_reading	9(10)V99	The all-time number of units read for this equipment and trigger code as at the date/time of entry of this reading.
		latest_reading	9(10)V99	Latest entered reading for this facility_code and trigger_code.
		latest_reading_date	D	Date of the latest entered reading for this facility_code and trigger_code.
		latest_batch_no	X(10)	System-allocated number of the latest batch of meter/condition readings.
		meter_limit	9(10)V99	The rollover point for the meter used for this facility_code and trigger_code.

TABLE	KEY	COLUMN	DATA TYPE	DESCRIPTION
		avge_interval_units	9(10)V99	Running average of intervals, in trigger code units, between successive meter/condition readings.
		avge_interval_days	9(10)V99	Running average of intervals, in days, between the dates of successive readings.
		total_no_readings	9(10)	Total number of readings that have provided the above averages.

Table 2 External Data Dictionary

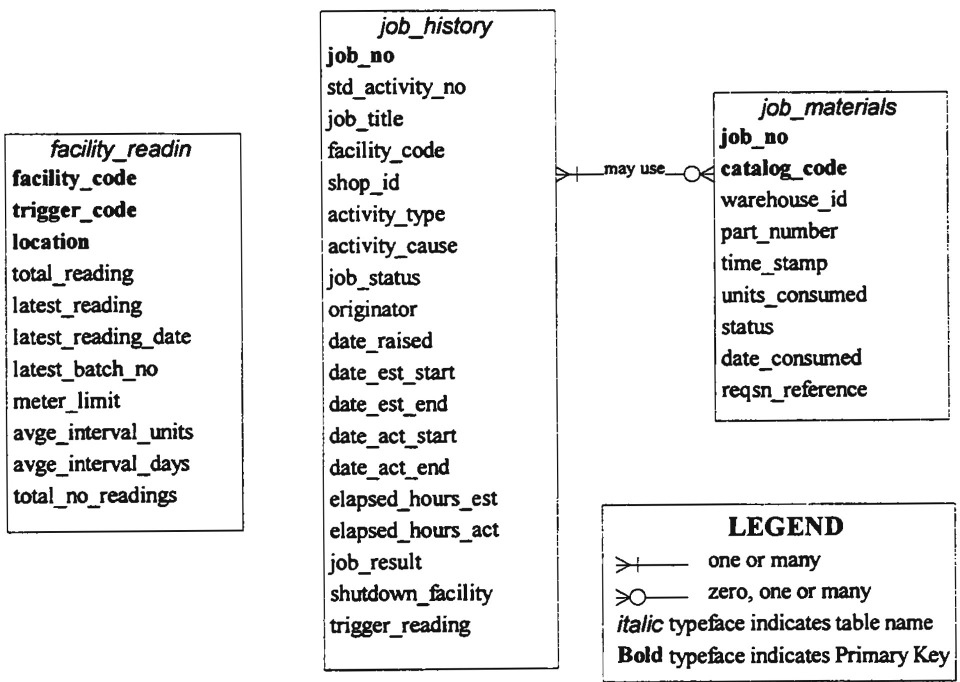
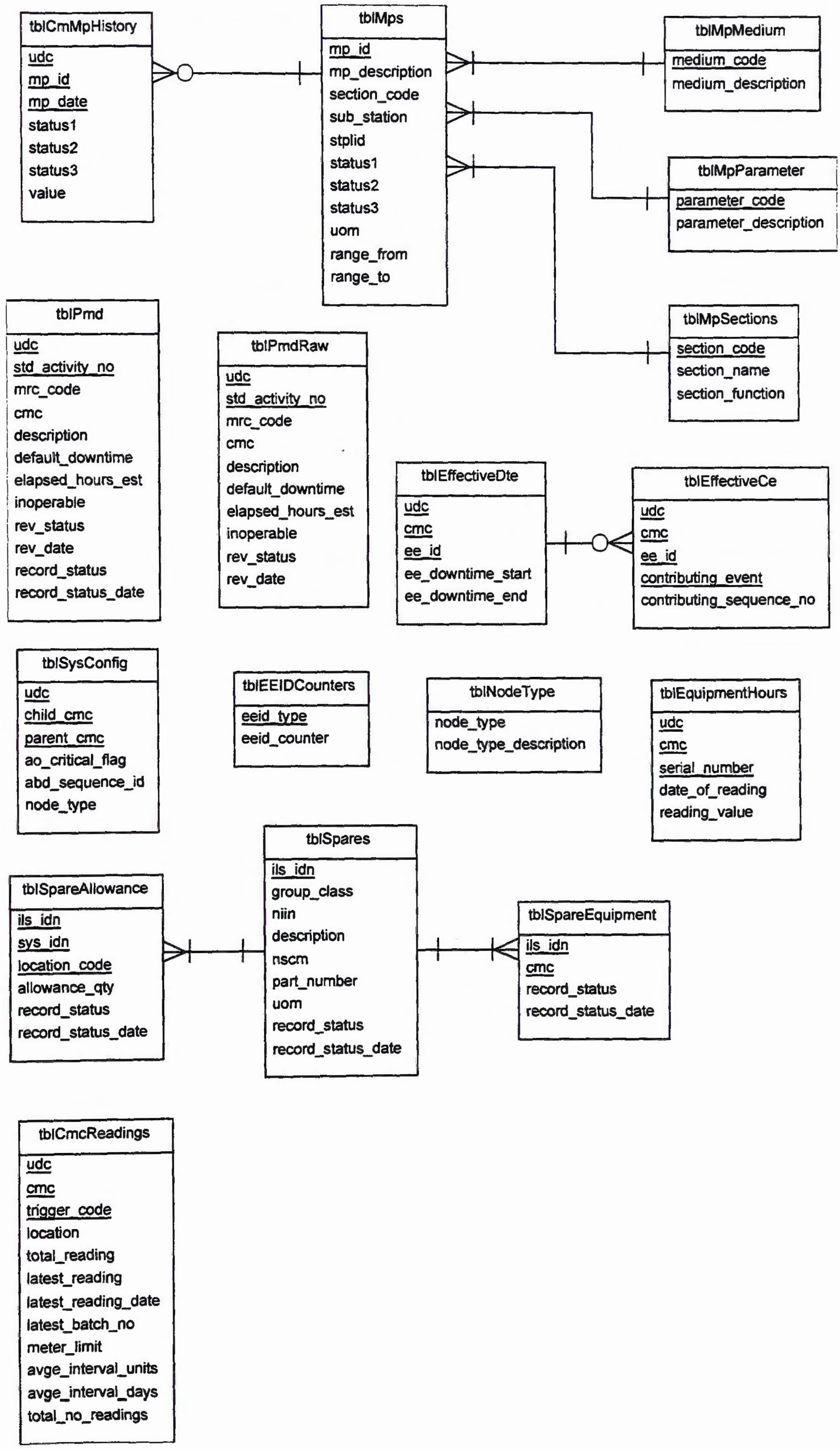
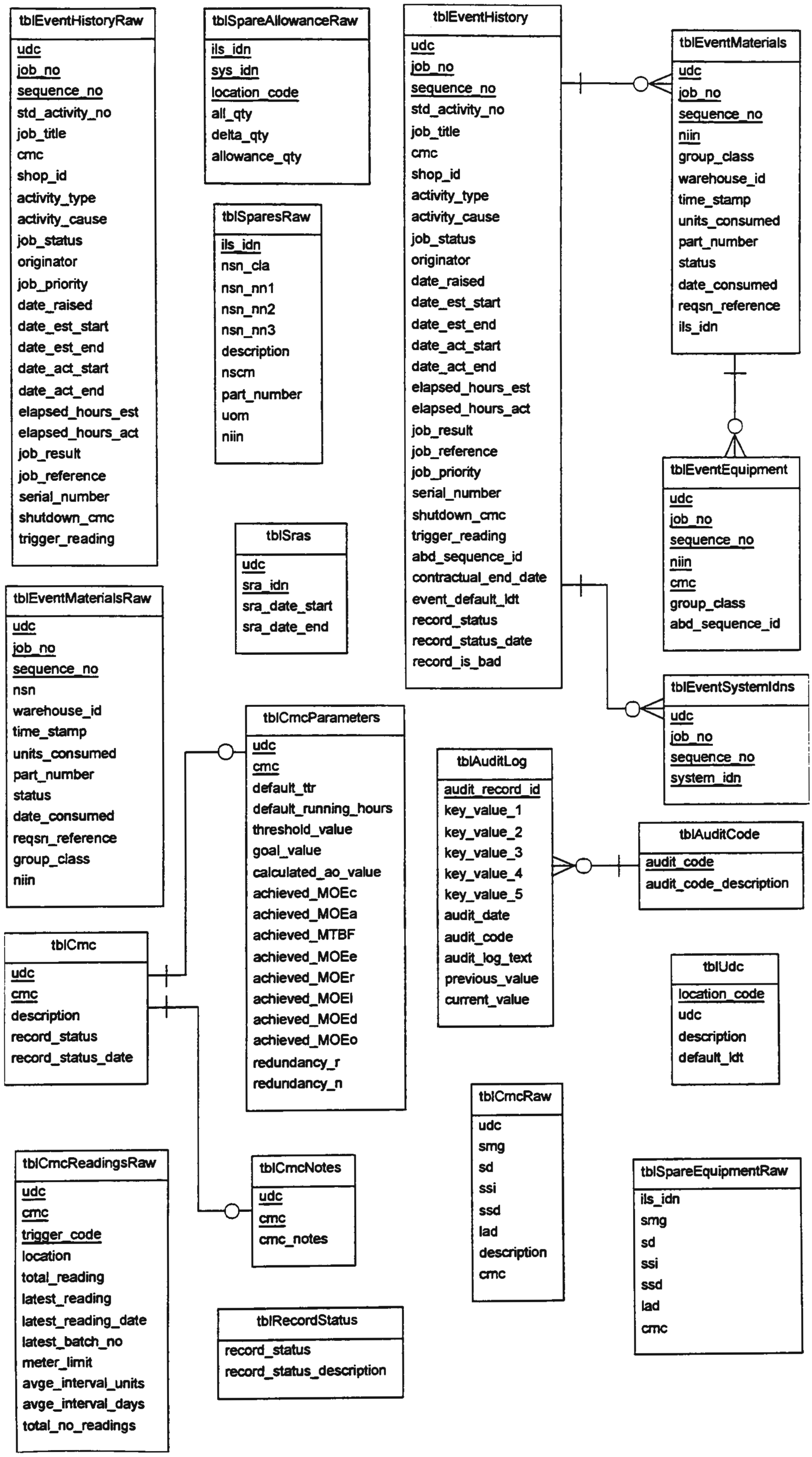


Figure 15 External Data Model

APPENDIX B. OARRS DATA MODEL




tblEventHistoryRaw
udc
job_no
sequence_no
std_activity_no
job_title
cmc
shop_id
activity_type
activity_cause
job_status
originator
job_priority
date_raised
date_est_start
date_est_end
date_act_start
date_act_end
elapsed_hours_est
elapsed_hours_act
job_result
job_reference
serial_number
shutdown_cmc
trigger_reading
tblSpareAllowanceRaw
ils_idn
sys_idn
location_code
all_qty
delta_qty
allowance_qty
tblSparesRaw
ils_idn
nsn_cla
nsn_nn1
nsn_nn2
nsn_nn3
description
nscm
part_number
uom
niin
tblEventHistory
udc
job_no
sequence_no
std_activity_no
job_title
cmc
shop_id
activity_type
activity_cause
job_status
originator
date_raised
date_est_start
date_est_end
date_act_start
date_act_end
elapsed_hours_est
elapsed_hours_act
job_result
job_reference
job_priority
serial_number
shutdown_cmc
trigger_reading
abd_sequence_id
contractual_end_date
event_default_ldt
record_status
record_status_date
record_is_bad
tblEventMaterials
udc
job_no
sequence_no
niin
group_class
warehouse_id
time_stamp
units_consumed
part_number
status
date_consumed
reqsn_reference
ils_idn
tblEventEquipment
udc
job_no
sequence_no
niin
cmc
group_class
abd_sequence_id
tblSras
udc
sra_idn
sra_date_start
sra_date_end
tblEventMaterialsRaw
udc
job_no
sequence_no
nsn
warehouse_id
time_stamp
units_consumed
part_number
status
date_consumed
reqsn_reference
group_class
niin
tblEventSystemIdns
udc
job_no
sequence_no
system_idn
tblCmcParameters
udc
cmc
default_ttr
default_running_hours
threshold_value
goal_value
calculated_ao_value
achieved_MOEc
achieved_MOEa
achieved_MTBF
achieved_MOEe
achieved_MOEr
achieved_MOEl
achieved_MOEd
achieved_MOEo
redundancy_r
redundancy_n
tblAuditLog
audit_record_id
key_value_1
key_value_2
key_value_3
key_value_4
key_value_5
audit_date
audit_code
audit_log_text
previous_value
current_value
tblAuditCode
audit_code
audit_code_description
tblCmc
udc
cmc
description
record_status
record_status_date
tblUdc
location_code
udc
description
default_ldt
tblCmcRaw
udc
smg
sd
ssi
ssd
lad
description
cmc
tblCmcReadingsRaw
udc
cmc
trigger_code
location
total_reading
latest_reading
latest_reading_date
latest_batch_no
meter_limit
avge_interval_units
avge_interval_days
total_no_readings
tblCmcNotes
udc
cmc
cmc_notes
tblRecordStatus
record_status
record_status_description
tblSpareEquipmentRaw
ils_idn
smg
sd
ssi
ssd
lad
cmc