Task (2)

JAN 27 2000

ENGINEERING DATA TRANSMITTAL

Page 1 of 1

1 EDT 627258

2 To (Receiving Organization)	3 From (Originating Organization)	4 Related EDT No
Distribution	Process Engineering	N/A
5 Proj /Prog /Dept /Div	**6 Design Authority/ Design Agent/Cog Engr**	**7 Purchase Order No**
Tank Monitor and Control System (TMACS)	M J Holm	N/A
8 Originator Remarks		**9 Equip /Component No**
This document describes the software design for the Tank Monitor and Control System (TMACS)		N/A
		10 System/Bldg /Facility TMACS/2750E/B105
11 Receiver Remarks 11A Design Baseline Document? [X] Yes [ ] No		**12 Major Assm Dwg No**
For release		N/A
		13 Permit/Permit Application No N/A
		14 Required Response Date

15 DATA TRANSMITTED

(A) Item No	(B) Document/Drawing No	(C) Sheet No	(D) Rev No	(E) Title or Description of Data Transmitted	(F) Approval Desig nator	(G) Reason for Trans mittal	(H) Origi- nator Dispo sition	(I) Receiv er Dispo sition
1	HNF-SD-WM-SDD-079	N/A	0	TMACS As-Built Design Document	ESQ	2	1	1

16 KEY

Approval Designator (F)	Reason for Transmittal (G)	Disposition (H) & (I)
E S Q D or N/A (see WHC CM 3 5 Sec 12 7)	1 Approval 4 Review 2 Release 5 Post Review 3 Information 6 Dist (Receipt Acknow Required)	1 Approved 4 Reviewed no/comment 2 Approved w/comment 5 Reviewed w/comment 3 Disapproved w/comment 6 Receipt acknowledged

17 SIGNATURE/DISTRIBUTION
(See Approval Designator for required signatures)

(G) Reason	(H) Disp	(J) Name	(K) Signature	(L) Date	(M) MSIN
1	1	Design Authority C C Scalef III	C.C. Scalef III	1/12/00	
		Design Agent			
1	1	Cog Eng M J Holm	MJH	1-11-00	
1	1	Cog Mgr J G Field	J G Field	1/24/2000	
1	1	QA T L Bennington	T L Bennington	1/23/2000	
1	1	Safety P L Smith	P L Smith	1/11/00	
1	1	Env P C Miller	P C Miller	1/11/00	
1	1	D A Selle	David A Selle	1/11/00	
1	1	W J Kennedy	W J Kennedy	1/11/00	

18	19	20	21 DOE APPROVAL (if required) Ctrl No
M J Holm 1/24/2000 Signature of EDT Originator Date	N/A Authorized Representative for Receiving Organization Date	J G Field 1/24/2000 Design Authority/ Cognizant Manager Date	[ ] Approved [ ] Approved w/comments [ ] Disapproved w/comments

BD 7400 172 2 (05/96) GEF097

S

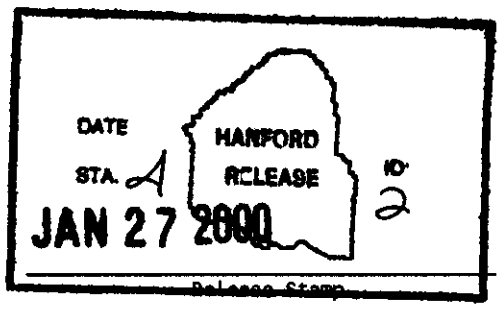
TMACS As-Built Design Document

M J Holm
Lockheed Martin Hanford Corporation, Richland, WA 99352
U S Department of Energy Contract DE-AC06-96RL13200

EDT 627258 UC 605
Org Code 74B00 Charge Code 102608/EB00
B&R Code EW3120071 Total Pages 128

Key Words Design Document, As-Built, TMACS, Tank Monitor and Control System

Abstract This document describes the software design for the Tank Monitor and Control System (TMACS)



1/27/00
Release Approval Date

Approved for Public Release

Tank Monitor and Control System (TMACS)
Software Design Document
As -Built

Authors

R Wayne Little
Battelle – PNNL
Updated 7/3/97

Paul M Jackson – Parsons
Last Update 11/10/1999

Table of Contents

Table of Figures

Table of Tables

1 INTRODUCTION

1 1 Purpose

This document captures the existing as-built design of TMACS as of November 1999 It will be used as a reference document to the system maintainers who will be maintaining and modifying the TMACS functions as necessary

The heart of the TMACS system is the "point-processing" functionality where a sample value is received from the field sensors and the value is analyzed, logged, or alarmed as required This Software Design Document focuses on the point-processing functions

1 2 Scope

The scope of this document is limited to the G2[1] component of the TMACS software, GSI[2] and driver internals are not a part of this document This document focuses on the existing functionality in the TMACS production system

1 3 Overview

The TMACS was originally designed in response to recommendations from the Defense Nuclear Facilities Safety Board on improved monitoring of waste tank temperatures It has since been expanded to monitor analog (continuous) values for temperature, surface level, tank pressure, ventilation flow, and hydrogen gas concentration "Discrete parameters or alarms of various types are also monitored TMACS, through Acromag[3], ENRAF[4], Westronics[6], and Panalarm[5] interfaces, has the capability to monitor most standard instrument and sensor types

1 4 Definitions and Acronyms

Acromag®	A commercially available I/O processor for analog and discrete signals Each Acromag® station has multiple channels, each of which can store a sensor reading

[1] G2 is a registered trademark of the Gensym® Corporation
[2] GSI is a registered trademark of the Gensym® Corporation
[3] Acromag is a registered trademark of Acromag, Incorporated
[4] ENRAF is a registered trademark of Enraf B V
[5] Panalarm is a registered trademark of Ametek Inc
[6] Westronics is a registered trademark and they are a subsidiary of ONIX Systems, Inc

Actor

Actors represent the things that interact with a system They represent everything that needs to exchange information with the system Typically actors are users and other computer systems or equipment See also use case

Alarm

A predefined condition that signals a warning or alert

Alarm annotation

For discrete points, each value is configured with an alarm state, annotation and display color The annotation provides the meaning of a discrete sensor's value, e g

Discrete Value	Alarm State	Annotation	Display Color
0	Normal	Open	Green
1	Normal	Closed	Green
2	Alarm	Opening	Red
3	Alarm	Closing	Yellow

Alarm limits

Limits at which a sensor's reading will trigger an alarm, continuous sensors typically have a low alarm limit and a high alarm limit TMACS currently has implemented only one high and one low alarm per continuous sensor, these are typically used for surveillance limits [temperature and level] or safety limits [pressure %H2])

Alarm processing

The subset of point processing that handles alarms This includes producing alarm messages, annunciating alarms, logging the alarms and handling acknowledgments For discrete points, each discrete value has an alarm state associated with it For continuous points the point value is compared against alarm limits Continuous points have a further subset of alarm processing called rate-of-change (ROC) alarm processing, see ROC See also alarm state and alarm type

Alarm state

Used to indicate the current status of an alarm, valid alarm states include

Alarm State	Meaning	Example
Normal	Able to communicate with sensor, not an alarm	Normal operation
Alarm	Able to communicate with sensor, in alarm	Temperature of 110 degrees, high alarm limit of 109
Unknown	Unable to communicate with sensor, or just initialized	Power failure, broken modem, reading out of instrument limits, etc

Alarm State	Meaning	Example
Disable (FUTURE)	Alarm processing turned off by an operator (user) [6]	Sensor disabled for repair
Inhibit (FUTURE)	Alarm processing turned off automatically via logic-computer control	When a valve is shut, computer automatically inhibits the related low pressure alarm

Alarm type — Used to indicate the kind of alarm for continuous i/o points only, the types of alarms include

Alarm Type	Code	Meaning
Alarm-High	AH	Value greater than high-alarm limit
Alarm-Low	AL	Value less than low-alarm limit
Rate-of-Change	ROC	The rate of change exceeded the allowable limit
No-Alarm	NA	Normal not in alarm Value within limits

Analog sensor — See continuous sensor

Annotation — The text describing the meaning of an alarm state Particularly used for discrete sensors

Annunciate — Communicate information to the user in a combination of ways - e g visually (text, color, flashing), audibly (beeping) Annunciate is an inclusive term intended to imply everything necessary to communicate the information It is particularly more than just displaying information, and normally used in the context of annunciating an alarm state, where information is provided in several places and ways (alarm message text and color, color of sensor display, color of related tank display, audibly, etc)

Bridge — A bridge is an io-driver particular to Gensym® software that controls the communication to the io-stations There is one bridge for each brand of io-station to be interfaced with TMACS At the point of this writing, bridges have been developed for the Acromag® 4000, the Enraf® CIU ,Westronics® and the Panalarm® Series 90 Annunciator The bridges communicate with the io-stations and their channels via a serial interface, and pass the readings back to the G2® portion of the software as io-point readings

[6] Although the enabling and disabling of alarm-processing is implemented, it is done through toggling a Boolean expression, *alarm-processing-enabled*, and is not currently reflected in the alarm-state

Channel	The part of an I/O station that receives direct readings from sensors e g For an analog device, this direct reading could represent milliamps A channel reading may need to be converted to a value
CIU	Communications Interface Unit A digital interface between the Enraf® surface level gauges and TMACS Created by Enraf®, the surface level gauge vendor
Configuration vector	Consists of the processing configuration for a discrete point which includes instrument limits, alarm limits, alarm color and annotation, annotations, and delta
Continuous point	Used synonymously with continuous sensor
Continuous sensor	An analog input point which measure a value that can vary continuously within some range, e g , a temperature sensor
Current alarm list	The collection/list of all currently outstanding alarms in TMACS The current-alarm-list is a list of all alarms that have either not been acknowledged or not been reset or both
Current value	The present value and quality status of a point
Deadband	The amount a continuous point value must change once it goes into alarm *in order for the point to go out of the alarm state and be reset* It is used to stop alarm chatter In TMACS it is commonly set to twice the delta band For a high-alarm, this will be the amount the point must decrease below the high-alarm-limit to reset for a low-alarm the amount it must increase above the low-alarm-limit before the alarm resets
Delta	The amount a continuous point's value must change to be considered a new or different reading
Deltaband	The range of values equal to the last recorded value ± the delta A reading within this range is not considered a change from the last recorded value
DFD	Data Flow Diagram A structured design diagram which shows processes, data stores, and data flows between them
Disable	A point or function is said to be disabled when processing is turned off by an operator (user) See also alarm state and contrast with inhibit
Discrete sensor	An input point with a pre-defined number of distinct integer values representing states For example, an alarm point is either ON or OFF
Discrete state configuration	The discrete-state configuration describes the full meaning of a particular valid value for a particular discrete io-point The set of discrete-state

configuration	valid value for a particular discrete io-point The set of discrete-state configurations give the full range of operation and meaning for a particular discrete io-point This is defined prior to putting a discrete io-point into service, and is used by point processing and alarm-processing
DST	Double Shell Tank A tank with an inner and outer shell The DST icons have a double circle around them in TMACS See also SST
Enraf®	A particular brand of level gauge/sensor used in TMACS to monitor the surface level in tanks Enraf® level gauges have a 4-20 ma analog output representing the measured level or a digital output of the measured level that passes through a CIU io station
Event	A stimulus that invokes an action, in a state transition diagram, a transition from one state to another is initiated by an event In TMACS, the Mother of All Events is an I/O channel acquiring a value at a time t2 > t1 (previous time) This is the event that fires the Point-Processing Whenever Rule
G2®	A programming language and software system provided by Gensym® Corporation that a large portion of TMACS is implemented in
Gensym®	The makers of the G2® software that TMACS is implemented in Gensym® has also provided some of the bridges that TMACS uses to talk to io-stations and databases
Historical value	A value (reading) older in time than the current value
HLAN	Hanford Local Area Network The network that covers the Hanford site, allowing connected computers to "talk to each other" The main TMACS and SACS computers are on a special sub-net of the HLAN
Inhibit	A point or function is inhibited when processing is automatically turned off by computer control logic See also alarm state and contrast with disable
Instrument limits	The limits that define valid readings for a device Values outside of this range are considered invalid readings
I/O, I-O, IO	Input/Output
io-device reading	The value read from an io-device The io-devices often perform some hardware conversion of the raw reading from the sensor See also raw reading and reading
Io-point	The logical representation of a field instrument sensor, or device that

Io-station	See station
Last good reading	The timestamp and value of a point that represent the last reading with a quality status of GOOD or TEST
Logging	The subset of point processing that stores a sensor reading outside of TMACS, e g to a disk file or database Logging can also done for sensor alarms, equipment alarms and important user actions (e g enable/disable point-processing for a point)
Method	An object-oriented programming feature added in version 4 to G2® A method is a specialized form of a procedure that is specific to an object class It, like an object's data, can be inherited and/or over-ridden in a subclass At this point, these are primarily used in the tank-constructor tool
Monitored system	A monitored system is a generalization of an item of interest that is monitored by io-point sensors Monitored systems can be specific items, like a tank or a collection of things related to a project (e g C-106 Sluicing, K-Basins) or another physical entity (e g AY/AZ Exhauster)
ODBC	A standard data base front-end access language Kind of a ' universal" data base language standard A SQL/ODBC bridge is used to communicate from TMACS to SACS
Panalarm®	A specialized io-station that is a particular brand of alarm panel used in the tank farms TMACS communicates with the Panalarm® Series 90 Annunciator A Panalarm® connects to and annunciates individual alarm sensors
Point	Used synonymously with I/O point and sensor
Point processing	Point processing is a term representing everything that is done (all functions processing, display, logging, etc) for sensor readings It has sub-components of alarm processing, rate of change processing(roc), test processing and logging These sub-components may be selectively disabled within TMACS
Procedure	The G2® provided equivalent of a subroutine Procedures can be called (synchronous - wait for return to continue) or started (asynchronous - continue with next action and statement while this procedure executes)
Quality state	The representation of the "goodness" or validity of a particular point's value

Quality State Name	Meaning	Example
Good	Communicating with	Normal operation

Quality State Name	Meaning	Example
	sensor and reading is within expected instrument limits	
Unknown	Don't have good reading	Lost communication (unable to get reading) Reading is out of instrument limits
Testing	Calibration or other testing of sensor in progress	NOT implemented at this point

Raw reading — The sensor reading in instrument units (e g millivolts) The raw reading becomes an io-device reading, which in turn becomes a reading For example, a thermocouple sensor has a raw reading in millivolts, this is converted to degrees Celsius times 10 (ten) at the channel (the io-device reading) an finally point processing converts the io-device reading to degrees Fahrenheit via the io-point's formula to arrive at the final reading

Reading — The value in final human understandable units obtained from a sensor at a particular point in time a value + timestamp pair Reading and value are often used synonymously See also raw reading and io-device reading

Reset — Return to alarm state NORMAL from alarm state of ALARM A point is said to "reset" when it goes out of alarm See also deadband

ROC — Rate-of-change A particular kind of alarm based on the rate at which a value is changing (i e greater or less than some amount of change over some amount of time)

Role — A G2® programming facility used to implement security and make certain functionality role-specific Every login to TMACS is assigned a role, which will determine the privileges and run-time behavior of TMACS Roles include t2-user (operators), surveillance (for the surveillance administrator) and administrator (the developer role)

Rule — In G2®, rules are a specific programming facility, which wait for a certain event to happen and then fire off a procedure or do something else as directed TMACS point-processing is based on these

SACS — Surveillance Analysis Computer System SACS is used as the repository for long-term storage and retrieval of TMACS data

Sensor — A sensor is a device used to measure a physical characteristic or state of

	an item of interest Common sensor types are surface level and temperature Sensor is used synonymously with point and i/o point
Sensor reading	*See reading*
SHMS	Standard Hydrogen Monitoring System A particular subsystem for monitoring gas related values - e g percent hydrogen, flow and trouble alarms These are connected to Acromag® io-stations in TMACS
SQL	Structured Query Language A language used to access information in a relational data base In TMACS, a SQL/ODBC bridge communicates with the SACS database
SST	Single Shell Tank These are tanks with a single layer of containment The SST icons have a single circle around them in TMACS See also DST
State	A state is a static condition of an object during which the object is receptive to new events
State transition	The transfer from one state to another based upon a particular event See also STD
Station	An I/O device that collects input signals from sensors The station has one or more channels which hold the io-device readings before they are passed on to the G2® portion of TMACS
Subworkspace	A "hidden" window intrinsically attached to almost every object in G2® A common paradigm employed in G2® is that clicking on an object will navigate to a subworkspace behind the object
STD	State Transition Diagram An STD is a compact notation that shows dynamic behavior by illustrating the possible states that may occur and indicating the events that will cause a transition between those states
Tank farm	A tank-farm is a physical area where a set of underground storage tanks are located The 200 West area contains tank-farms S, SX, SY, T, TX, TY and U The 200 East area contains tank-farms A, AN, AP, AX AY, AW AZ, B, BX, BY and C
TC	Thermocouple A thermocouple is a temperature sensor
Thermocouple	See TC
TMACS	Tank Monitor and Control System
Use case	Use cases are a strategy for partitioning the requirements of a system according to a user view, according to the major input events to the

	system from a user perspective According to Ivar Jacobson, the creator of use cases, "A use case model uses actors and use cases These concepts are simply an aid to defining what exists outside the system (actors) and what should be performed by the system (use cases) "
Valid	A reading is said to be valid when it is a legal good reading • For continuous sensors, valid readings are those that fall within the specified instrument limits and invalid readings are those that exceed the specified instrument limits • For discrete points, invalid readings are any reading other than values specified in the configuration
Value	The actual value of a point in some human understandable units, e g 10°F For discrete points, a value is a numerical representation of a state Value is often used interchangeably with reading, although technically, value is slightly more specific See current value, historical value, last good reading, and reading
Westronics®	A commercially available I/O processor for analog and discrete signals Each Westronics® station has multiple channels, each of which can store a sensor reading
Workspace	The fundamental building block provided by G2® for building a program Basically it's a window on a screen where you can put stuff The main workspaces are named Everything permanent in TMACS needs to live on some workspace

2 REFERENCES

2 1 TMACS Specific Documentation

This section contains references to other documents explicitly written about some aspect of TMACS Include for each Title, Author, Document Number, Description/Purpose

TMACS System Description
Author Chuck Scaief
Document Number WHC-SD-WM-TI-671
Description Excellent overview of TMACS Read this first for a good introduction This also contains the details of all the hardware devices employed by TMACS, including sensors, io-stations and communication equipment This also contains nice color pictures of the equipment and the TMACS screens

TMACS Functional Requirements
Author Ron Wandling
Document Number HNF-SD-WM-SFR-006
Description Early documentation of some specific requirements for TMACS

TMACS Input/Output (I/O) Termination Point Listing (The Tag List)
Author TWRS Operations Design Authority
Document Number WHC-SD-WM-TI-594
Description The definitive listing of all io-point sensor and other io devices associated with TMACS It also lists sensors that are scheduled for connection in the near future The actual connection status is indicated so that one may discern which sensors are presently being monitored

TMACS Device Configuration Listing
Author Steven C Cantrell
Document Number WHC-SD-WM-TI-717
Description This document contains the listings for Acromag Terminal Unit's configuration files associated with the Tank Monitor and Control System (TMACS)

Master Drawing List H-2-81778
Author Master Drawing List
Document Number (Drawing Number) H-2-81778
Description This contains a listing of ALL the engineering drawings for the original TMACS, including all the field instrumentation

TMACS Performance Analysis Under Steady State Conditions
Authors Bob Bass and Steve Washburn
Document Number WHC-SD-WM-ANAL-048 Rev 0

Description	An in depth analysis of TMACS performance for the purpose of seeing how many more sensors the current system can support (a lot more), and what it would take to handle all sensors for all tanks (a bigger CPU, but doable)

2 2 External References

2 2 1 Gensym® Documentation

G2® Reference Manual, Version 4 0, 1995
G2® System Procedures Reference Manual Version 4 0, 1995

2 2 2 Object-Oriented Technology References

Object-Oriented Software Engineering A Use Case Driven Approach

Authors	Ivar Jacobson
Document Number	N/A
Publisher	Addison-Wesley ACM Press, 1993 (revised)
Description	The definitive book on Use Cases Good in depth introduction to OO concepts

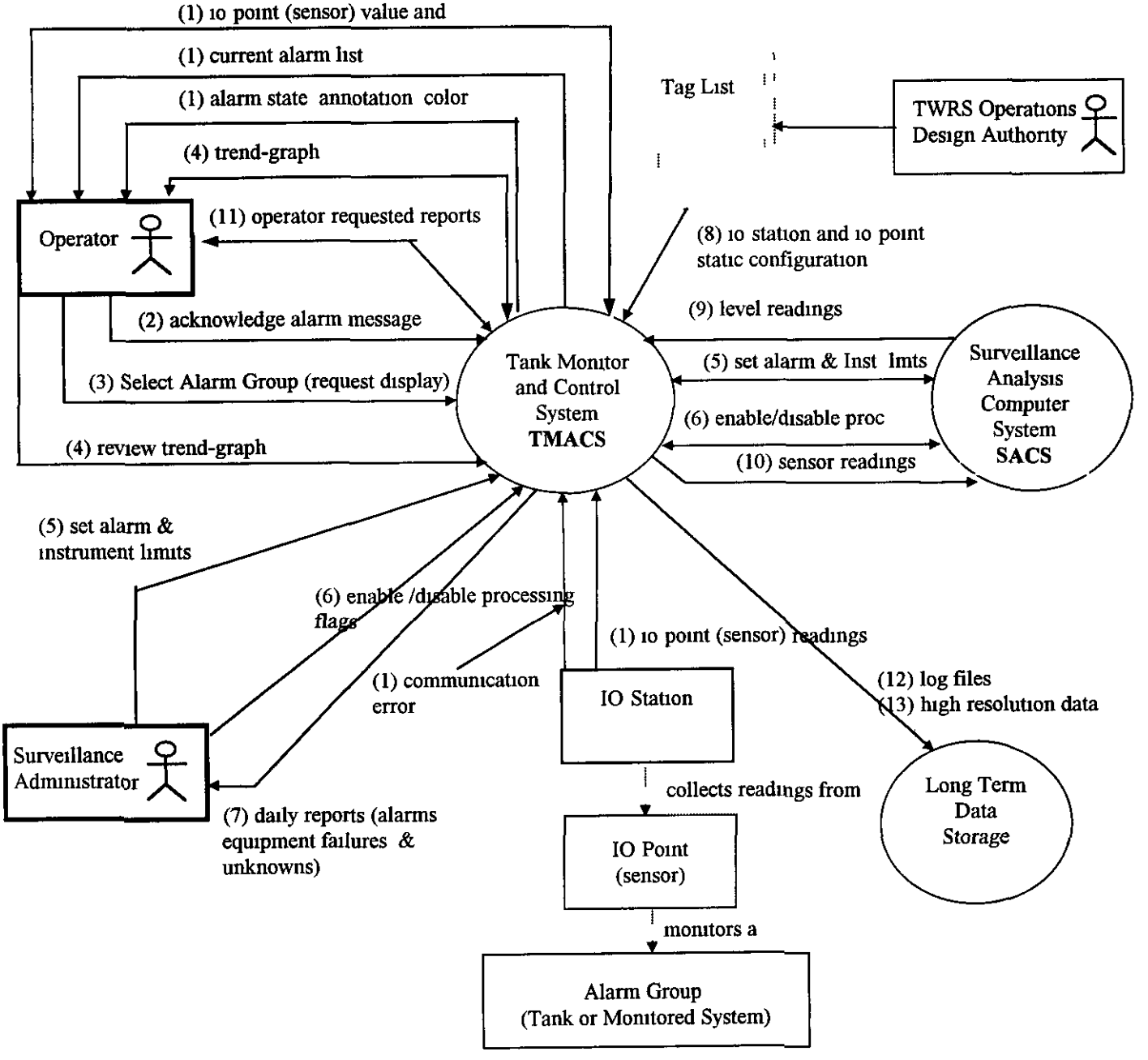
Real-Time Object-Oriented Modeling

Authors	Brian Selic, Garth Gullekson, and Paul T Ward
Document Number	Can borrow this one from the PNNL Technical Library
Publisher	Wiley Processional Computing, 1994
Description	A resource for some real-time modeling methods and diagrams e g State Transition Diagrams

3 SYSTEM OVERVIEW AND CONTEXT

Figure 1 TMACS Context Diagram shows the context of TMACS with external actors/interfaces and major data flows It also shows the mapping of the context diagram to the Use Cases, which are presented in the following section The number in parentheses is the number of the corresponding use case

Figure 1 TMACS Context Diagram



4 USE CASE MODEL

4 1 Introduction To Use Cases

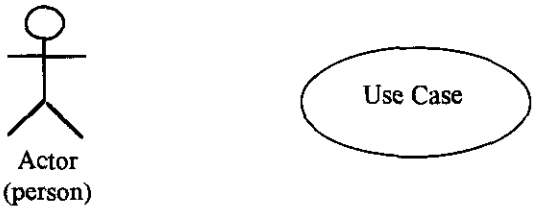
Only a very brief introduction to the subject is provided here a full treatment of Use Cases is beyond the scope of this document For a full treatment of use cases see *Object-Oriented Software Engineering - A Use Case Driven Approach* by Ivar Jacobson

Use cases are a strategy for partitioning the requirements of a system according to a user view, according to the major input events to the system from a user perspective According to Jacobson, 'A use case model uses actors and use cases These concepts are simply an aid to defining what exists outside the system (actors) and what should be performed by the system (use cases) The actors represent what interacts with the system They represent everything that needs to exchange information with the system ' Typically actors are users and other computer systems or equipment Use cases represent a series of interactions between an instance of an actor and the system in order to accomplish something of value from a user perspective Mock ups of a user-interface are used where appropriate to facilitate the description of interactions between an actor and the system Use cases should provide a basis for testing independent of the design and implementation

Use cases are an initial partitioning and requirements identification strategy, they are first defined independently without concern to overlapping functions with other use cases Where appropriate, these may be integrated on a per-actor basis following the individual descriptions

4 1 1 Use Case Model Notation

Figure 2 Use Case Model Notation



Actors represent the things that interact with a system They represent everything that needs to exchange information with the system

Use cases are a strategy for partitioning the requirements of a system according to a user view, according to the major input events to the system from a user perspective

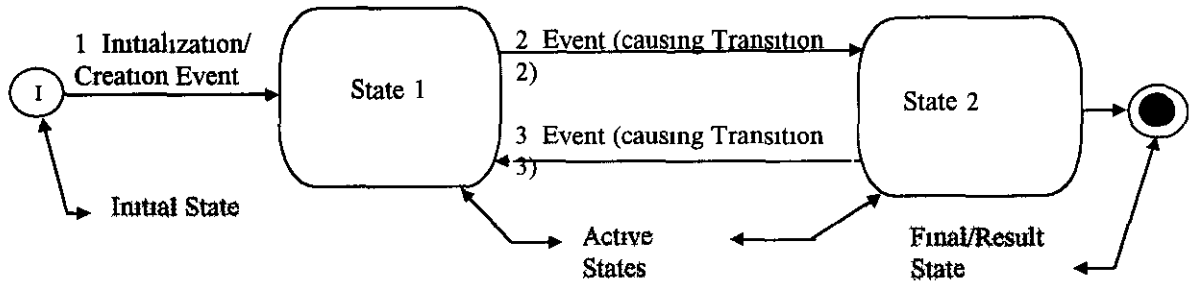
4 1 2 State Transition Diagram Notation[7]

As the TMACS system is a real-time system, a great deal of the required behavior can be clearly captured and communicated in state machine models The main value of a state transition diagram is that it compactly shows all the possible behaviors for an object Events that are not listed as transitions mean that they are effectively ignored

This is intended as a brief introduction to the ideas and notation employed in the state transition diagrams in this document A state is a static condition of an object during which the object is receptive to new events When an event occurs, the object responds by changing its state to reflect the new history of the object The new state depends on the event that occurred and on the previous state This transfer from one state to another is called a state transition The event that causes a state transition is called a triggering event or, simply a trigger When a transition is triggered by an event, the object may perform some action

State transition diagrams are simply a graphic representation of the state machine The conventions for the diagrams are shown below A state is represented by a round-tangle (rectangle with rounded corners) and a directed arc (line with an arrow) represents a transition from the source to the destination state The transitions are labeled with the event that triggers it The "before I'm created" initial point is indicated by a small circle with an "I" in it When a point is first created it will automatically take the initial transition associated from the initial point The final or terminal state (a double circle with a black center), if present represents the end or "object no longer exists" state - kind of a state of non-being

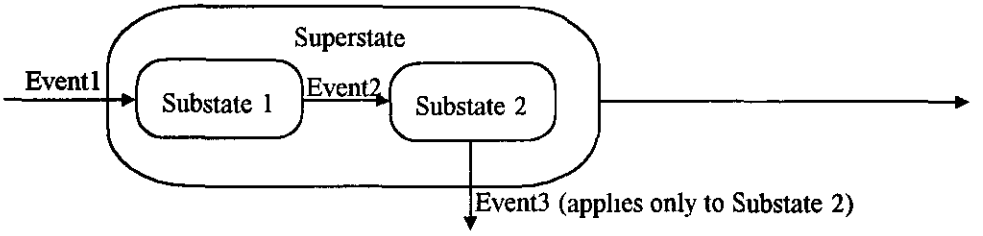
Figure 3 Dynamic Model (State Transition Diagram) Notation



A state may have other states defined within it For example, a siren may be in the ON state, within that state there are two others, steady and wavering The superstate and each of the other states may recognize a different set of events as shown in Figure 4

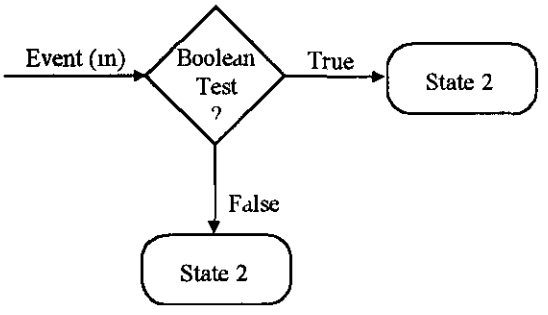
[7] The notation and format and description presented here are drawn in part from Real Time Object-Oriented Modeling, Brian Selic, Garth Gullekson, and Paul T Ward, John Wiley & Sons, 1994, pp 222-236 esp

Figure 4 State Generalization (Nesting) Notation



Sometimes an event can lead to two different states depending on the contents of the event or of other information available to the state This branching is shown in Figure 5

Figure 5 Decision Points (Branching) Notation



Note: an event going through a decision-point/branch will follow ONLY one branch out of the test as appropriate

The details of a transition are captured in text as follows

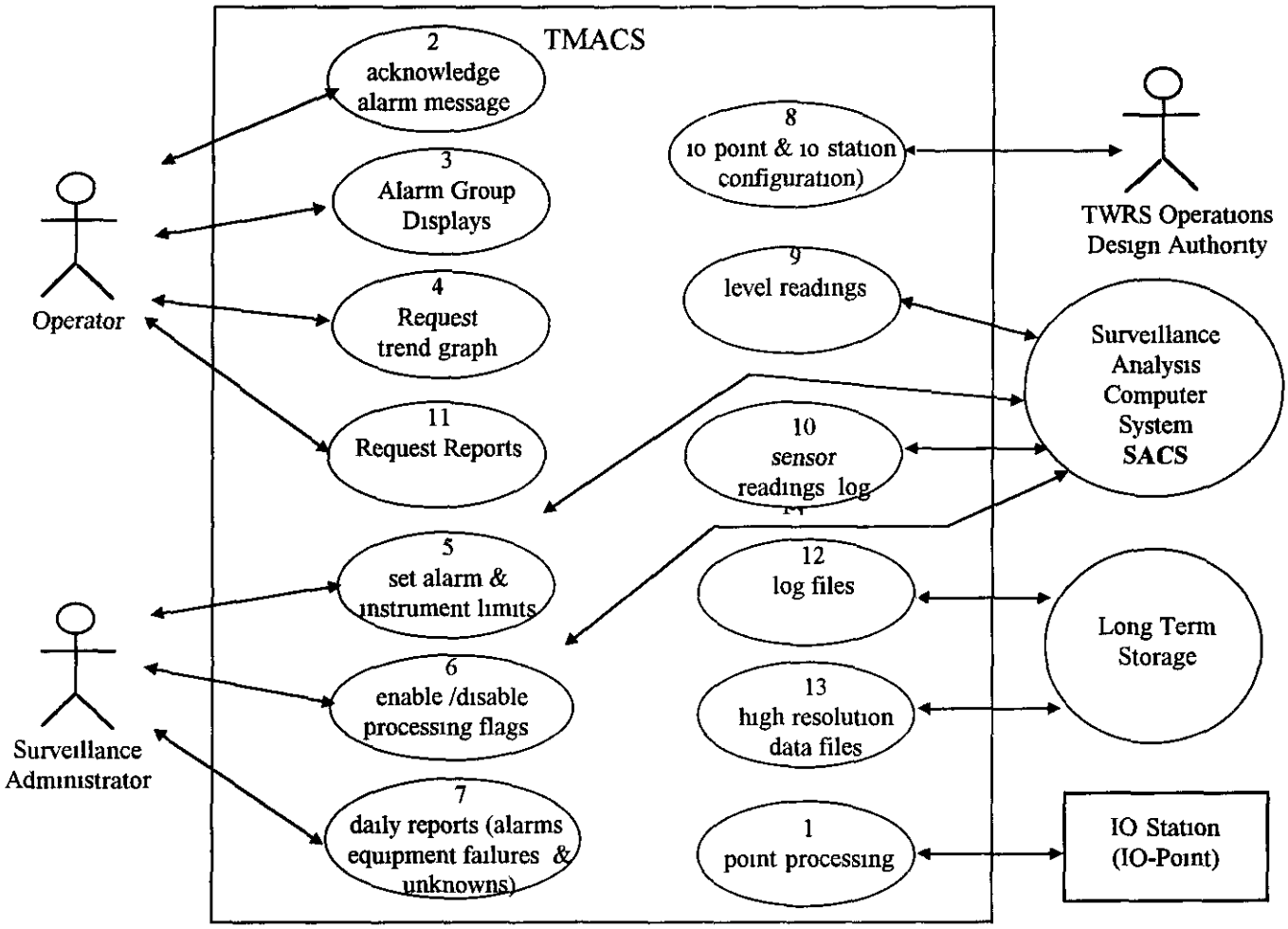
transition T
triggered by (name-of-triggering-signal-event, guard-condition
or name-of-triggering-signal-event guard-condition
or)
action (action-code/procedure)

where T = the name of the transition
name-of-triggering-signal-event = the name of the triggering signal of the event that causes the transition
guard-condition = an optional Boolean expression that if present must evaluate to TRUE for the transition

As the name TMACS (Tank Monitor and Control System) implies, monitoring a defined set of instrument sensors is at the heart of the system There is an expectation that the system perform a set of functions (processing) on each sensor reading this set of functions is referred to as point processing The purpose of this segment is to introduce the basic concepts, definitions and functions employed in point processing These concepts, definitions and functions will be employed in the following Use Cases, which exhaustively partition the requirements by event-type (for real-time point processing) and by user-initiated functional areas respective to user domain roles

4 2 Use Case Model (System Overview)

Figure 6 TMACS Use Case Model



4 3 Io-Station Use Cases

4 3 1 Use Case 1 Point-Processing

4 3 1 1 Definitions - Value And State

The value of an io-point/sensor is simply the actual value of the sensor in some human understandable units, e g , 10 degrees Fahrenheit for a temperature sensor or 100 inches for a level sensor The timestamp (actual collection date and time) of the reading is always kept with each reading

The state of a point/reading can be broken down out into two distinct concepts or sub-states

1 Quality Status[8] - The representation of the "goodness" or validity of a particular point s value, defined as follows

Quality Status Name	Meaning	Example(s)
Good	Communicating with sensor and reading is within expected instrument limits	Normal operation
Unknown	Don't have a current good reading	Lost communication (unable to get reading) Reading is out of instrument limits Just initialized, no readings yet
Test	Test/Calibration mode	Calibration of sensor in progress

2 Alarm State - used to indicate the current status of an alarm (a predefined condition that signals a warning or an alert), valid alarm states include

Alarm State	Meaning	Example
Normal	Able to communicate with sensor, not in alarm (within alarm limits)	Normal operation
Alarm	Able to communicate with sensor in alarm (exceeds alarm limits)	Temperature of 110 degrees, high alarm limit of 109
Unknown	Unable to communicate with sensor or just initialized	Power failure, broken modem etc
Disable[9]	Alarm processing turned off by an operator (user)	Sensor disabled for repair

In addition to the alarm-state, there is information, which gives the meaning of the alarm For continuous sensors, the meaning is done by alarm-type, for discrete sensors by an annotation

Continuous sensors can have associated alarm limits which, when exceeded, define an alarm-type - LOW-ALARM or HIGH-ALARM Only low and high limits have been implemented to date The normal case is called IN-RANGE In addition, a limit of how much a value can change per unit time, a rate-of-change limit, can be

[8] For unknown historical reasons, Quality State has always been referred to in the singular - i e Quality Status, whereas the Alarm State has been referred to as Alarm State We'll maintain the inconsistency so as not to "confuse the innocent"

[9] Note to the programmer Disable is not implemented via the alarm-state field it is implemented via a Boolean, alarm-processing-enabled

applied to a continuous sensor When this limit is exceeded it defines an alarm-type of ROC-ALARM The normal case is called ROC-IN-RANGE When everything is normal (i e IN-RANGE and ROC-IN-RANGE) the alarm-type is called NO-ALARM See the table in the definitions section

Discrete sensors are a bit simpler Each potential value of the sensor has a predefined alarm-state (NORMAL or ALARM) and annotation giving the meaning e g the definition for a discrete sensor of a valve could be something like

Discrete Value	Alarm State	Annotation	Display Color
0	Normal	Open	Green
1	Normal	Closed	Green
2	Alarm	Opening	Red
3	Alarm	Closing	Yellow

4 3 1 2 Overview

Point-processing is everything the system does in response to the receipt of a sensor reading

A high-level overview of the point-processing Use Case is as follows

WHEN an io-point sensor reading is received
(1) IF the value exceeds the specified instrument limits (is an invalid reading) THEN
Process Unknown, which is composed of
skip alarm check
log Unknown reading
make display reflect quality-status = Unknown
ELSE (reading is valid/within instrument limits)
IF value is changed or needs processing[10] THEN
(2) Process Changed Point which is composed of
Check Alarms and do Related Alarm Processing and Annunciation
Log Point
Display Point (value and quality status)
ENDIF
ENDIF
END WHEN

The above overview can be conceptually divided into two sections

1 Receive the Value and Check for Change and Validity (the main logic structure plus handling invalid points)

[10] There are three basic situations that require processing The first is a simple change in value such that the new reading exceeds the Delta The second is when the quality-status = Unknown and therefore any reading is a change The third is when the new reading is Within-Delta AND (BUT) the alarm-state = ROC-Alarm Normally, Within-Delta readings are not processed, but if the point is in a rate-of-change alarm, then the fact that it didn't change is a change, so a valid reading must be processed in this case

2 Process Changed Point (the required processing for changed readings)

The details of these will be broken down and given in the following subsections
An alternate overview of point processing is

IF (one of the following)

1	Delta exceeded	(value-based)
2	Quality-status unknown	(quality-status based)
3	ROC not-in-range	(alarm-based state)
4	First polling of new day	(time-based)
5	Operator Requested	(operator-initiated)

THEN

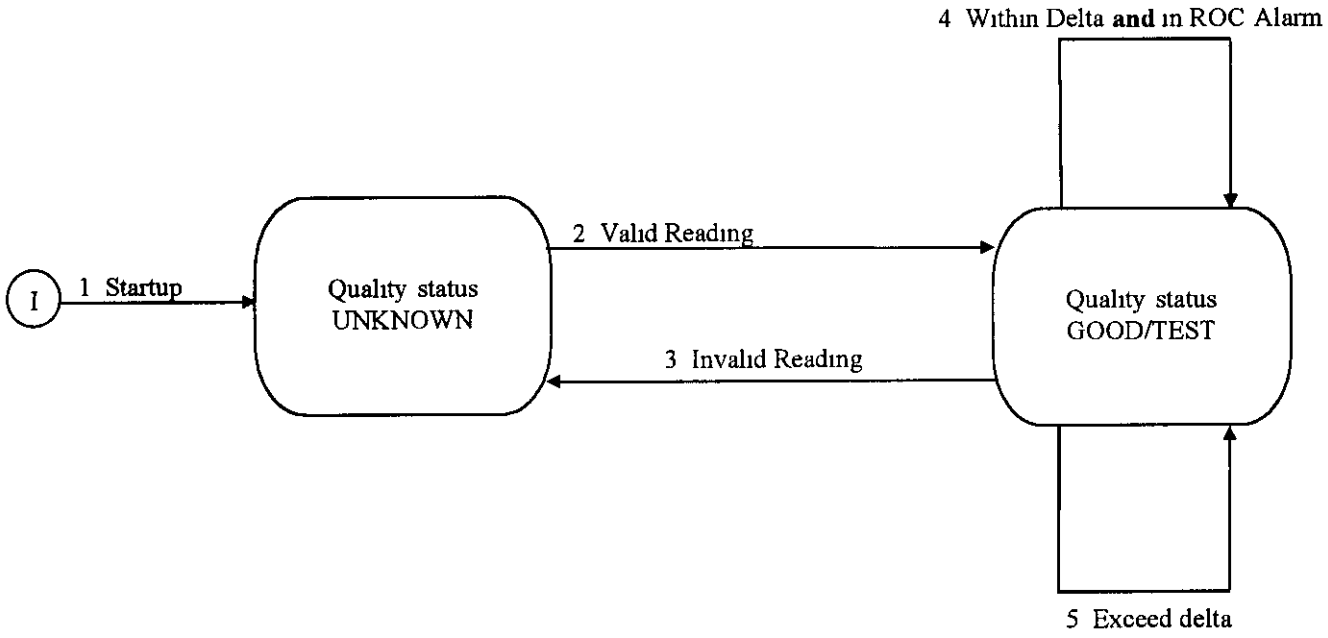
Check Instrument Limits
IF in-range
THEN
Process Changed Point
ELSE
Process Unknown

4 3 1 3 Receive Value and Check for Change and Validity

Per the pseudocode in the above description, 'receive value and check for change or validity" determines if the reading is valid (for continuous points this is within the instrument limits, for discrete points it means it's a defined value) If not, it processes this condition If it is valid, then it checks to see if it is considered changed Changed points are passed to the next logical chunk process-changed-points

Point Processing does an initial "screening' on the readings to see if there is a change or not (recall that any change of value, quality-status or alarm-state must be handled) These tests are embedded in the guard-conditions of the transitions in the diagram below

Figure 7 Quality-Status State Transition Diagram



The following table gives an alternate view of the above STD The initial states are noted on the left At the top are the possible combinations of Inside/Outside of Instrument Limits (i e , valid or invalid) and Exceeds-Delta/Within-Delta The inner cells show the state that the quality-status will transition to and give a reference number of the transition back to the STD

Table 1 Receive Value and Check for Change and Validity

	Reading Exceeds Delta-Check (change in value)		Reading Within Delta (no change in value)	
Initial quality-status	Reading Invalid	Reading Valid	Reading Invalid	Reading Valid
GOOD/TEST	UNKNOWN Transition (3) invalid reading	GOOD/TEST Transition (5) valid reading	NOT POSSIBLE	no-change, EXCEPT* if in ROC-Alarm Transition (4)
UNKNOWN	no-change	GOOD/TEST Transition (2) valid reading	no-change	GOOD/TEST Transition (2) valid reading

The remaining parts of the point processing use-case all start with a specific output of "Receive Value and Check for Change and Validity," which "Receive a changed and valid reading " Other situations (invalid or unchanged) are already covered in this portion of the point-processing use case

The following table gives a matrix view of the Quality-Status STD It gives a mapping of the initial quality-status (before point processing, shown on the left) to the resultant quality-status (after point processing, shown

across the top) The matrix is filled either with the number of the transition(s) from the STD or an informational note

Table 2 Quality-Status State Transition Matrix

Initial (quality-status)	Resultant quality-status GOOD/TEST	Resultant quality-status UNKNOWN
GOOD/TEST	Transitions 4 & 5	Transition 3
UNKNOWN	Transition 2	no-action (exceed instrument limits)

1 Transition Startup
Triggered by initialization
Action {set quality-status unknown }

Note there are actions done at startup, but these are outside the scope of point-processing At this point, it is important to note only that all points are initialized to quality-status of UNKNOWN at system startup The details will be described in the Initialization section of this document

2 Transition Valid Reading
Triggered by {Get-reading, IF value is within the specified instrument limits for continuous or defined for discrete}
Action {Process-changed-point}

NOTES 1 In this case where initial quality-status = UNKNOWN, the delta-check is irrelevant, because ANY valid value is a change

3 Transition Invalid Reading
Triggered by {Get-reading, IF value exceeds the specified instrument limits for continuous or is undefined for discrete }
Action {Process-Unknown skips alarm check, log UNKNOWN reading make display reflect quality-status = UNKNOWN}

NOTE This transition exists only from GOOD/TEST to UNKNOWN and does NOT exist from UNKNOWN, it is important to note that exceeding the instrument limits while already in the UNKNOWN state has no effect as it is not a change

4 Transition Within delta and in ROC-alarm
Triggered by {Get-reading, Within delta of last good reading AND alarm-state includes ROC-alarm}
Action {Process-changed-point}

NOTES 1 This is an exception to the rule for point s with quality-status = GOOD/TEST being within the delta check limits All other points are discarded since there is no change However, if we're in a rate-of-change alarm state, then the fact that we didn't change is a change (i e no longer experiencing a rate-of-change), so we will process a valid point

2 By definition, an unchanged GOOD/TEST point cannot exceed the instrument limits it is implicit here that it also passed the instrument-limit-check

5 Transition Exceeds delta
Triggered by {Get-reading, IF value > last-good-reading +/- delta}
Action {Process-changed-point}

4 3 1 3 1 Details of Supporting Functions for Receive Value and Check for Change and Validity

4 3 1 3 1 1 Get-Reading Get a Sensor Reading

Collect a sensor reading in realtime from its respective I/O processor A reading can be taken up to once per second per sensor A maximum of 10,000 points is anticipated The raw reading shall be converted from its raw device units to the appropriate engineering units as specified Readings will always be assumed to be in the appropriate human understandable units Operator can request individual sensor reading

4 3 1 3 1 2 Delta-Check Check to see if the Delta has been exceeded

A continuous sensor's value remains the same UNTIL it receives a new value To qualify as a "new value in a reading, the last-good-reading must change by some pre-defined delta, e g a temperature must change by more than 1 degree from the stored value If this delta band (last-good-reading +/- delta) has not been exceeded, then the value is considered not to have changed This is a key concept There are two exceptions where the delta-check is not performed these are the first reading taken on a new day and when an operator request a reading from a device In both of these instances the readings are acted upon as if they had exceeded the delta

For discrete points the delta-check is just a check to see if the value has changed

4 3 1 3 1 3 Validity-Check Check that the value reported in the reading is a valid value

For a reading to qualify as valid, for continuous points it must be within the specified range or set of values that the respective sensor can produce A reading outside these instrument limits is considered an error

For continuous points, Validity-Check is conceptually Check-Instrument-Limits, which simply compares the reading against the instrument limits and returns an indication of whether or not the reading is in range

For discrete io-points, validity is being within the set of predefined valid values that a discrete sensor can have

4 3 1 4 Process Changed Point

The outline for Process Changed Point is as follows

```
IF alarm processing is enabled THEN
    DO alarm processing
        IF reading is from a continuous point THEN
            DO limit checks
                Check against low and high alarm limits, annunciate as appropriate
            END (DO limit checks)
            IF ROC-alarm-processing is enabled THEN
                Check rate of change against ROC alarm limit and process as appropriate
            ENDIF
        ELSE (reading is a discrete point)
```

Check alarm state of value per configuration of discrete point, annunciate as appropriate
ENDIF
END (Do alarm processing)
ENDIF (alarm processing is enabled)
IF logging is enabled THEN
DO Log Point
write the sensor name of the point, value unit (continuous), timestamp(s), quality status, alarm state (discrete) or alarm-type (continuous), and alarm meaning to disk
END (Log Point)
ENDIF (logging is enabled)
DO Display Point
Display the value and quality status for the reading
END (DO Display Point)

The three basic parts of Process Changed Point i e

1 Alarm Processing Check Alarms and do Related Alarm Processing and Annunciation
2 Log Point
3 Display Point (value and quality status)

will be detailed in the following sections

4 3 1 4 1 Alarm Processing

4 3 1 4 1 1 Check-Alarms Check for Alarms

See the Value and State section for an introduction to and explanation of setting alarm-state and alarm-type

4 3 1 4 1 1 1 Continuous io-points

Alarm checking for continuous io-points has two distinct components, range-limit checking and rate-of-change limit checking When there are no alarms of either kind (i e the status for all alarms is IN-RANGE), then the alarm state is NORMAL and the alarm-type is NO-ALARM When there is an alarm the alarm state is ALARM and the alarm-type is the composite of the range-alarm status [LOW-ALARM or HIGH-ALARM] plus the ROC-alarm-status [ROC-ALARM] The following shows the composite alarm-type that is derived from combining the range-alarm-status (shown on left) with the ROC-alarm-status (shown across the top), the entries in the matrix are the respective composite continuous alarm-type Note that the alarm-state is not shown but is ALARM for any of the alarms and NORMAL where both are IN-RANGE

Note also the strange implementation detail of switching the order of the words between the range-alarm-status and the alarm-type -e g a range-alarm-status of LOW-ALARM results in an alarm-type of ALARM-LOW Also, the ROC alarm-types are shown using the plus (+) and minus (-) signs appended to ROC - e g ROC-alarm-status of ROC-HIGH-ALARM yields an alarm type of ROC+

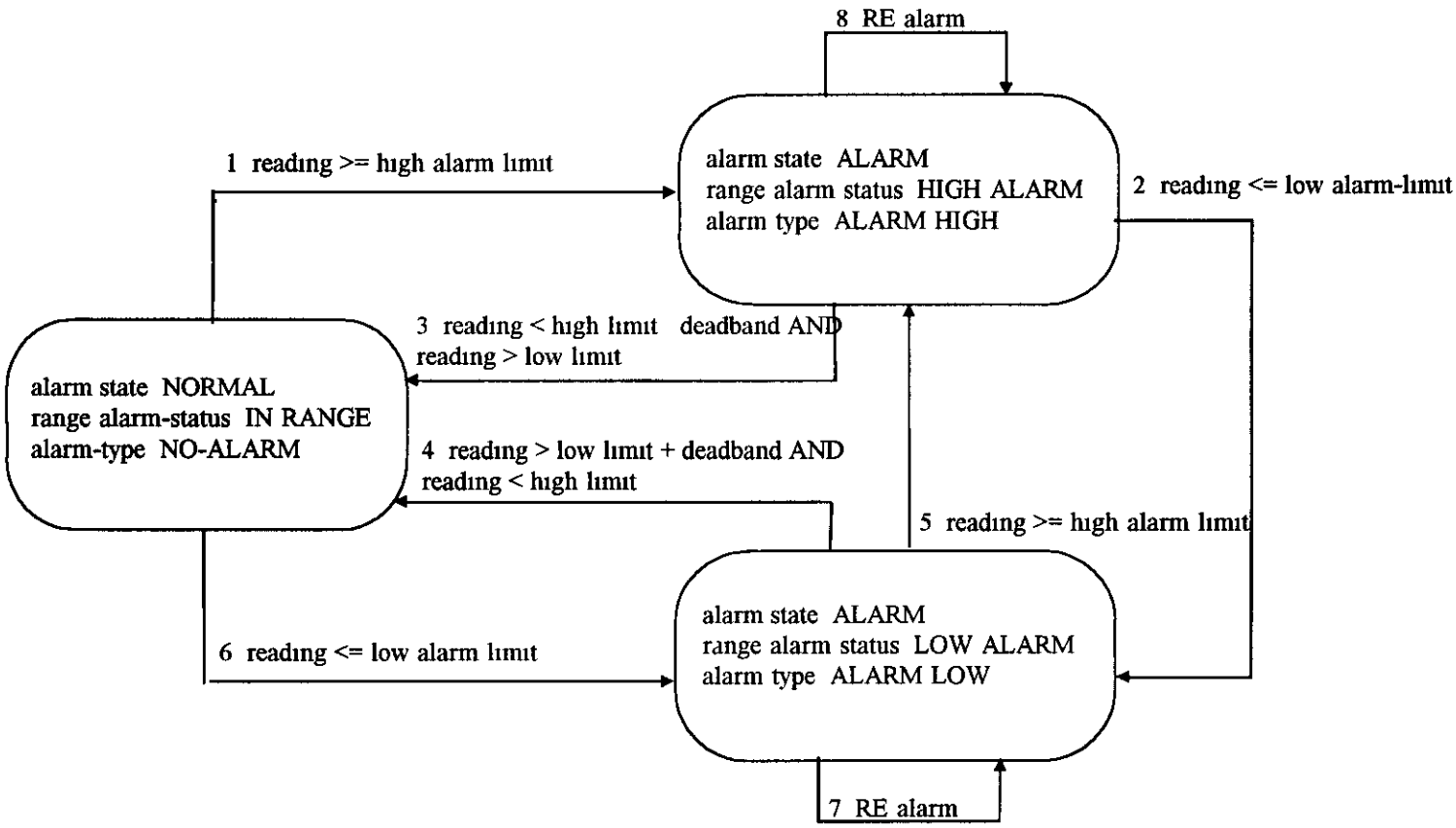
Table 3 Composite Alarm-Type for Continuous IO-Points

ROC-Alarm-Status (right) Range-Alarm-Status (below)	ROC-LOW-ALARM	ROC-HIGH-ALARM	ROC-IN-RANGE
LOW-ALARM	ALARM-LOW/ROC-	ALARM-LOW/ROC+	ALARM-LOW
HIGH-ALARM	ALARM-HIGH/ROC-	ALARM-HIGH/ROC+	ALARM-HIGH
IN-RANGE	ROC-	ROC+	NO-ALARM

4 3 1 4 1 1 1 1 Range Limit Checking for Continuous IO-Points

Range limit checking involves the simple checking of the current reading against pre-defined low and high alarm limits There is also a requirement is to avoid alarm chatter - the situation where a reading cycles just above and below a limit causing multiple "nuisance" alarms Deadband filtering is a mechanism employed to eliminate alarm chatter Once a point goes into an alarm, it will not be reset (considered back to normal and out of alarm) until it has exceeded the-limit +/- the-deadband-value The deadband value is subtracted from a High-Alarm limit and added to a Low-Alarm limit

Figure 8 Range Alarm State Transition Diagram for Continuous IO Points



The following table provides a matrix view of the above check alarm function for range-limit alarms The initial state combination is given on the left, the triggering events are shown across the top and the resultant alarm-state range-alarm-status is the matrix entry

Table 4 State/Event Matrix for Range Alarm

Event (to right) Initial alarm-state Range-alarm-status	reading <= low-alarm-limit	reading >= high-alarm-limit	reading > low-alarm-limit & reading < high-alarm-limit
UNKNOWN N/A	ALARM LOW-ALARM	ALARM HIGH-ALARM	NORMAL IN-RANGE
NORMAL IN-RANGE	ALARM LOW-ALARM	ALARM HIGH-ALARM	NORMAL IN-RANGE

Event (to right) Initial Alarm-state Range-alarm-status	reading > low-alarm-limit + deadband AND reading < high-limit	reading < high-alarm-limit - deadband AND reading > low-limit	(1) reading < low-alarm-limit + deadband OR (2) reading > high-alarm-limit - deadband
ALARM LOW-ALARM	NORMAL IN-RANGE	N/A	(1) no change
ALARM HIGH-ALARM	N/A	NORMAL IN-RANGE	(2) no change

The following table gives a matrix view of the Quality-Status STD It gives a mapping of the initial alarm-state/type (before alarm processing, shown on the left) to the resultant alarm-state/type (after alarm processing, shown across the top) The matrix is filled either with the number of the transition(s) from the STD or an informational note The UNKNOWN alarm-state was not shown in the STD, but the actions for the transitions parallel those from the NORMAL IN-RANGE combination as indicated below

Table 5 State Matrix for Range Alarm Status

(initial) alarm-state range-alarm status	UNKNOWN N/A	NORMAL IN-RANGE	ALARM LOW-ALARM	ALARM HIGH-ALARM
UNKNOWN N/A		within limits - no tran	same as 6	same as 1
NORMAL IN-RANGE	not-done	within limits - no tran	6	1
ALARM LOW-ALARM	not-done	4	7	5
ALARM HIGH-ALARM	not-done	3	2	8

Note that the customer has only asked for a single low and a single high alarm to date Conceptually, there could be multiple low and high alarm limits, each with a specific name and limit Functionally these would work

similarly to above, by adding the specific name of the low or high alarm limit A severity precedence would also need to be indicated such that the "lowest low alarm" or "highest high alarm" would be indicated at the point that more than one low alarm limit or more than one high alarm limit had been exceeded This is not a current requirement and has not been implemented in the current design

1 Transition reading >= high-alarm-limit
Triggered by {Get-valid-reading, IF value >= high-alarm-limit }
Action {Annunciate alarm (create new alarm message)}

2 Transition reading <= low-alarm-limit
Triggered by { Get-valid-reading, IF value <= low-alarm-limit }
Action {Annunciate alarm (create new alarm message)}

NOTE This transition is extremely unlikely, but still should be accounted for

3 Transition reading < high-alarm-limit + deadband AND reading > low-limit
Triggered by {Get-valid-reading, IF value < high-alarm-limit + deadband AND value < low-limit}
Action {Annunciate alarm (indicate reset)}

NOTE Once in the alarm state, new in-alarm values that don t exceed the deadband tolerance stay in the alarm state and don't normally cause any processing with the alarm-processing function

4 Transition reading > low-alarm-limit + deadband AND reading < high-limit
Triggered by {Get-valid-reading, IF value < high-alarm-limit + deadband AND value < low-limit}
Action {Annunciate alarm (indicate reset)}

NOTE Once in the alarm state, new in-alarm values that don't exceed the deadband tolerance stay in the alarm state and don't normally cause any processing with the alarm-processing function

5 Transition reading >= high-alarm-limit
Triggered by {Get-valid-reading IF value >= high-alarm-limit}
Action {Annunciate alarm (create new alarm message)}

NOTE This transition is extremely unlikely, but still should be accounted for

6 Transition reading <= low-alarm-limit
Triggered by {Get-valid-reading, IF value <= low-alarm-limit}
Action {Annunciate alarm (create new alarm message)}

7 & 8 Transition RE-alarm
Triggered by {Get-valid-reading, IF reading exceeds respective alarm-limit AND Quality-Status cycled }
Action {Annunciate alarm (create new alarm message)}

NOTE This ONLY happens when the sensor is already in the alarm-state AND the quality-status cycles from GOOD/TEST to Unknown and back to Good IF the quality status stays GOOD/TEST, then there is no re-alarm If it stays Unknown then a new reading can't be in alarm

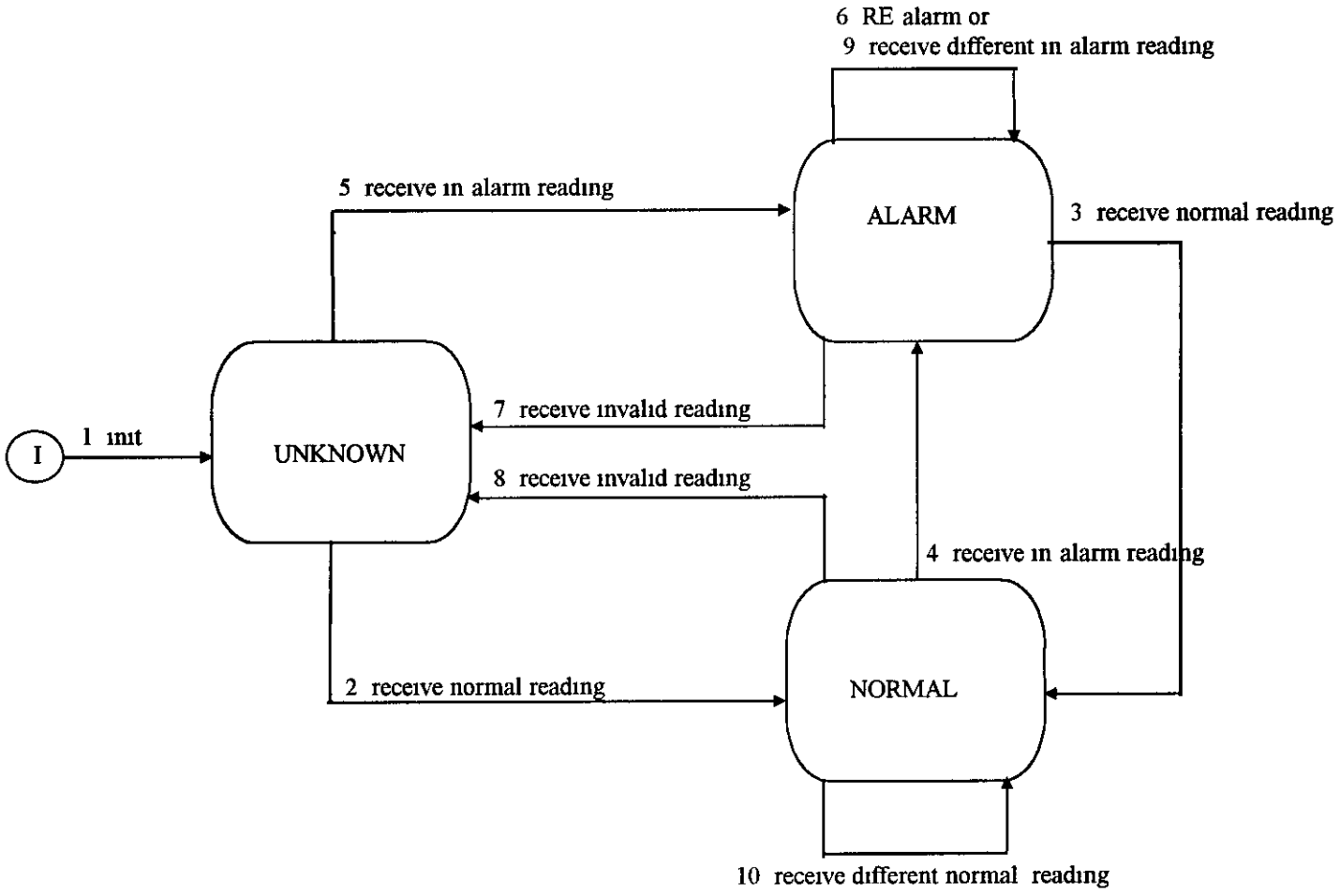
4 3 1 4 1 1 1 2 Rate of Change (ROC) Alarm Limit Checking for Continuous IO-Points

The ROC alarm limit check is done by comparing the current reading against historical reading(s) The check is to see if the rate of change (amount of change over time) exceeds the specified limit The limit is specified as a absolute number of units of change (same unit the value is normally measured in) per some number of time-interval-units - e g , 4 degrees per hour If the value of the reading drops more than that amount per the specified time interval, this initiates an ROC-LOW-ALARM If the value of the reading increases more than this amount per the specified time interval this initiates a ROC-HIGH-ALARM Thus the values for the roc-alarm-status can be [IN-RANGE, ROC-LOW-ALARM or ROC-HIGH-ALARM] If the system is not recording a history of the readings, or if the time period of the recorded history is less than the time-interval specified for the roc-alarm, then the roc-alarm-status will be ROC-UNKNOWN If the rate-of-change is specified as some number of units per 24 hours, then until the system has been up for 24 hours, the roc-alarm-status will be ROC-UNKNOWN

4 3 1 4 1 1 2 Alarm Checking for Discrete io-points

Per the pre-defined configuration for each valid discrete io-point value, the alarm state is either ALARM or NORMAL Note that the delta-check for discrete sensor readings will prevent the same reading from registering more than one alarm for a contiguous series of identical alarm-state readings

Figure 9 Discrete Alarm-State State Transition Diagram



The following table gives the corresponding transition that causes the initial alarm state (shown on the left) to be changed to the resultant alarm-state (shown across the top)

Table 6 State Matrix for Discrete Alarm State

Resultant alarm-state to right Initial alarm-state below	UNKNOWN	NORMAL	ALARM
UNKNOWN		2	5
NORMAL	8	10 (or none if same value)	4
ALARM	7	3	6 and 9

1 Transition Initialization
Triggered by {Initialization/Startup}
Action {Set Alarm State = Unknown)}

2 Transition receive normal reading (from unknown)
Triggered by {Get-valid-reading IF configuration for received value's alarm state = Normal}
Action {Set alarm-state to NORMAL and log That s it }

3 Transition receive normal reading (from alarm)
Triggered by {Get-valid-reading, IF configuration for received value's alarm state = Alarm}
Action {Annunciate alarm (indicate reset) and log}

4 & 5 Transition receive in-alarm-reading
Triggered by {Get-valid-reading, IF configuration for received value's alarm state = Alarm }
Action {Annunciate alarm (create new alarm message) and log}

6 Transition RE-alarm
Triggered by {Get-valid-reading IF configuration for received value s alarm state = Alarm AND Quality-Status}
Action {Annunciate alarm (create new alarm message) and log}

NOTE This ONLY happens when the sensor is already in the alarm-state AND the quality-status cycles from GOOD/TEST to UNKNOWN and back to Good IF the quality status stays GOOD/TEST, then there is no re-alarm If it stays UNKNOWN, then a new reading can t be in alarm

A "re-alarm" is when the quality-status of an in-alarm point goes from GOOD/TEST to UNKNOWN and back to GOOD/TEST, but the in-alarm reading didn't change In this case, the system shall generate a new alarm message when a sensor re-alarms due to communication or instrument failure

7 & 8 Transition Receive Invalid Reading
Triggered by {Get-invalid-reading}
Action {Set alarm-state to UNKNOWN rest already handled by process-unknown function in first part of point processing}

NOTE For continuous alarms the alarm-state is ONLY unknown upon initialization UNTIL a valid reading is received, after that UNKNOWN conditions are handled by the quality-status and the alarm-state (NORMAL or ALARM) is retained This should probably be the same for discretes, but is not currently For discrete sensor readings, any invalid reading or communication-loss sets the quality status to UNKNOWN The memory of what the alarm-state and color was is done via another mechanism

9 Transition Receive Different in-alarm Reading
Triggered by {Get-valid-reading, IF configuration for received value's alarm state = Alarm AND value is different from last reading}
Action {Annunciate alarm (new alarm) and log}

NOTE This assumes a non binary-alarm that has more than one value that describes a different type of alarm For binary sensors (alarm or normal) it will never happen

10	Transition	Receive Different Normal Reading
	Triggered by	{Get-valid-reading, IF configuration for received value s alarm state=Normal AND value is different from last reading}
	Action	{Annunciate alarm (new alarm) and log}

NOTE This assumes a non binary-alarm that has more than one value that describes a different type of alarm For binary sensors (alarm or normal) it will never happen

4 3 1 4 1 2 Details of Supporting Functions for Alarm Processing

The basic support functions for alarm processing are annunciate-alarm and log-alarm These are each described in the following sections

4 3 1 4 1 2 1 Annunciate-Alarm

Annunciate-alarm has several components that include

- Annunciate the individual sensor alarm
- Annunciate the alarm group display
- Manage the current alarm display list

4 3 1 4 1 2 1 1 Annunciate Individual Sensor Alarm

Visual warning, an audible warning and an alarm message shall indicate each transition from a non-alarm to an alarm state Alarm messages shall indicate the sensor tag-list name, the timestamp of the event (reading), the respective meaning (annotation for discrete sensor readings, alarm-type for continuous points) and the respective display color

Each reset (transition from alarm back to normal) shall be indicated in the original alarm message and logged A visible indication of an unacknowledged alarm shall be given on its primary display and shall persist until that alarm has been acknowledged, even if the alarm state goes back to normal before the alarm is acknowledged An alarm that has been reset and acknowledged shall be removed from display The most recent alarm shall be displayed as such by the system

4 3 1 4 1 2 1 2 Handling of Current-Alarm-List

Each alarm message shall be added to the current-alarm-list (i e the list of all current alarms in the system) This list shall contain only one instance of an alarm message for each sensor Alarm messages are to be removed from the current-alarm-list only when the alarm has been reset (value returned to normal) and the alarm message has been acknowledged The only exception to this is for ' re-alarms, i e , in the case of a "re-alarm, the old alarm message shall be removed Requirements and functions related to acknowledgment shall be covered more fully under the Operator Use Cases

4 3 1 4 1 2 1 3 Annunciation of Respective Alarm-Group

Any change in alarm-state or alarm-type will cause the system to update the status of the respective alarm-group so that it reflects the highest priority (i e , the most severe) state

The behavior of annunciating the alarm-group should be as follows

IF all sensors in the alarm group are Disabled THEN
alarm group display color = gray

ELSE IF all of the alarm-states of the sensors in the alarm group are No-Alarm THEN
IF all of the sensors are quality-status=Good
alarm group display color = green
ELSE (some sensors in alarm group are quality-status=Unknown)
alarm group display color = white
ELSE (some alarm-states are Not No-Alarm - some alarms) THEN
IF any alarm-colors are Red THEN
alarm group display color = Read
ELSE (none are Red) IF any alarm-colors are Yellow THEN
alarm group display color = Yellow
ELSE
at present only Red and Yellow are defined for alarms
Lower Priority colors would be handled here
ENDIF
ENDIF
IF any alarms are Unacknowledged THEN
alarm group display color should be blinking
ENDIF

4 3 1 4 1 3 Log-Alarm

Every alarm shall be logged to disk The log will contain all the information in the alarm message (see annunciate-alarm), excepting the display color

4 3 1 4 2 Log-Reading

A change of value, quality-status or alarm-state (discrete) or alarm-type (continuous) of a point shall generate a log to disk if logging is enabled for the point This log entry shall also be printed on the alarm printer A log entry shall include the sensor tag-list-name of the point, the value the associated units (continuous) the associated timestamp, the quality status, the alarm state (discrete) or the alarm-type (continuous), and the meaning of the alarm (the annotation or alarm-type)

At a minimum, the following items will be logged for continuous io-points

- Tag name
- Value
- Units
- Collection time (timestamp)
- Quality-status
- Alarm-type

The following items will be logged for discrete io-points

- Tag name
- Value
- Annotation
- Collection-time
- Quality-status
- Alarm-state

- Descriptor

4 3 1 4 3 Display-Reading

The individual status for each sensor shall be displayed on the primary display The states that must be discernible are

Quality-status = GOOD/TEST, not in alarm
Quality-status = GOOD/TEST, in alarm (not acknowledged)
Quality-status = GOOD/TEST, in alarm (acknowledged)
Quality-status = UNKNOWN (see note below)

IF the sensor was in alarm when it's quality-status went UNKNOWN, THEN the alarm-state-display-color at that time shall also be indicated

The value, as converted to user-defined and understandable units, will be displayed for all continuous points

The display of a discrete sensor shall include its annotation or label, and its respective display color as configured The value for a discrete sensor is not required to be displayed

4 3 1 4 3 1 Display Color

The sensor display color for the io-point shall operate as follows

IF the sensor is Disabled (i e point-processing is disabled) THEN
    io-point display color = gray
ELSE IF the quality-status=Good THEN
    IF the alarm state = No-Alarm THEN
        io-point display color = green
    ELSE (alarm-state not No-Alarm - i e some alarm(s)) THEN
        io-point display color = Alarm color (red or yellow)
ENDIF
ELSE (quality-status=UNKNOWN) THEN
    IF alarm-state was No-Alarm When quality-status went Unknown THEN
        io-point display color = White
    ELSE (was in alarm when quality-status went Unknown) THEN
        io-point display color = White
        PLUS additional indication (e g surrounding color) of alarm color
    ENDIF
ENDIF

4 3 1 5 Alternate Paths Communication Errors

4 3 1 5 1 Lose Communication with Sensor

When communication is lost with a sensor the system shall set the quality status for the point to UNKNOWN log the event to disk, and make and the display reflect the UNKNOWN sensor status

Restoration of communication with a sensor is handled as a normal part of point-processing

4 3 1 5 2 Lose Communication with I/O Station

When communication is lost with an io-station, the system should notify the operator with an alarm message of the equipment failure, giving the time of the failure and the identity (name) of the io-station Then for each sensor connected to the io-station, the system shall initiate a "lose communication with io-point" event, which will have the actions as specified under that use case

4 3 1 5 3 Communication Restored with I/O Station

When communication is restored with an io-station, the original alarm message shall be modified to indicate that it has been restored If the alarm has been acknowledged, it will be removed from the display

No specific action is initiated relative to the respective sensors attached to the io-station They will be handled by the normal point-processing functions when their new readings are next received

4 3 1 6 Example Instances of Point Processing Use Cases for Continuous Sensors

4 3 1 6 1 First Reading for a Sensor Valid Non-Alarm Value

This example describes the handling of an initial reading for a sensor, where the reading is valid and is not in alarm

Pre and Post Conditions of Point Processing values for this io-point

Variable	Value-Before	Value-After
quality-status	UNKNOWN	GOOD
alarm-state	UNKNOWN	NORMAL
alarm-type	NO-ALARM	NO-ALARM
sensor display color	WHITE	GREEN

Aggregate values for all io-points in alarm group

Variable	Value-Before	Value-After
number with quality-status=Unknown	1	0
number with alarm-state=Alarm	0	0
number with alarm display color = Red	0	0
number with alarm display color = Yellow	0	0
alarm-group display color	WHITE	GREEN

ACTIONS

The system has initialized the quality-status and alarm-state for the sensor both to UNKNOWN at startup

The system receives the reading from the sensor, initiating the point processing use case

The reading is within the instrument limits, so it is valid

Because the quality-status is UNKNOWN, the process-changed-point processing is initiated

Alarm Processing The system checks the reading against the alarm limits It passes both the range limit alarm checks and the ROC alarm check- i e , there is no alarm, so alarm-state goes to NORMAL and alarm-type to NO-ALARM

Log-Reading The system logs the point to disk with timestamp, value, units, sensor-tag, quality-status=GOOD, alarm-type=NO-ALARM

Display Reading The quality-status is GOOD and the alarm-state is NORMAL so sensor display is updated to green All sensors in group have quality-status GOOD and all have alarm-state of NORMAL, so alarm-group display is updated to GREEN

4 3 1 6 2 Reading Within Delta band Valid Non-Alarm Value

This example describes the actions during normal operation where the value of the reading received within the delta of the value of the last reading The new reading is valid and is not in alarm

Pre and Post Conditions of Point Processing values for this io-point

Variable	Before-Value	After-Value
quality-status	GOOD	GOOD
alarm-state	NORMAL	NORMAL
alarm-type	NO-ALARM	NO-ALARM
sensor display color	GREEN	GREEN

Aggregate values for all io-points in alarm group

Variable	Before-Value	After-Value
number with quality-status=Unknown	0	0
number with alarm-state=Alarm	0	0
number with alarm display color = Red	0	0
number with alarm display color = Yellow	0	0
alarm-group display color	GREEN	GREEN

ACTIONS

The system receives the reading from the sensor initiating the point processing use case

Delta-Check Compare this value with last-good-reading +/- delta It's within so do NOT call process changed point, that's it!

Note There are NO changes!

This is a key concept A sensor's value remains the same UNTIL it receives a new value (by changing more than delta), or it receives a bad value (e g exceeds instrument limits) or no value when polled (communication error)

4 3 1 6 3 Reading Exceeds Deltaband Valid Non-Alarm Value

This example describes the actions during normal operation where the value of the reading received exceeds the delta of the value of the last reading The new reading is valid and is not in alarm

Pre and Post Conditions of Point Processing values for this io-point

Variable	Before-Value	After-Value
quality-status	GOOD	GOOD
alarm-state	NORMAL	NORMAL
alarm-type	NO-ALARM	NO-ALARM
sensor display color	GREEN	GREEN

Aggregate values for all io-points in alarm group

Variable	Before-Value	After-Value
number with quality-status=Unknown	0	0
number with alarm-state=Alarm	0	0
number with alarm display color = Red	0	0
number with alarm display color = Yellow	0	0
alarm-group display color	GREEN	GREEN

ACTIONS

The system receives the reading from the sensor initiating the point processing use case

The reading is within the instrument limits, so it s valid

Because the value exceeds the delta check, the process-changed-point processing is initiated

Alarm Processing The system checks the reading against the alarm limits It passes both the range limit alarm checks and the ROC alarm check There is no alarm, so no changes

Log-Reading The system logs the point to disk with timestamp, value, units, sensor-tag, quality-status=GOOD alarm-type=NO-ALARM

Display Reading The new value of the reading is indicated in the display the quality-status is good and the alarm-state is NORMAL so sensor display remains green All sensors in group have quality-status GOOD and all have alarm-state of NORMAL so alarm-group display remains GREEN

Note that there are no changes to the listed variables, BUT the new value would be displayed and logged

4 3 1 6 4 Going into Alarm Exceeds High Alarm Limit

This example describes the actions during normal operation where the previous reading was valid but not in alarm, and the value of the current reading exceeds the high-alarm limit

Pre and Post Conditions of Point Processing values for this io-point

Variable	Before-Value	After-Value
quality-status	GOOD	GOOD
alarm-state	NORMAL	ALARM
alarm-type	NO-ALARM	HIGH-ALARM
sensor display color	GREEN	RED

Aggregate values for all io-points in alarm group

Variable	Before-Value	After-Value
number with quality-status=Unknown	0	0
number with alarm-state=Alarm	0	1
number with alarm display color = Red	0	1
number with alarm display color = Yellow	0	0
alarm-group display color	GREEN	RED

ACTIONS

The system receives the reading from the sensor, initiating the point processing use case

The reading is within the instrument limits, so it's valid

Because the value exceeds the delta check, the process-changed-point processing is initiated

Alarm Processing The system checks the reading against the alarm limits It exceeds the high-alarm-limit and passes the ROC alarm check The system creates an alarm message indicating it is unacknowledged, displays it as the most recent alarm, and adds it to the current-alarm-list The system logs the alarm message

Log-Reading The system logs the point to disk with timestamp value units sensor-tag, quality-status=GOOD, alarm-type=HIGH-ALARM

Display Reading The new value of the reading is indicated in the display, the quality-status is good and the alarm-state is ALARM so sensor display reflects the alarm color configured for HIGH-ALARM for this sensor All sensors in group have quality-status GOOD and there is only one sensor in alarm, so the alarm-group display changes to reflect the alarm color of this sensor

4 3 1 6 5 Coming out of Alarm/Return to Normal and Alarm-Group Annunciation

This example describes the actions during normal operation where the previous reading was valid and in alarm and the value of the current has returned to normal (exceeded the deadband) This situation also includes another sensor in the same alarm group which is also in alarm at a lower priority (yellow display), which also provides an example of how alarm-group annunciation works

Pre and Post Conditions of Point Processing values for this io-point

Variable	Before-Value	After-Value

quality-status	GOOD	GOOD
alarm-state	ALARM	NORMAL
alarm-type	HIGH-ALARM	NO-ALARM
sensor display color	RED	GREEN

Aggregate values for all io-points in alarm group

Variable	Before-Value	After-Value
number with quality-status=Unknown	0	0
number with alarm-state=Alarm	2	1
number with alarm display color = Red	1	0
number with alarm display color = Yellow	1	1
alarm-group display color	RED	YELLOW

ACTIONS

The system receives the reading from the sensor, initiating the point processing use case

The reading is within the instrument limits, so it s valid

Because the value exceeds the delta check the process-changed-point processing is initiated

Alarm Processing The system checks the reading against the alarm limits The value is less than the high-limit minus the deadband AND is greater than the low-alarm limit The system updates the original alarm message indicating it is reset If the alarm has been acknowledged it is removed from the displays The system logs the alarm message reset, including the time of reset and the duration of the alarm

Log-Reading The system logs the point to disk with timestamp value units sensor-tag, quality-status=GOOD and alarm-type=NO-ALARM

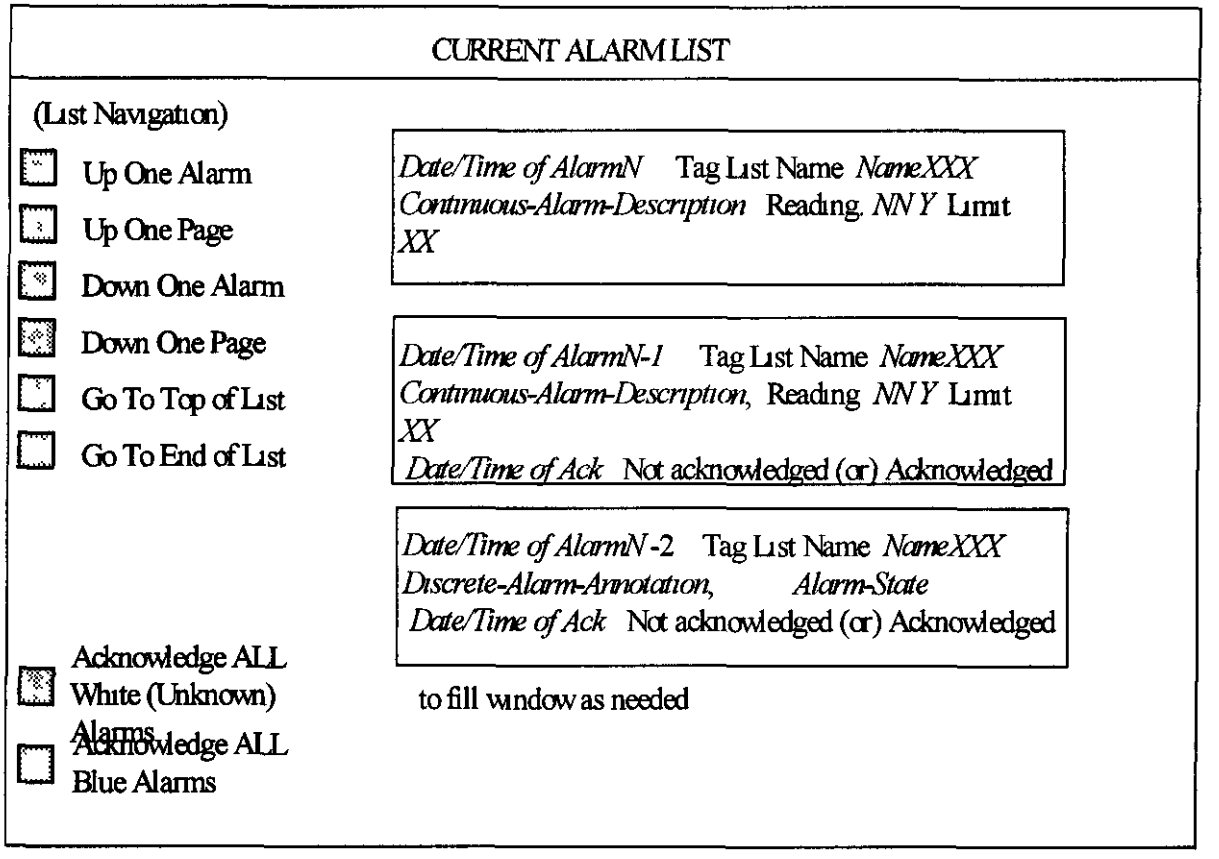
Display Reading The new value of the reading is indicated in the display, the quality-status is good and the alarm-state is GOOD so sensor display returns to GREEN All sensors in group have quality-status GOOD and there is only one sensor still in alarm so the alarm-group display changes to reflect the YELLOW alarm color of this sensor

4 4 Operator Use Cases

4 4 1 Use Case 2 Acknowledge Alarm Messages

The operator gives the command to view the current alarms

The system presents the following window to the operator


CURRENT ALARM LIST
(List Navigation)
Up One Alarm
Up One Page
Down One Alarm
Down One Page
Go To Top of List
Go To End of List
Acknowledge ALL White (Unknown) Alarms
Acknowledge ALL Blue Alarms
Date/Time of AlarmN Tag List Name NameXXX Continuous-Alarm-Description Reading NNY Limit XX
Date/Time of AlarmN-1 Tag List Name NameXXX Continuous-Alarm-Description, Reading NNY Limit XX
Date/Time of Ack Not acknowledged (or) Acknowledged
Date/Time of AlarmN-2 Tag List Name NameXXX Discrete-Alarm-Annotation, Alarm-State
Date/Time of Ack Not acknowledged (or) Acknowledged
to fill window as needed

The operator has the following choices to navigate through the current alarm list

- Up one alarm
- Up one page
- Down one alarm
- Down one page
- Go to top of list
- Go to bottom of list

The operator can acknowledge individual alarms by selecting the alarm message from the current alarm list to be acknowledged (i e , mouse click on the alarm message)

When an alarm is acknowledged, the system will perform the following actions

- Log the acknowledgment to disk
- Remove the alarm from the current-alarm-list IF it already reset (returned to normal)
- Indicate the acknowledgment visibly (e g , change display from "NOT Acknowledged to "Alarm Acknowledged" including the timestamp of when the acknowledgment occurred
- Indicate acknowledgment audibly (e g , stop beeping)

ALTERNATE COURSES

Acknowledge All White Alarms

The operator can acknowledge ALL outstanding White alarms by selecting "Acknowledge ALL White Alarms White alarms consist of situations causing the quality-status to go UNKNOWN which include instrument-limits-exceeded and communication/equipment failure

The system will then perform the acknowledgment functions listed above for each alarm caused by UNKNOWN quality-status/alarm-state (e g loss of communication)

Acknowledge All Blue Alarms

The operator can acknowledge ALL outstanding Blue alarms by selecting Acknlowledge ALL Blue Alarms' Blue alarms are caused when the following processes are enabled or disabled

- Point Processing
- ROC
- Alarm Processing
- Logging
- Test Mode

Acknowledge All Alarms for an Alarm Group

The operator selects a particular alarm group (i e tank or monitored system)
The system displays a detailed view of the respective alarm group

The operator then selects the ' Acknowledge All Alarms' function which is contained on the display

The system will then perform the acknowledgment functions listed above in the normal course for each alarm caused by UNKNOWN quality-status/alarm-state (e g loss of communication) Per the color requirements, this should be all WHITE alarms

4 4 2 Use Case 3 Alarm Group Displays

The system shall present as the main display screen a view of the highest alarm groups that is easily recognizable to the operators It is highly desirable that the layout of the tank farms be similar to the familiar schematic of East and West Tank Farms as commonly seen in the drawings such as Hanford Tank Farm Facilities Status

There are four alarm group types The first one is the tanks within a farm The second one is the farm as a whole, The third one is an external monitored system, which is an SSC outside the 200 Area, And the Fourth one is between farms

Each alarm group should display an indication of the quality-status (GOOD or UKNOWN) or the highest priority alarm for its respective alarm group The operator selects a particular alarm group and continues until the lowest alarm group is displayed The lowest alarm group provides a detailed view of the sensors contained within in the alarm group Each sensor display will indicate its value (or for discretes, its annotation for its on/off state) and the respective quality-status and alarm-status

The operator will be able to print an alarm group (from any of the alarm groups) from the main operator console

4 4 3 Use Case 4 Request Trend Graphs

The operator has a choice to review individual trend graphs or review a grouping of trend graphs (if applicable) at the lowest level trend graph

4 4 3 1 Individual Trend Graphs

The operator can review a individual sensor trend graph by selecting the desired sensor The system displays in a graphical form the sensors data as recorded over a specific time period (default is seven days)

4 4 3 1 1 Sensor History (Tabular Format)

The operator can display the recorded sensor data (sensor history) in a tabular format by clicking on the individual trend graph of a chosen sensor

4 4 3 1 2 *Sensor History Details*

The operator change the time period for which the data is displayed on an individual trend graph the user can click on the tabular data table This displays a screen that contains various time period options (i e 1hr, 8hr 24hr 7 days, 31 days) as well as other sensor history details such as highest reading during selected time period Clicking on the desired time period option automatically updates the individual sensor trend graph

4 4 3 1 3 Individual Sensor Details

The operator can view specific details associated with a sensor by selecting the button containing the letter "D" on the individual trend graph screen The sensor details displayed are different depending on the type of sensor chosen Sensor details can include

Detail	Description
Current Reading	The reading as last read
Last Good Reading at	The date and time of the last good reading
Last Update Method	The method used to get the last good reading (Get, Poll RPC)
Point Processing	Flag indicating if point processing is enabled (TRUE FALSE)
Alarm Processing	Flag indicating if alarm processing is enabled (TRUE, FALSE)
Test Processing	Flag indicating if test processing is enabled (TRUE, FALSE)
Logging	Flag indicating if logging is enabled (TRUE FALSE)
Raw Value	The value received from the io-station
Polling Freq Sec	The frequency in seconds that the io-station is polling the sensor for readings

The operator can also update the current reading of the chosen sensor by clicking on the button containing the letter "U" on the sensor details screen

4 4 3 2 Request Trend Graphs for Groups of Sensors

To review a grouping of trend graphs the operator selects from the given list of trend-graph-groups The groupings are provided by sensor type, with the exception of temperature sensors which is subdivided by riser

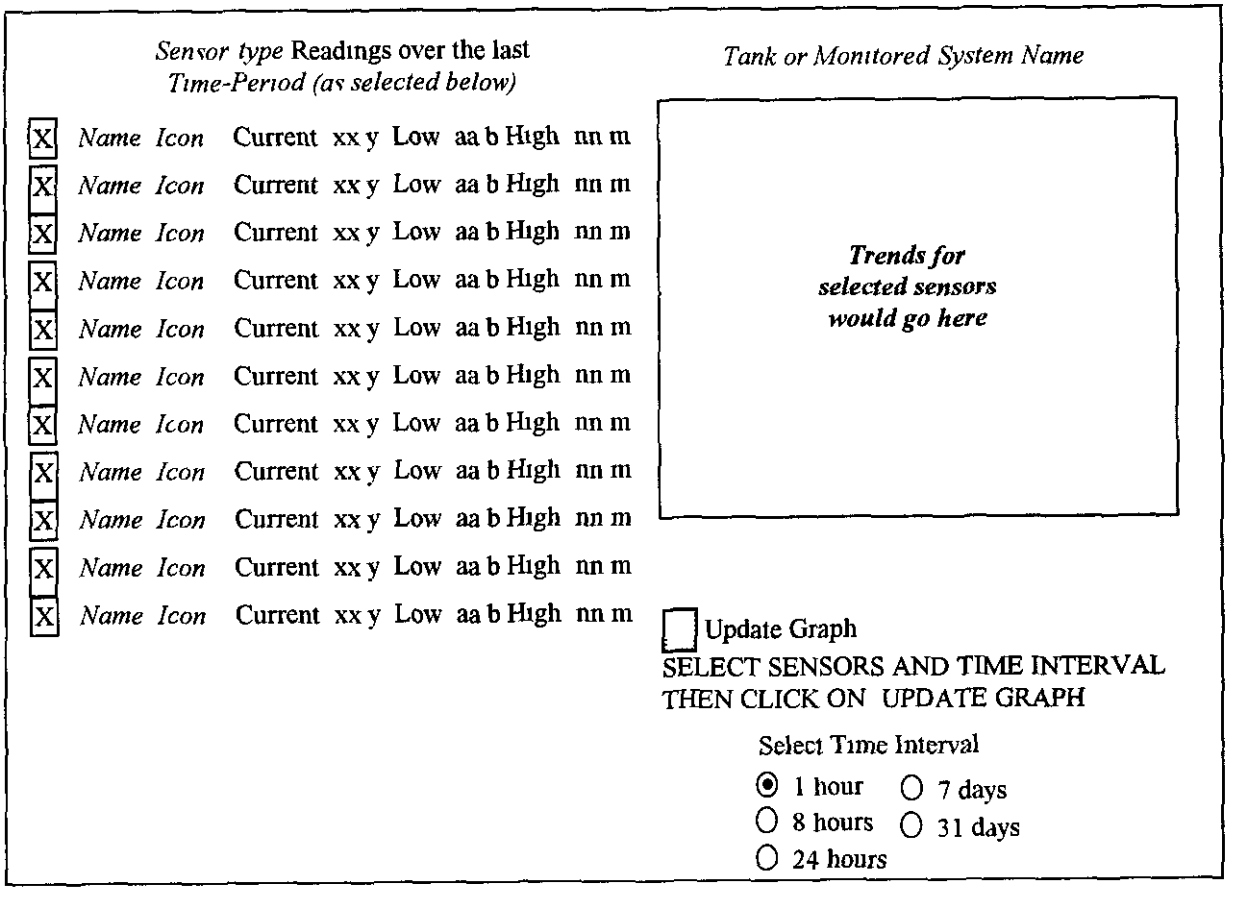
The operator selects from the provided list of sensor groups present, which could include

- All temperature sensors for a tank
- All temperature sensors on a riser (one for each riser)
- All reference temperature sensors
- All level sensors
- All hydrogen sensors
- All pressure sensors
- All flow sensors

The system displays the following window to the operator

Note that the following are "logical' placeholders and will contain information as indicated

Tank or M	The name of the tank - e g TX-118
Sensor-type	The type of sensors grouped on this graph - e g Temperature, Pressure, Level
Time-Period	1 hour, 1 day 7 days,
Name	The name of the sensor
Icon	A plot icon/marker used to uniquely identify this sensor on the trend-graph
xx y, aa b, nn m	Actual numbers representing the respective current low and high readings for the sensor



The operator selects or deselects the particular individual sensors from the group he/she desires displayed

The system indicates that the sensor is selected (e g , check box filled) or de-selected (e g , check box empty)

The operator selects the desired time interval for the trend graph display from the options

- 1 hour
- 8 hours
- 24 hours
- 7 days
- 31 days

The system indicates the selected time-interval and updates the values for each sensor on the graph for the current, low and high readings according to the readings during the selected time-interval, the graph itself is not updated yet read on

The operator then selects the Update Graph function

The system displays a trend graph of the selected sensors over the selected time interval Each sensor is discernible on the trend chart by the indicated unique plot marker beside the sensor name

If desired, the operator can then cycle through the selection process, obtaining the newly defined plot by again selecting the Update Graph function

The operator shall be able to print a selectable-trend graph on demand from the main operator console

4 4 4 Use Case 11 Request Reports

Reports are generated as needed on status of sensors, such as in service or not, with date and time stamp

4 5 Surveillance Administrator Functions/Use Cases

4 5 1 Use Case 5 Set Alarm and Instrument Limits

4 5 1 1 Set Alarm Limits

The system administrator can set alarm limits manually by changing the high or low alarm limits attribute of continuous sensors The system administrator can also initiate a process (engineering control of alarm limits) that will retrieve alarm limits that have been inputted into the SACS database externally from TMACS *Note At the time writing this document only temperature and surface level continuous sensors can be updated using engineering control of alarm limits*

4 5 1 2 Set Instrument Limits

The system administrator can set instrument limits manually by changing the high or low instrument limit atttribute of a continuous sensor

4 5 2 Use Case 6 Enable/Disable Point-Processing Functions

A User Interface for this Use Case is not specified, because these functional requirements were not mandatory to be implemented as user-functions When the surveillance administrator desires to accomplish one of these functions, an official (recorded for change-control purposes) request is made to the TMACS administrators and they accomplish the requested function

The surveillance administrator shall be able to disable/enable all point-processing functions for a sensor

When all point-processing functions for a sensor are disabled the system will

- Indicate the disabled status on the sensor's display by turning it gray
- Ignore readings from the disabled sensor

When all point-processing functions for a sensor are enabled, the system will

- Set the quality-status to UNKNOWN for the sensor
- Process readings from the disabled sensor

All sensors for a particular tank or monitored system shall be able to have all-point processing functions enabled/disabled

The surveillance administrator shall also be able to individually disable the following subsets of point-processing

- All alarm-processing
- Only rate-of-change alarm processing (i e a subset of all alarm-processing)
- Logging
- Test Mode

When all alarm-processing is disabled, the system shall skip alarm checking (unless the point is currently in alarm), but do all other enabled portions of point processing

When rate-of-change alarm processing is disabled, the system will still do alarm-processing for range limits and discrete alarms

When logging is disabled, the system will not log any readings (new or unknown) to disk

When in Test Mode the quality status will be "TEST" instead of "GOOD"

4 5 3 Use Case 7 Daily Reports

Once daily, the system shall automatically create the three reports for delivery to the surveillance administrator

- Daily Alarm Report
- Daily Equipment Failure Report
- Daily Unknown Report

The basic requirements behind these reports is that the surveillance administrator is responsible to monitor and respond to alarms and equipment problems in the tank farms On a daily basis, they must be able to respond to the following questions

- What is in alarm?
- What equipment is not communicating?
- What equipment needs repair?

When tanks (sensors) are in alarm, the surveillance administrator must provide an explanation, which will basically be what is wrong with the tank, or what is wrong with the equipment If there is a problem with the equipment, then a repair order will need to be generated

The daily reports provide an efficient way for the surveillance administrator to monitor the thousands of sensors that are connected to TMACS

4 5 3 1 Daily Alarm Report

- Alarms A snapshot report of all alarms in effect at a particular time each day

The report shall look like the following

Reading Sensor Tag Name Date/Time	Value	Units	Quality Status	Alarm Type[11]	Last Good
AN103-NSH-VP1-03J-10-1 11 40 37 a m	0	DSC	GOOD	ALARM	26 Feb 97
AN103-NXS-VP1-03J-18-1 11 40 18 a m	0	DSC	GOOD	ALARM	26 Feb 97
AN105-NSH-VP3-05J-10-1 11 45 07 a m	0	DSC	GOOD	ALARM	26 Feb 97
AN105-NXS-VP3-05J-18-1 11 45 21 a m	0	DSC	GOOD	ALARM	26 Feb 97
BY105-TI-R001-07 11 47 35 p m	78 98	F	GOOD	ALARM	2 Mar 97
C108-TI-R001-04 09 37 18 p m	68 72	F	GOOD	ALARM	2 Mar 97
C108-TI-R001-05 11 51 03 p m	68 36	F	GOOD	ALARM	2 Mar 97
S109-TI-R004-07 11 50 32 p m	69 98	F	GOOD	ALARM	2 Mar 97
S109-TI-R004-08 11 25 29 p m	62 42	F	GOOD	ALARM	2 Mar 97
SX104-FI-04J-07-1 04 27 18 a m	-0 013	CFM	GOOD	ALARM	27 Feb 97
SX104-NXS-04J-18-1 04 16 25 a m	0	DSC	GOOD	ALARM	27 Feb 97
TX105-TI-R004-04 11 47 44 p m	221 9	F	GOOD	ALARM	2 Mar 97
TX108-TI-R004-01 08 01 03 p m	59 18	F	GOOD	ALARM	2 Mar 97
TX108-TI-R004-02 05 09 23 p m	58 46	F	GOOD	ALARM	2 Mar 97
TX108-TI-R004-03 04 46 39 p m	57 38	F	GOOD	ALARM	2 Mar 97
TX113-TI-R008-10 04 26 09 p m	60 98	F	GOOD	ALARM	2 Mar 97
TX115-TI-R003-08 11 26 06 p m	62 06	F	GOOD	ALARM	2 Mar 97
TX117-LI-R11A-01A 10 41 38 a m	174 84	INCHES	UNKNOWN	ALARM	25 Feb 97
U203-TI-R004-01 09 23 16 a m	31 64	F	UNKNOWN	ALARM	2 Mar 97
U203-TI-R004-08 09 23 16 a m	31 1	F	UNKNOWN	ALARM	2 Mar 97

DSC = discrete

[11] Note This should be alarm-type per the heading, but at the point of this writing there is an outstanding problem-report indicating that for continuous points, alarm-state should be replaced with alarm-type

```
UNKNOWN = means value is out of instrument range or TMACS cannot
communicate with sensor
Date/Time of Last Good Reading = this is the date and time we were getting
values from the sensor   The time indicated here is when the value
displayed on TMACS (and here) was last known to be good
```

4 5 3 2 Daily Equipment Failure Report

- Equipment Failures A report of all equipment related errors for the previous day - e g , communication failure and respective reestablishing of communication, invalid value (exceeded instrument limits) Acknowledgments of these messages shall be included in the report

The report should look like the following

Note that the first four entries represent a logical sequence of

Lost Communication (Not acknowledged)
Acknowledgment of #1
Communication Reestablished
Acknowledgment of #3

After that are a few more examples of typical equipment error conditions/messages

---

25 Mar 97 1 15 50 a m DATA ACQUISITION EQUIPMENT
Station-ID ENRAF-STA-00-00-001
TMACS has LOST communication with this Station
NOT ACKNOWLEDGED

---

25 Mar 97 1 15 50 a m DATA ACQUISITION EQUIPMENT
Station-ID ENRAF-STA-00-00-001
TMACS has LOST communication with this Station
25 Mar 97 1 16 03 a m ALARM ACKNOWLEDGED

---

25 Mar 97 1 17 02 a m DATA ACQUISITION EQUIPMENT
Station-ID ENRAF-STA-00-00-001
TMACS has ESTABLISHED communication with this Station
NOT ACKNOWLEDGED

---

25 Mar 97 1 17 02 a m DATA ACQUISITION EQUIPMENT
Station-ID ENRAF-STA-00-00-001
TMACS has ESTABLISHED communication with this Station
25 Mar 97 1 17 05 a m ALARM ACKNOWLEDGED

---

25 Mar 97 1 25 50 a m DATA ACQUISITION EQUIPMENT
Port /dev/cuc6 Line No 0 CIU 0 GAUGE 4
[Enraf Error] 47 Enraf Field Device Not Communicating - Timed Out
NOT ACKNOWLEDGED

25 Mar 97 1 48 54 a m TAG NAME U203-TI-R004-02
Reading Out of Range -- field input or sensor error Reading -197 86
NOT ACKNOWLEDGED

4 5 3 3 Daily Unknown Report

A snapshot report of all UKNOWN readings in effect at a particular time each day

The report shall look something like the following

Good Reading Sensor Tag Name	Value	Units	Quality Status	Alarm Type[12]	Last Date/Time
AN101-LI-R02A-01A	42 47	INCH	UNKNOWN	NORMAL	2 Mar 97 08 11 18 p m
AN103-FI-VP1-03J-07-1	0 0	CFM	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
AN103-NI-VP1-03J-06-1	0 0	PCT-H2	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
AN103-NI-VP1-03J-12-1	0 0	PCT-H2	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
AN105-FI-VP3-05J-07-1	0 0	CFM	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
AN105-NI-VP3-05J-06-1	0 0	PCT-H2	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
AN105-NI-VP3-05J-12-1	0 0	PCT-H2	UNKNOWN	UNKNOWN	28 Jan 97 01 43 00 p m
BY101-TI-R001-01	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-02	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-03	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-04	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-05	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-06	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-07	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-08	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-09	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-10	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m

[12] Note This should be alarm-type per the heading, but at the point of this writing, there is an outstanding problem-report indicating that for continuous points, alarm-state should be replaced with alarm-type

BY101-TI-R001-11	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-12	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-13	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-R001-14	0 0	F	UNKNOWN	UNKNOWN	25 Feb 97 08 00 26 a m
BY101-TI-U1-RJA	90 5	F	UNKNOWN	UNKNOWN	6 Aug 96 01 21 36 p m
BY101-TI-U1-RJB	91 04	F	UNKNOWN	UNKNOWN	6 Aug 96 01 51 37 p m
BY112-TI-R015-LOW	80 42	F	UNKNOWN	NORMAL	25 Feb 97 10 01 53 a m
T109-TI-R008-11	59 54	F	UNKNOWN	UNKNOWN	20 Dec 96 05 02 02 p m
TX117-LI-R11A-01A	174 84	INCHES	UNKNOWN	ALARM	25 Feb 97 10 41 38 a m
U203-TI-R004-01	31 64	F	UNKNOWN	ALARM	2 Mar 97 09 23 16 a m
U203-TI-R004-02	58 28	F	UNKNOWN	NORMAL	1 Mar 97 05 21 25 p m
U203-TI-R004-08	31 1	F	UNKNOWN	ALARM	2 Mar 97 09 23 16 a m

DSC = discrete
UNKNOWN = means value is out of instrument range or TMACS cannot communicate with sensor
Date/Time of Last Good Reading = this is the date and time we were getting values from the sensor The time indicated here is when the value displayed on TMACS (and here) was last known to be good

4 6 TWRS Design Authority

4 6 1 Use Case 8 IO-Point & IO-Station Configuration

This use case deals with adding and configuring instrumentation from the master configuration file (the Tag List)

All configuration changes to TMACS must be done in strict compliance with the change-control procedures for the TWRS Design Authority Group

The requirement to be able to add and configure new instrumentation is another functional requirement that was not required to be accomplished by user function i e it is acceptable if this is accomplished by programmer intervention It is mandatory that the configuration of all added sensors and io-stations match the specifications given in the TMACS Input/Output (I/O) Termination Point Listing (The Tag List)

The following basic functions are required

- Add/Configure Alarm Groups (e g C-106 Sluicing)
- Add/Configure New I/O Stations
- Add/Configure New Sensors

For individual sensors, the following items shall be configured
For all io-points (continuous and discrete)

- Tag-list-names
- Label
- Display colors
- Description of the sensor
- Respective alarm group
- Respective io-station and channel
- History retention period

For continuous points only

- Delta
- Units of reading
- Formula for converting from raw instrument reading to units
- High and low instrument limits
- High and low alarm limits with their respective display colors (Display color determined administratively)
- Deadband
- Rate-of-change alarm limit and respective display color

For discrete points only, for each valid value

- Color
- Annotation
- Alarm-state

4 6 1 1 Actual Field Unit I/O Station Configuration

This is external to TMACS For example using a PC an Acromag®'s configuration data can be downloaded to it for ensuring synchronization with TMACS

4 7 Sacs Interface Use Cases

4 7 1 Use Case 9 Level Readings from SACS

4 7 1 1 Manual Surface Levels Daily

Level readings from SACs are read automatically daily Individual sensor readings can be manually retrieved

4 7 2 Use Case 10 Readings to SACS

4 7 2 1 One Reading Per Sensor Per Day

Certain sensor data are required to be stored once per day to meet regulatory requirements The readings from these sensors are transferred once per day to SACs

4 8 Long Term Data Storage Cases

4 8 1 Use Case 12 Daily Readings - "High Resolution" Data

Sensor readings that are collected by TMACS are recorded to one of two ASCII files (Discrete Sensor History File or Continuous Sensor History File) for long term storage Whenever a discrete sensor state changes it is recorded in the Discrete Sensor History File Whenever a continuous sensors reading value exceeds the delta limit, is the first reading of a new day, or an update is requested then it is recorded to the Continuous Sensor History File On a nightly basis the data in these two files are then processed and stored in a Microsoft SQL database The database can then be accessed to retrieve data without affecting the operation of the TMACS system The format of the Continuous and Discrete Sensor History Files are defined in section 8 "TMACS FILES FORMATS'

4 8 2 Use Case 13 Other Log Files Daily

There are several types of TMACS log files that are created on a daily basis for archival purposes These include

- Alarm Log Files - used for recording Sensor Alarm events
- Equipment Failure Log Files - used for recording I/O device equipment failure events
- Double-Shell Tank Daily File – a snapshot of the current readings for double shell tank sensors
- Single-Shell Tank Daily File – a snapshot of the current readings for single shell tank sensors
- Test Tank History File – used for recording the activity of a specified tank
- Discrete shutdown file - used to record the last reading for every TMACS discrete sensor
- Continuous shutdown file - used to record the last reading for every TMACS continuous sensor
- System Performance Log Files – used to record performance data every hour

The format of these files are defined in section 8 'TMACS FILES FORMATS"

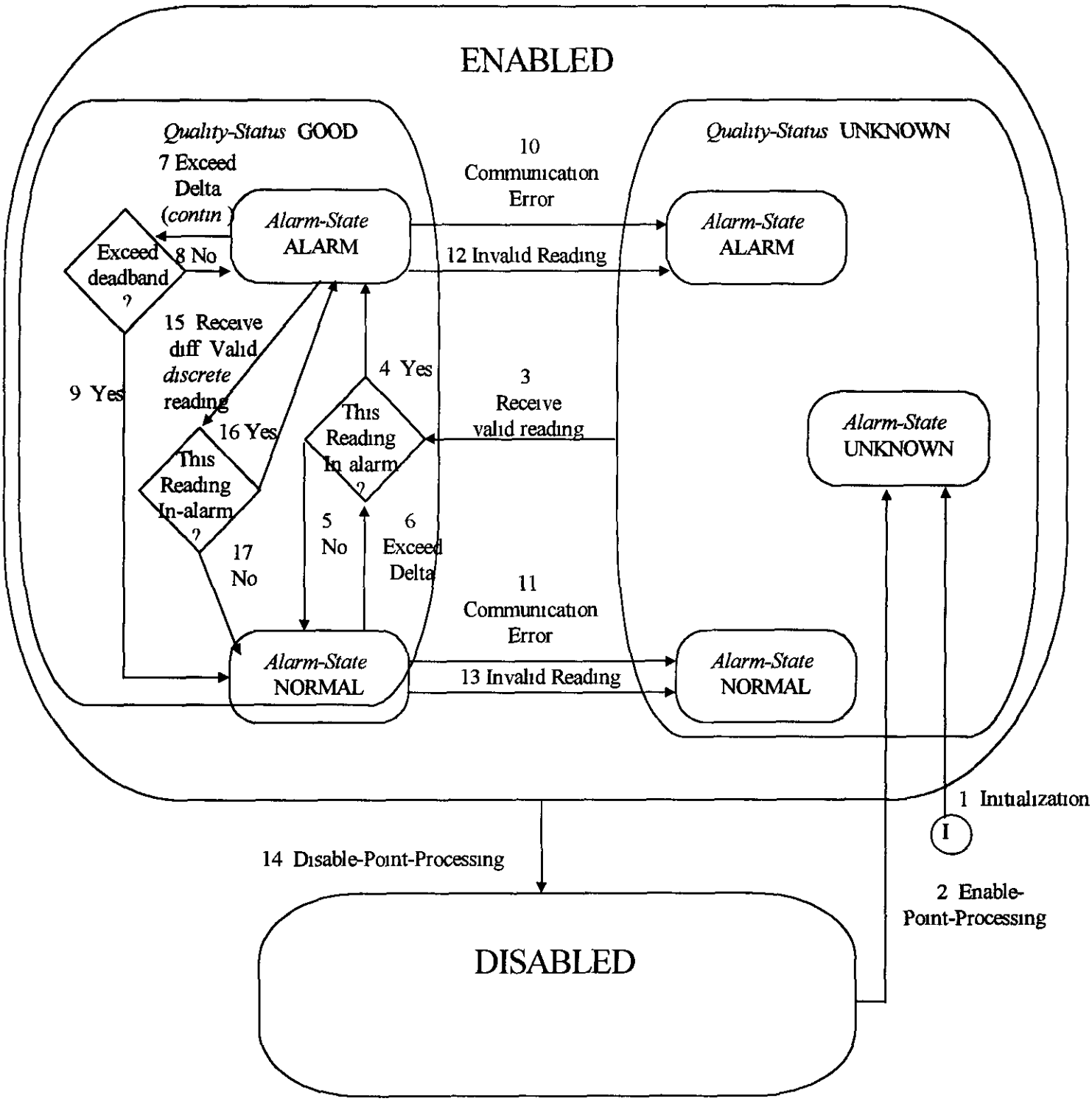
4 9 Integrated Use Case Requirements Models

4 9 1 Point-Processing

4 9 1 1 Quality-Status and Alarm-State

The following STD provides an integrated view of point processing including the combinations of quality-status and alarm-state A separate diagram will show the details of alarm-message handling

Figure 10 Integrated Point Processing STD



The following table provides a matrix view of the above integrated point processing STD The initial states are shown on the left The resultant states are shown across the top The entries represent the number of the event/transition that effects the respective state change, numbered per the transitions on the above state transition diagram

The 2-letter entries (Q S) are combinations of the possible values for Quality-Status and Alarm-State, where

Q = quality-status, and the possible values are

U = Unknown
G = Good/TEST
S = alarm-State, and the possible values are

N = Normal
A = Alarm and
U = Unknown

Table 7 Integrated Point Processing State Matrix

		Disabled	E U U	N U N	A U A	B G U (never)	L G N	ED G A
Disabled			2 1					
E	U U	14	*				3+5	3+4
N	U N	14		*			3+5	3+4
A	U A	error if try			*		3+5	3+4
B	G U (never)							
L	G N	14		11, 13			6+5	6+4
ED	G A	error if try			10, 12		7+9 or 15+17	7+8 or 15+16

As discrete sensors do not have an alarm-type, the above table is sufficient to describe the behavior of discrete points The following table gives a modified/expanded state matrix view of integrated point processing specific to continuous io-points including range-alarms only (ROC alarms are not included) The following explicit changes have been made to the "pure" state matrix above -

- The disabled rows and columns were removed (they would be the same)
- Alarm Types of Low (L), High (H) were added to each case where alarm-state = Alarm
- Alarm Type of In-Range (I) was added to each case where alarm-state = Normal
- Alarm-Type of undefined (x) was added to each case where alarm-state = Unknown
- Refinements of Transition 4 were noted as follows
  - 4l In alarm exceeds low-alarm-limit
  - 4h In alarm exceeds high-alarm-limit

Table 8 Integrated Point Processing Continuous IO-Point State Matrix.

	UUX	UNI	UAL	UAH	GNI	GAL	GAH
	1						
UUX	*				3+5	3+4l	3+4h
UNI		*			3+5	3+4l	3+4h
GAL			*		3+5	3+4l	3+4h
GAH				*	3+5	3+4l	3+4h
GNI		11, 13			6+5	6+4l	6+4h
GAL			10,12		7+9	7+8	7+**
GAH				10, 12	7+9	7+**	7+8

- * = When quality-status=UNKNOWN, any invalid reading or communication-error will NOT cause a state change

- ** = This is theoretically possible (to go from Low-Alarm to High-Alarm or vice-versa) IF the time between readings was large enough It isn t shown in the composite STD, but is shown in the Range Alarm STD in the point-processing section

1 Transition Initialization
Triggered by initialization
Action { set quality-status unknown and alarm-state unknown }

Note there are actions done at startup but these are outside the scope of point-processing At this point, it's important to note only that all points are initialized to quality-status and alarm-state of UNKNOWN at system startup The details will be described in the Initialization section of this document

2 Transition Enable-Point-Processing
Triggered by {User-initiation}
Action {Set quality-status and alarm-state to UNKNOWN enable point-processing and wait for reading}

3 Transition Receive valid reading
Triggered by {Get-reading, Delta exceeded and IF value within the specified instrument limits for continuous or is defined for discrete}
Action {Process-Changed Point Do Alarm Processing (to determine full transition)}

NOTE Receiving a valid reading transitions the quality-status to GOOD but the alarm processing determines the alarm-state

4 Transition In Alarm?/YES
Triggered by {Alarm-Processing found in-alarm point and alarm processing is enabled}
Action {Annunciate-Alarm (create new alarm message), if logging enabled Log reading}

5 Transition In Alarm?/NO
Triggered by {Alarm-Processing found normal/non-alarm point or alarm processing disabled}
Action {Annunciate normal reading if logging enabled Log reading}

6 Transition Exceed Delta
Triggered by {Get-reading IF value differs from previous reading by +/- delta }
Action {Process-Changed Point Do Alarm Processing (to determine full transition)}

NOTE For discrete points, exceeding the delta is just getting a different value than the previous reading

7 Transition Exceed Delta (continuous points)
Triggered by {Get-reading, IF value differs from previous reading by +/- delta for continuous io-points}
Action {Process-Changed Point Do Alarm Processing (to determine full transition)}

NOTE This transition is ONLY for continuous io-points See Transition 15 for the equivalent for discrete io-points

8 Transition Exceed Deadband?/NO
Triggered by {reading > high-limit - deadband or reading < low-limit + deadband}
Action {No new alarm, if logging enabled Log reading}

NOTE This transition relevant ONLY for continuous io-points

9 Transition Exceed Deadband?/YES
Triggered by {reading < high-limit - deadband or reading > low-limit + deadband}
Action {Process-Changed Point Do Alarm Processing (to determine full transition)}

10 & 11 Transition Communication Error
Triggered by {Communication Error while trying to obtain reading}
Action {Process-Unknown}

NOTE The requirement reflected here is that the system remember the alarm-state (and alarm-type for continuos points) when the quality-status goes UNKNOWN

12 & 13 Transition Invalid reading
Triggered by {Get-reading, IF value exceeds the specified instrument limits for continuous or is undefined for discrete}
Action {Process-Unknown}

NOTE The requirement reflected here is that the system remember the alarm-state and alarm-type (continuous) when the quality-status goes UNKNOWN

14 Transition Disable-Point-Processing
Triggered by {User initiated, IF alarm-state is not ALARM}
Action {Disable-point-processing Subsequent readings for this sensor ignored until enabled It is desirable that the quality-status be updated to reflect the disabled state }

NOTE Alarm processing for any io-point may not be effectively disabled while the point is in ALARM i e The disable shall not take effect while the point is in alarm

15 Transition Receive diff valid discrete reading
Triggered by {Get-valid-reading, IF value differs from previous reading}
Action {Process-Changed Point Do Alarm Processing (to determine full transition)}

NOTE This transition is ONLY for discrete io-points

16 Transition In Alarm?/YES
Triggered by {value defined/configured as in ALARM}
Action {Re-Alarm, Annunciate-Alarm (create new alarm message), if logging enabled Log reading}

NOTE This transition is ONLY for discrete io-points

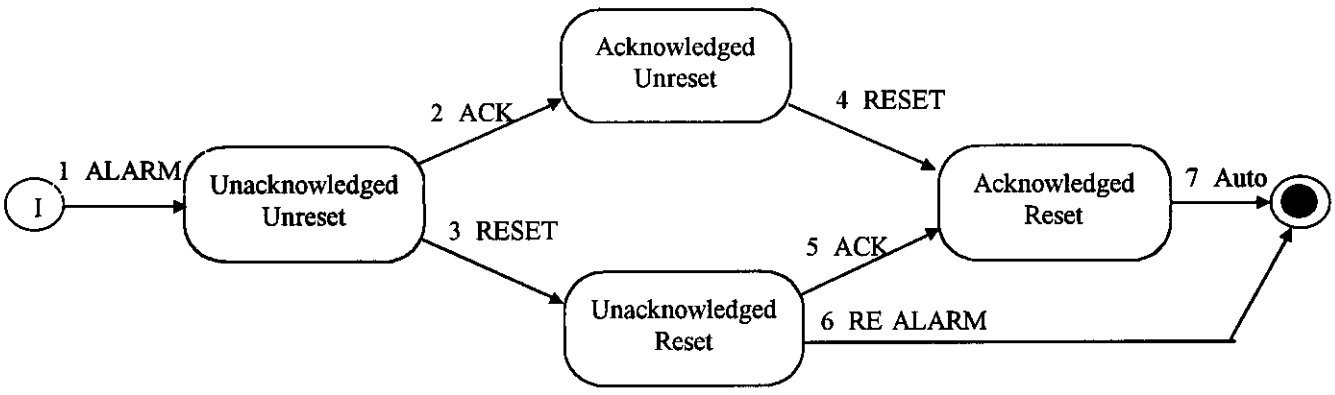
17 Transition In Alarm?/NO
Triggered by {value defined/configured as NORMAL}
Action {Annunciate normal reading if logging enabled Log reading}

NOTE This transition is ONLY for discrete io-points

4 9 1 2 Alarm Message Handling (Point Processing & Operator Acknowledgment)

The following STD shows the integration between point-processing functions and operator acknowledgment of alarms The states are the states an alarm-message The initial state represents the no-alarm condition that exists before an alarm message is created The final state indicates the nil afterlife" of an alarm-message once it has been destroyed/deleted

Figure 11 Integrated State Transition Diagram for Alarm Message Handling



The following table gives a state-matrix view of the alarm-message STD The initial states are shown on the left The resultant states are shown across the top The cells contain the transitions that effect the respective state change, numbered per the STD

Table 9 State Matrix for Integrated Alarm Message Handling

		Unacknowledged		Acknowledged		Gone
		Unreset	Reset	Unreset	Reset	
Initial State		1				
Unacknowledged	Unreset	*	3	2		
	Reset		*		5	
Acknowledged	Unreset			*	4	
	Reset					7

- * = The alarm message will remain in the same state (no transition) under the following conditions
  - For continuous io-points new readings, but the deadband is not exceeded
  - For discrete io-points the same in-alarm reading (value) is received
- Gone represents the terminated/annihilated "state" (i e this puppy has been destroyed/removed)

1 Transition ALARM
Triggered by {Alarm-Processing Detects an in-alarm point}
Action {Create Alarm Message indicating unacknowledged, log alarm}

2 Transition ACK
Triggered by {Operator acknowledging alarm-message}
Action {Indicate alarm acknowledged log acknowledgment}

3 Transition RESET
Triggered by {Receiving non-alarm value for this io-point}
Action {Indicate alarm reset calculate duration of alarm log reset (done as logging normal reading)}

4 Transition RESET
Triggered by {Receiving non-alarm value for this io-point}
Action {Indicate alarm reset calculate duration of alarm log reset (done as logging normal reading), kick off transition 7}

5 Transition ACK
Triggered by {Operator acknowledging alarm-message}
Action {Indicate alarm acknowledged, log acknowledgment, kick off transition 7}

6 Transition RE-alarm
Triggered by {Receive another alarm for this io-point while Quality-Status = UNKNOWN}
Action {Remove old alarm message - Annunciate alarm (create new alarm message) via new transition #1}

NOTE There are several ways for this to happen e g when the sensor is already in the alarm-state AND the quality-status cycles from Good to Unknown and back to Good IF the quality status stays GOOD,

then there is no re-alarm If it stays Unknown, then a new reading can't be in alarm Note also that the reading that caused the RE-ALARM event also would be initiating another ALARM event

A "re-alarm" is when the quality-status of an in-alarm point goes from GOOD to UNKNOWN and back to GOOD, but the in-alarm reading didn't change In this case, the system shall generate a new alarm message when a sensor re-alarms due to communication or instrument failure See the functional test for a definitive list of the ways this can happen

7 Auto
Triggered by {Arrival at Acknowledged and Reset state}
Action {Delete/Remove the alarm from the current-alarm messages}

5 TMACS HIGH-LEVEL DESIGN

5 1 Tmacs System Design Overview

This section provides an overview of how TMACS is put together from a system perspective (software and hardware)

5 1 1 TMACS Process (Hardware) Diagram and Description

The purpose of the process diagram is to illustrate the target hardware for the Tank Monitor and Control System, and show how the key processes are allocated The diagrams are split out showing the development and the production environments

Figure 12 TMACS Process (Hardware) Diagram - Development Environment

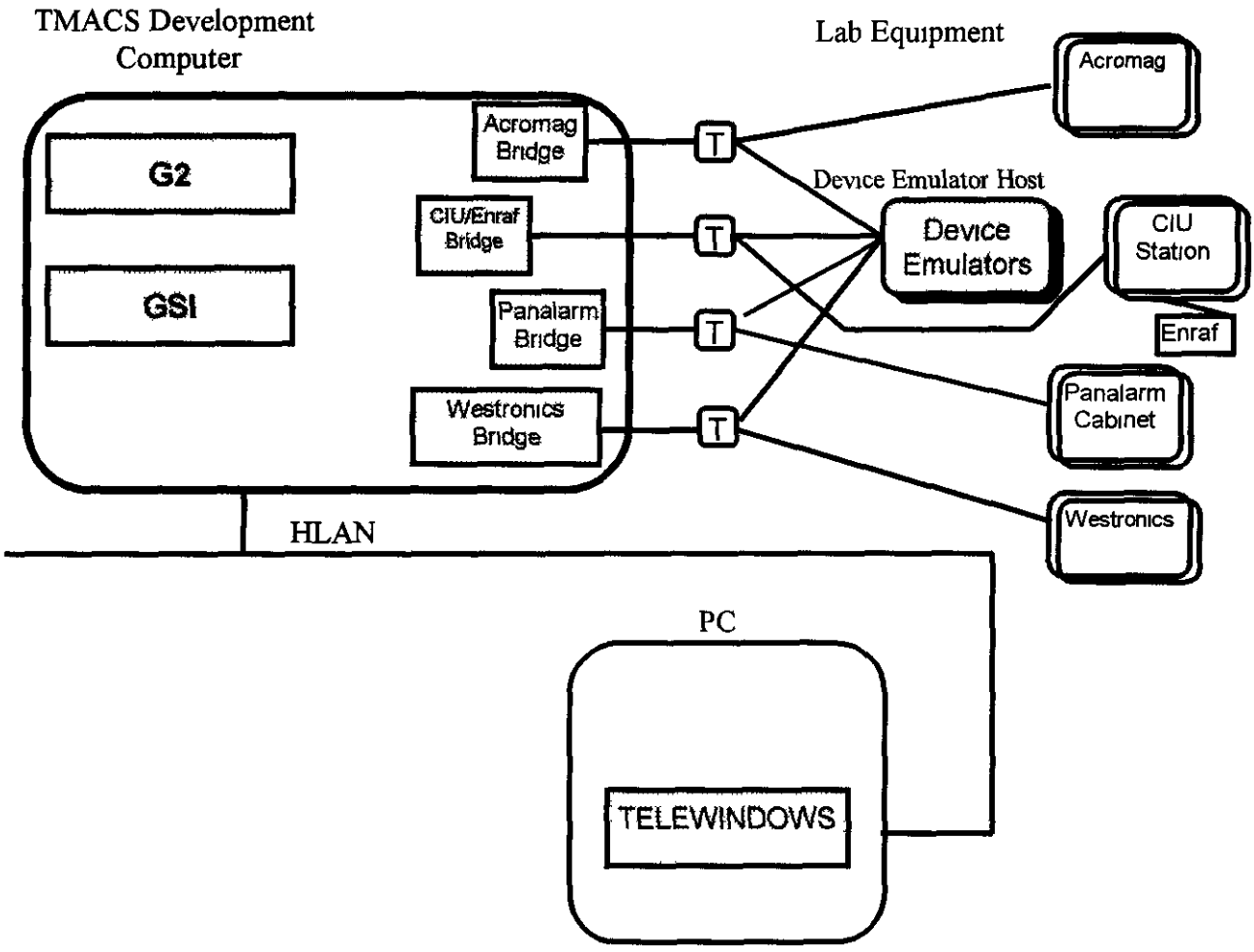
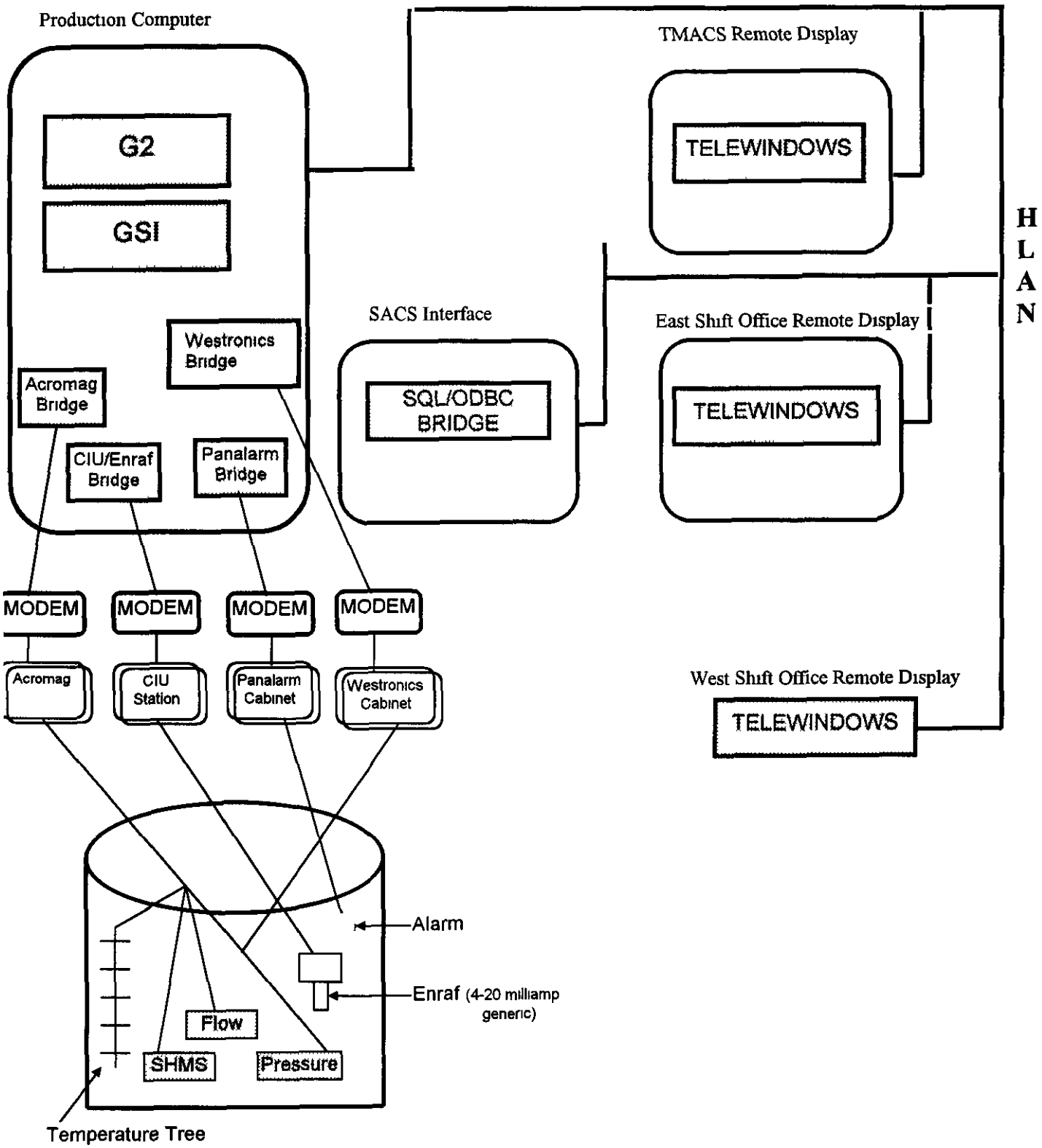


Figure 13 TMACS Hardware Diagram - Production Environment



5 1 2 TMACS Subsystem Module Hierachy

The following section contains the TMACS Subsystem Module Hierarchy

Table 10 TMACS Subsystem Moduel Hierachy

Top Level	Level 1	Level 2	Level 3	Level 4
TMAC MAIN	TMACS-LIB	GFI-MOD	SYSMOD	UILROOT
			GENSYM-RAISED-BUTTONS	
		TMACS-DEFINITIONS	GENSYM-RAISED-BUTTONS	
			G2-DATABASE	
		GENSYM-RAISED-BUTTONS		
	ACROMAG-ROOT	GENSYM-RAISED-BUTTONS		
		TMACS-DEFINITIONS		
	ENRAF-ROOT	LEVEL		
		GENSYM-RAISED-BUTTONS		
		TMACS-DEFINITIONS		
	PANALARM-ROOT	SYS-MOD		
		TMACS-DEFINITIONS		
	WESTRONICS-ROOT	GENSYM-RAISED-BUTTONS		
		TMACS-DEFINTIONS		
	ODBC-ROOT	G2-DATABASE		
		TMACS-DEFINTIONS		
	SURVEILLANCE-LIMITS-ROOT	ODBC-ROOT		
	PRINTER-ROOT	GENSYM-RAISED-BUTTONS		

Top Level	Level 1	Level 2	Level 3	Level 4
		TMACS-DEFINITIONS		
	HTFF	TMACS-DEFINITIONS		
	DATA-RECOVERY	GENSYM-RAISED-BUTTONS		
		SYS-MOD		
	POINT-PROCESSING	GENSYM-RAISED-BUTTONS		

6 OBJECT-ORIENTED MODELS

6 1 Object Model (Class Hierarchy)

This section contains object models showing the existing class hierarchy with data attributes and relationships

6 1 1 Object Model Notation

The following notation is a subset of OMT (Object Modeling Technique, James Rumbaugh, et al) Object Model notation

Figure 14 Object Model Notation, Part 1

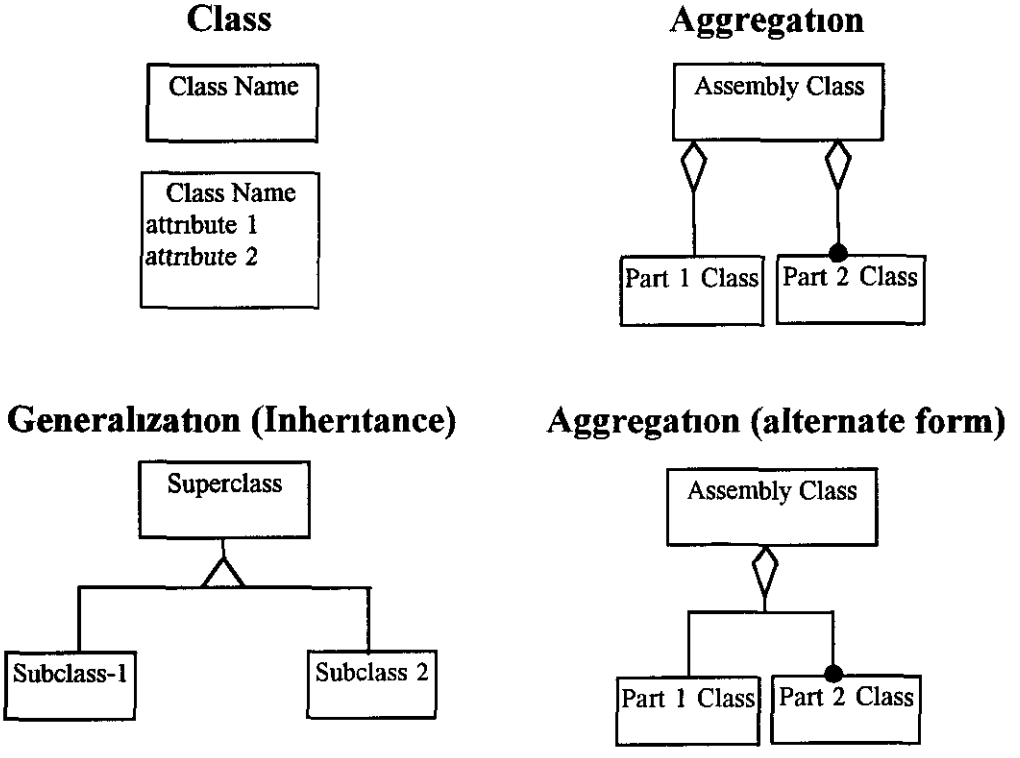
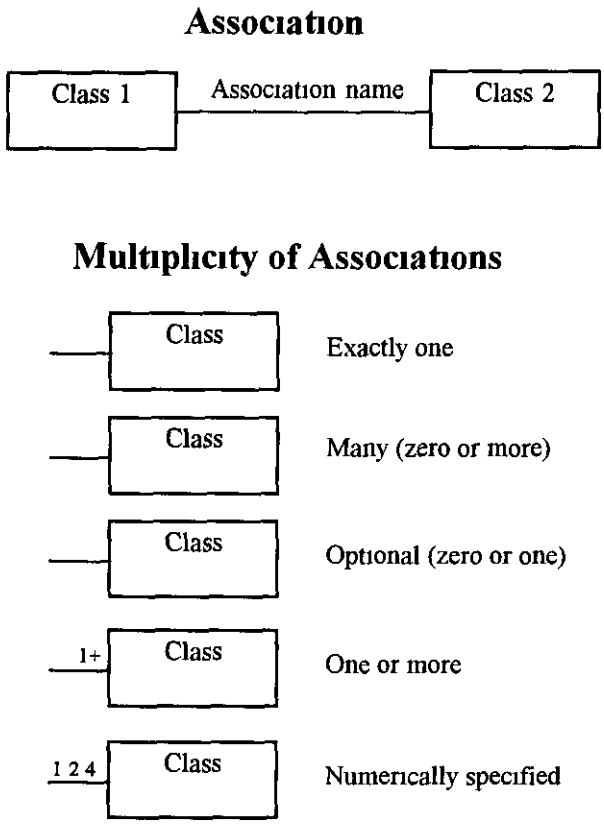


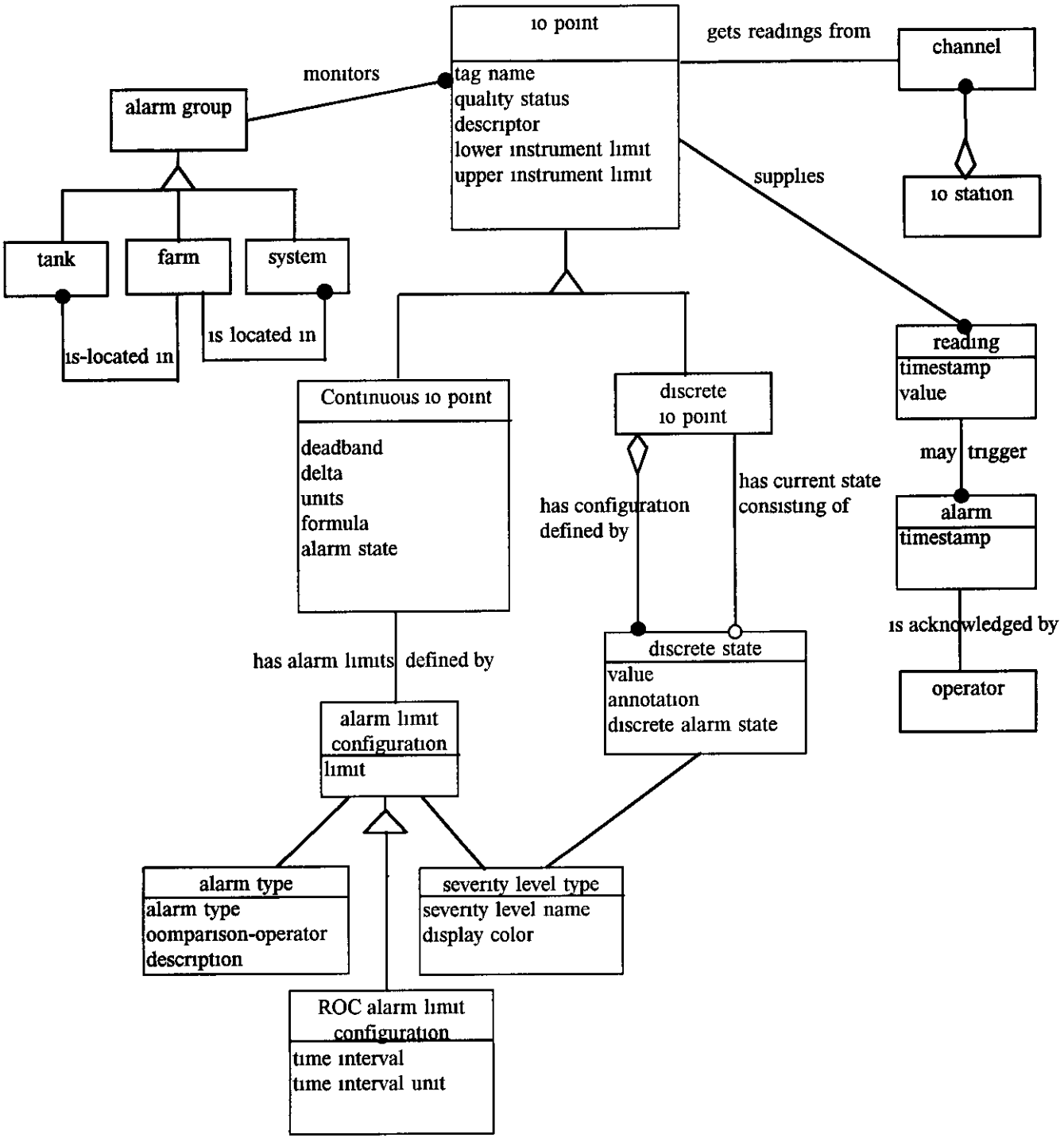
Figure 15 Object Model Notation, Part 2



6 1 2 Conceptual Object Model (Class Hierarchy)

This is an idealized conceptual model of the class hierarchy from an analysis/ideal perspective It is organized according to the relationships that exist between the modeled objects in the business or ' real-world " It is NOT how the class hierarchy is actually implemented, see the next section Physical Object Model (Class Hierarchy) " for that

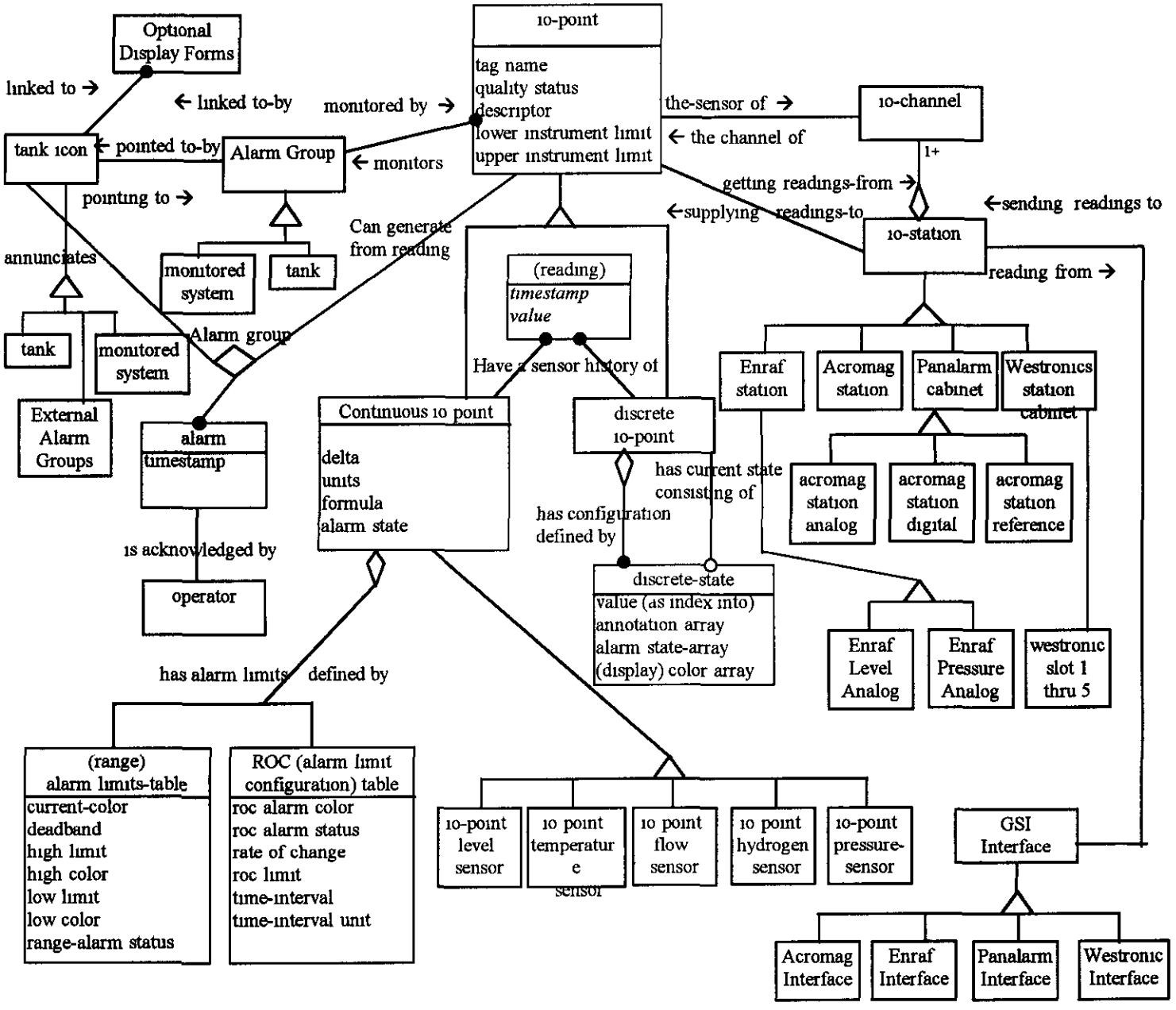
Figure 16 Conceptual Object Model (Class Hierarchy)



6 1 3 Physical Object Model (Class Hierarchy)

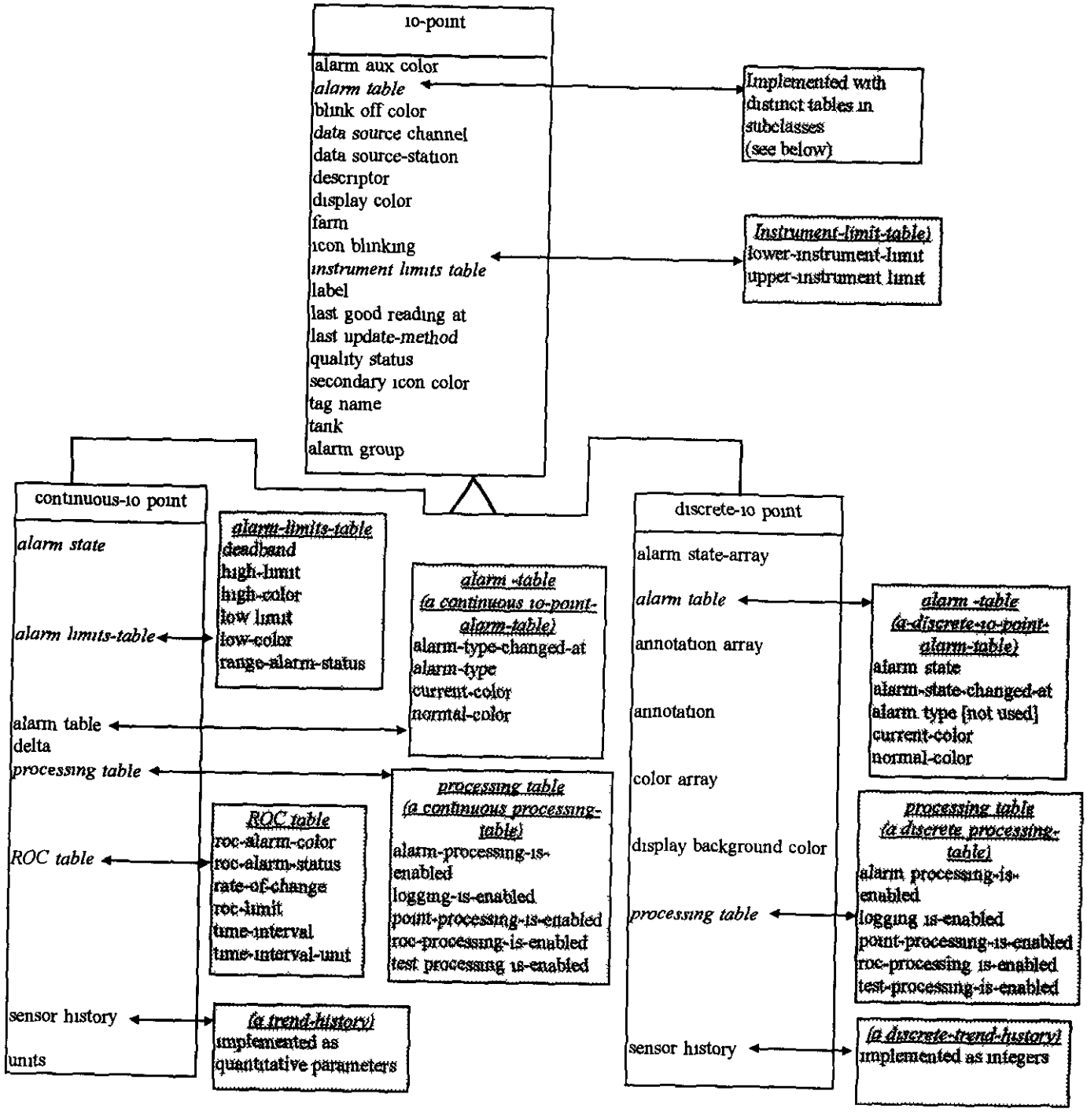
This model reflects the existing TMACS as-built implementation To the standard OMT notation, relationships implemented in G2® as relationships, have been shown with relationship-name → The arrow indicates the direction of the relationship Relationships without arrows are not implemented directly as relationships

Figure 17 As-Built Object Model (Class Hierarchy)



The purpose of the following diagram is to show the detailed physical structure including sub-tables of 10-point and its subclasses, continuous-10-point and discrete-10-point The details of all attributes and tables are given in the data dictionary

Figure 18 Detailed As-Built Object Model for IO-Point



6 2 Data Dictionary

CAVEATS/NOTES

- Attributes are named as they exist in the current TMACS application
- Per the normal rules and understanding of class structures involving inheritance, all attributes of a superclass are assumed to be inherited by its subclasses, retaining the same format and definition unless otherwise noted The attribute definitions for subclasses will only contain additional or changed attributes i e attributes defined in the superclass (e g io-point) will not be repeated in the respective subclasses (e g continuous-io-point) unless they are changed in some way
- For convenience, attributes are listed in alphabetic order Attributes from sub-tables in G2® are also listed in alphabetic order along with the other attributes, with a note indicating what table they are from Attributes which are themselves table names, are also listed alphabetically, with their respective attributes listed The definition is given with the individual entry This duplication is intentional for the purpose of making this document more useful to the maintenance programmers

6 2 1 Object Class IO-Point

Superclass G2® Quantitative Variable

Definition An io-point is the representation of either a physical sensor or a derivation of one or more physical sensors that is used to monitor the value of some parameter of a waste tank or other system of interest Common sensor types are level and temperature

Identifier tag-name
Need to describe tag-name conventions here

ATTRIBUTES

Alarm-aux-color	Used as the color for the border area around a sensor icon Always transparent (effectively same as display-color) unless the quality status is UNKNOWN and there was a previous existing alarm state In this situation, alarm-aux-color is set to the alarm-display-color of the previous alarm-state and communicates the information that an alarm-state existed before the quality status went UNKNOWN Domain [a color] Type symbol
Alarm-processing-is-enabled (in processing-table)	Indicates whether alarm processing is enabled or not Domain [TRUE/FALSE] Type

Alarm-Group	The alarm group the sensor is monitoring
Alarm-state-changed-at (in alarm-table) (in alarm-state TABLE)	The date/time of the last alarm state change Domain [a **time-stamp** DD Mon YY HH MM SS]
Alarm-table [TABLE]	Implemented with subtypes of io-point-alarm-table, (continuous-io-point-alarm-table & discrete-io-point-alarm-table), both having the identical attributes of alarm-state alarm-state-changed-at alarm-type current-color normal color [See notes in continuous & discrete io-points for usage]
Alarm-type (in alarm-table)	The type of continuous io-point alarm [really should be added there] Domain [AL | AH | ROC | NA] (Alarm-High, Alarm-Low, Rate-of-Change, or No-Alarm respectively) Type symbol
Blink-off-color	The color to display when the icon blinks "off" For sensors it is always set to gray for a contrast For aggregate-icons (e g tanks or monitored systems) it is used to indicate multiple alarm states and reflects the next-highest priority state color when more than one alarm-state exist in the respective sensor set Domain [symbol white|grey| red|yellow]
Collection-time [G2® Built-in]	The time which the parameter received a value - i e the time the reading was received from the io-station This is a G2® feature of all variables and parameters Domain [legal date/time] Type float
Current-color (in alarm-table)	The current color of the sensor? NEED TO DETERMINE DIFFERENCE WITH DISPLAY COLOR! Domain [symbol green|white| grey| red|yellow]

Data-source-channel	The io-channel that holds the raw value from the sensor Derived from the relationship *the-channel-of* (io-point *is the-sensor-of* channel/ channel is *the-channel-of* the io-point) Domain [See channel] Type
Data-source-station	The io-station that is supplying-readings-to this io-point Derived from relationship *getting-readings-from* (*getting-readings-from / supplying-readings-to*) Domain [See io-station]
Descriptor	The sensor descriptor field from the tag-list giving a description of the type and use for the sensor Domain [a 40 character alpha-numeric string]
Display-color	Holds the current display color, For discrete points, it is derived from the configuration vector per the current value Domain [symbol green | white | gray | red | yellow]
Farm	The farm the sensor is associated with Not stored in a specific G2®-relationship Domain [see tank-farm]
Formula [G2® Built-in]	The conversion formula to convert the raw io-sensor value as reported by the sensor to the human understandable **last-recorded-value** number per the given **unit**
Icon-blinking	Indicates if the icon is blinking, representing an unacknowledged alarm Domain [TRUE/FALSE] Future This should be a derived attribute from a to-be created attribute like *unacknowledged-alarm-exists* [T/F]
Instrument-limits-table [TABLE] containing lower-instrument-limit upper-instrument-limit	The boundaries for valid instrument readings Outside of this range, the quality-status is = UNKNOWN and the value is indeterminate These exist for both continuous and discrete io-points They are implemented in different sub-tables (io-point-instrument-limits table and discrete-io-point-instrument-limits table respectively), **but** they have exactly the same contents For the discrete points, this is a "design/implementation kludge" taking advantage of the fact that all discretes used so far happen to have a consecutive range of legal values (usually 0,1)

Label	A user supplied abbreviation of the tag-list name for display of the point on the user interface Domain [An N? character alphanumeric string]
Last-good-reading-at	The timestamp of the last reading with a quality-status of GOOD Domain [**time-stamp** mm/dd/yy hh mm ss]
Last-recorded-value [G2®-Built-In]	The value of the last reading with a quality-status of GOOD While processing an incoming point, this will be used to reference the LAST good reading, and if the new reading (accessed via *value*) is *good* this is *updated* A *synonym* to "value" (another G2®-Built-In) Domain [value between lower-instrument-limit and upper-instrument-limit]
Last-update-method	The method in which the reading was obtained Examples are GET POLL and RPC
Logging-is-enabled (in processing-table)	Indicates whether logging is enabled or not Domain [TRUE/FALSE]
Lower-instrument-limit (in instrument limits table)	The lowest valid reading for this sensor Domain [value]
Name [G2®-built-in]	The unique name This is a G2®-built-in feature for ALL objects
Normal-color (in alarm-table)	The non-alarm color of the sensor? Not sure how this is used? NEED TO FIND OUT Is it always GREEN? Domain [symbol green | ?]
Point-processing-is-enabled (in processing-table)	Indicates whether point-processing is enabled or not Point processing includes all processing including logging and alarm processing Domain [TRUE/FALSE]
Processing-table [TABLE]	Table of on/off variables for disabling/enabling aspects of processing alarm-processing-is-enabled *logging-is-enabled* point-processing-is-enabled roc-processing-is-enabled testing is enabled

	NOTE As implemented, continuous-processing-table and discrete-processing-table are subtypes of processing-table, BUT contain EXACTLY the same attributes ROC processing doesn't have meaning for discretes
Quality-status	The representation of the goodness" or validity of the io-point's value A quality status of GOOD/TEST indicates normal communication with the sensor and the value is within instrument limits and not in test mode A quality status of UNKNOWN indicates that currently a reading is not available from the sensor Domain [quality-status-parameter (a symbol) GOOD | UNKNOWN
Secondary-icon-color	At initialization set to white for enabled points or gray for disabled points Used in design to do blinking of unacknowledged alarms by alternately holding the blink-off-color of the point and then the display-color of the point See procedure blink-icons Domain [a color]
Sensor-history	Previous readings which are paired values with time-stamps Implemented via quantitative parameters with history on Continuous points use "trend-history" (quantities) and discretes' use discrete-trend-history (integers)
Tag-name	The tag-list name of this sensor Domain [alpha-numeric string by conventions]
Tank	The tank the sensor is monitoring Derived from relationship *monitoring (monitoring/ monitored-by)* Domain [See tank]
Testing-Is-Enabled (InProcessing Table)	Indicates whether testing is enabled or not
Upper-instrument-limit (in instrument-limits-table)	The highest valid reading for this sensor Domain [A floating point number]
Value [G2® Built-in]	The value of the latest reading Saved in last-good-reading by point processing This is a G2® feature of all variables and parameters Domain [as specified by the type of the sensor- usually a float]

6 2 1 1 Object Class Continuous IO-Point

Superclass IO-Point

Definition A continuous io-point is an analog io-point continuous/analog sensors constantly are measuring a value that can vary within some range e g a temperature sensor

Identifier tag-name (from io-point)

ATTRIBUTES

Alarm-limits-table [TABLE]	A collection of alarm limit information containing deadband high-color high-limit low-color low-limit range-alarm-status
Alarm-state [value PLUS TABLE]	Value/Table describing the current alarm-state, includes value alarm-type alarm-type-changed-at Domain [symbol normal | alarm] (per update-sensor-alarm-status) NOTE A continuous-io-point-alarm-table is used for this, BUT Bob indicates that NONE of the table attributes (besides the last-recorded-value) are used here - i e alarm-type, normal-color, current-color and alarm-type-changed-at are not used from this table They ARE used from the continuous-io-point-alarm-table of the alarm-table attribute BUT per code, at least alarm-type and alarm-type-changed-at are set in update-sensor-alarm-type
Alarm-state-changed-at (in alarm-state TABLE)	The date/time of the last alarm state change Domain [a **time-stamp** DD Mon YY HH MM SS]
Alarm-table [TABLE] [DIFFERING USAGE]	Implemented with subtypes of io-point-alarm-table, (continuous-io-point-alarm-table & discrete-io-point-alarm-table), both having the identical attributes of alarm-state alarm-state-changed-at alarm-type current-color normal color

Alarm-type (in alarm-state subtable)	The current composite alarm state = range-alarm-status + ROC-alarm-status Domain (Range-alarm-status) [low-alarm, high-alarm, in-range] Domain (ROC-alarm-status) [roc-alarm, roc-in-range] Domain (Composite) [Any combo of 1 of each of above, except *in-range/roc-in-range = no-alarm*]
Deadband (in alarm-limits-table)	The amount the value must change once it goes into alarm in order for the point to go out of the alarm state Used to stop alarm chatter Domain [A floating point number]
Delta	The amount the **last-recorded-value** must change by in order to be considered a new or different value Domain [A floating point number]
High-color (in alarm-limits-table)	The alarm display color for the high-limit alarm See note on low-alarm-color
High-limit (in alarm-limits-table)	When the value exceeds this limit, a high-limit range alarm (HIGH-ALARM) is initiated Domain [A floating point number]
Low-color (in alarm-limits-table)	The alarm display color for the low-limit alarm Future This should be a symbolic like CAUTION, WARNING, INFORMATION so could be changed one place
Low-limit (in alarm-limits-table)	When the value goes below this limit a low-limit range alarm (LOW-ALARM) is initiated Domain [A floating point number]
Range-alarm-status (in alarm-limits-table)	Current status of the range alarms Domain [symbol in-range | low-alarm | high-alarm]
Rate-of-change (in roc-table)	Current rate of change of the value Domain [A floating point number]
roc-alarm-color (in roc-table)	The color for the ROC alarm display Domain [symbol red | yellow]
roc-alarm-status (in roc-table)	Current state of ROC alarm Domain [ROC-IN-RANGE, ROC-ALARM]

roc-limit (in roc-table)	Current rate of change of the value Domain [A floating point number]
roc-processing-is-enabled (in processing-table)	Indicates whether or not Rate-of-Change processing is enabled Domain [TRUE/FALSE]
roc-table [TABLE] roc-alarm-status roc-alarm-color roc-limit time-interval time-interval-units	Rate of Change alarm configuration information table - Changing more than **ROC-limit** number of **units** within **time-interval** number of **time-interval-units** (e g 4 hours) will initiate an ROC-alarm
Time-interval (in roc-table)	The magnitude of the time interval for testing for an ROC alarm See roc-table explanation Domain [A floating point number]
Time-interval-units (in roc-table)	The respective units for **time-interval** for testing for an ROC alarm See roc-table explanation Domain [A floating point number]
Units	The units for the sensors reading in **last-recorded-value** Domain [inches, degrees C]

6 2 1 2 Object Class Discrete IO-Point

Superclass IO-Point

Definition A discrete io-point is a sensor with a pre-defined number of distinct values, each of which represents a particular state

Identifier tag-name (from io-point)

ATTRIBUTES

Alarm-configuration-vector (conceptual) includes **alarm-state-array**	Each vector includes the following for each legal discrete value (which is an index into the array) The alarm state Domain [symbol normal | alarm]
annotation-array	The meaning of the state Domain [symbol An alpha-numeric string]
color-array	The display color for this state Domain [symbol green | red | yellow | blue]

Alarm-table [TABLE] [DIFFERING USAGE]	**alarm-type** is NOT used Annotation explains the meaning of the alarm Alarm-type is not relevant for discretes
Annotation	The current annotation for the discrete state Currently this is a direct attribute, while alarm-state and current-color are in the alarm table
Current-color (in alarm-table)	For discrete points only this is used in the free-from table display Possibly used to toggle the colors when blinking
Current-state (conceptual) includes	The current state vector/information
last-recorded-value	current value
alarm-state [in alarm-table]	current alarm state (ALARM or NORMAL)
annotation	meaning of current state
display-color	current display color
Color-array	See above under alarm-configuration-vector Domain [symbol-array]
Display-bkgd-color	Used for updating the background color of the free-form tables that implement discrete status display

6 2 2 Object Class IO-Station

Superclass multiple inheritance from gsi-message-service, gsi-data-service and quantitative-variable

Definition An io-station is an input/output device that collects input signals from sensors and passes them to a requesting system The station has one or more channels, which hold the actual raw values

Identifier tag-name (from io-point)

ATTRIBUTES

GSI®-interface-name	The particular GSI® interface object that this station is communicating with Note that this is a built-in" G2® attribute that comes from GSI®-data-service

GSI® variable status	Reflects the status of the GSI® interface This is how the station knows if communication has been lost (i e an error code shows up here) Domain integer
Last-Update-Method	Indicated the method last used to obtain a reading from the sensor Valid values are Get – Used at initialization Poll - Normal method of obtaining a reading RPC – Remote Procedure Call, used when operator request a reading
Station-status	Status of this io-station Domain integer

6 2 2 1 Object Class Acromag®-Station

Superclass IO-Station

Definition An Acromag® is a particular brand of io-station

Identifier station (unique numeric identifier)

ATTRIBUTES

Station	The unique numeric identifier assigned to each Acromag® station
ChNN	Note - particular channels are defined ONLY in the actual instances of acromag®-stations Typically acromag®-stations have 16 channels, numbered either Ch01 - Ch16 or Ch17 - Ch32 consecutively and each of these (e g , Ch17) shows up as an attribute in the instance of the acromag®-station These channels are what hold the actual io-device reading which comes from the sensor and is passed, from here to G2® Type either discrete-acro-channel (discretes) or acr-channel-parameter (analog) or reference
Mask	Used to indicate which channels are actually present/active in the acromag® The channels that are "masked off" are not active It's an 8 bit mask Domain [ ]
Obj-index	Internal pointer used by GSI® to pass data back and forth to the

	right io-station This must be unique across all io-stations for a particular bridge Domain [1 N]
Type	The type of devices attached to the Acromag® Or the mode the Acromag® is running in Domain [analog, digital reference]
Polling-freq	The polling frequency of the station Type acr-channel-parameter
Line Number	When used identifies the communication line number associated with the code operated switch

6 2 2 1 1 Object Class Acromag®-Station-Analog

Superclass Acromag®-Station

Definition An Acromag®-Station-Analog is an Acromag® configured for continuous sensors (ch 1-16)

ATTRIBUTES

6 2 2 1 2 Object Class Acromag®-Station-Digital

Superclass Acromag®-Station

Definition An Acromag®-Station-Digital is an Acromag® configured for discrete sensosrs (ch 17-32)

ATTRIBUTES

6 2 2 1 3 Object Class Acromag®Station-Reference

Superclass Acromag®-Station

Definition An Acromag®-Station-Reference is an Acromag® configured for reference sensors (ch 33 and 34)

ATTRIBUTES

6 2 2 2 Object Class Enraf®-Station

Superclass IO-Station

Definition An Enraf® is a particular brand of io-station used to connect to Enraf® Level Gauge sensors

Identifier station (unique numeric identifier)

ATTRIBUTES

Line-num	Domain Integer
ciu-addr	The address of the CIU the station is connected to Domain Integer
ENRAF-Type	Domain Level, Pressure-1 Pressure-3
Gauge-addr	Domain Integer
Line-num	Domain Integer
Polling-freq	The polling frequency of the station Domain {30,60]

6 2 2 2 1 Enraf Station Level

Superclass Enraf-Station

ATTRIBUTES

Alarm_status	Domain [f,]
Alarm_status_index	Domain Integer [0 5]
Alarm_status_mode	Type enraf®-alarm-status-parameter, normal low_alarm high_alarm block or freeze active, motor_limit_switch, alarm_data_error
Displacer-position	Domain [f]
Level_status	Domain [f,]
Level_status_index	Domain integer [0,X]
Level_status_mode	Type enraf®-level-status-parameter
Waste_level	Domain [f]

6 2 2 2 2 Enraf Station Pressure

Superclass Enraf-Station

ATTRIBUTES

Pressure_Reading	Floating point integer
Pressure_Status	Domain Symbol with five characters

6 2 2 3 Object Class Panalarm®-Station

Superclass IO-Station

Definition A Panalarm® Station is a particular brand of io-station

ATTRIBUTES

Cabinet-number	Domain integer
Number of Columns	Domain integer
Cabinet Label	A description of the panalarm cabinet
Obj-index	Type quantitative-parameter
Polling Frequency	Domain Integer
Number of Rows	Domain integer
Slots-inuse-Slot Mask	Domain Slot A and Slot C Domain Integer

6 2 2 4 Object Class Panalarm Station for 4x4x2AC

Superclass Panalarm Station
Definition This is an example of a 4 by 4 column panel where each panel has a slot A and slot C in use Others panels will follow same format where they can have up to four slots in use
Identifier Identifier
Attributes ch[01-04]-[01-04]-[a or c], is given by a discrete panalarm-channel There are 32 combinations

6 2 2 5 Object Class Westronics Station

Superclass IO-Station
Definition Particular brand of IO-Station

Attributes

Station-address initially is 1
Slot type initially is 1-analog through 5-analog
Register-count initially is 1-20
Polling frequency is initially 1-4
Line number is initially 0-7
Obj-index initially is given by a quantitative parameter

6 2 2 6 Object Class Westronics Station Slot (1-5)

Superclass Westronics Station

Definition One slot of a five slot Westronic IO-Station

Attributes pt010-pt029(twenty point with ten offset) Slot 1 Is initally given by a West channel parameter
pt030-pt49 Slot 2
pt050-pt69 Slot 3
pt070-pt89 Slot 4
pt090-pt109 Slot 5

6 2 3 Object Class IO-Channel

Superclass Quantitative-Parameter (G2® built-in)

Definition The io-channel is an attribute of an io-station It receives the raw readings from sensors via the GSI interface

ATTRIBUTES None

6 2 3 1 Object Class **ACR-Channel-Parameter**

Superclass IO-Channel

Definition The ACR-Channel-Parameter is an atribute of the ACROMAG-Station analog and the ACROMAG-Station-Reference

Attributes None

6 2 3 2 Object Class **Discrete-ACRO Channel**

Superclass IO-Channel

Definition The discrete-ACRO-channel-Parameter is an attribute of the ACRO-station-digital

Attributes None

6 2 3 3 Object Class **ENRAF-IO-Channel**

Superclass IO-channel

Definition **The ENRAF-IO-channel is an attribute of the ENRAF-station-level and** ENRAF-station-pressure

Attributes None

6 2 3 4 Object Class **Discrete-Panalarm-Channel**

Superclass **IO-channel**

Definition The discrete-panalarm-channel is an attribute of the panalarm-station

Attributes None

6 2 3 5 Object Class **West-Channel-Parameter**

Superclass IO-channel

Definition The west-channel-parameter is an attribute of the Westronics-station-slot[1 through 5]

Attributes None

6 2 3 6 Object Class **Discrete-ENRAF-Channel**

Superclass IO-channel

Definition The discrete-ENRAF-channel parameter is an attribute of the ENRAF-station-level

Attributes None

6 2 4 Object Class GSI®-Interface-From-TMACS-Definitions

Superclass GSI®-Interface (G2® built-in)

Definition The general parent of all interfaces used in TMACS

ATTRIBUTES Example of ENRAF-06, interface

GSI® connection configuration	Details of connection including the protocol, host-name and port-number e g tcp-ip host "localhost" port-number 22206 Type G2®-built-in
Remote process initialization string	Downloaded to bridge, and used to initialize the serial port (for line, baud, data bits, etc) ' -d digi_7 –s 2400 –b 7 –h 1 –j 1 –p O –r 2 –q N –n 300 –e N-v 10[3],60[30] 600[100],3600[120]-xY' Type G2®-built-in

start-command	'startsrv bat@"EnrafDriver 22206@""
stop-command	"stopsrv bat@"EnrafDriver 22206@""

6 2 4 1 Object Class Acromag®-Interface

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 4 2 Object Class Enraf®-Interface

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 4 3 Object Class Panalarm®-Interface

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 4 4 Object Class SACS-Interface (SQL/ODBC)

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 4 5 Object Class Printer-Interface

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 4 6 Object Class Westronics-Interface

Superclass GSI®-Interface-From-TMACS-Definitions

6 2 5 Object Class Alarm-Message

Superclass Message (G2® built-in)

Definition This is a general alarm, which is subclassed into sensor and communication alarms

6 2 5 1 Object Class Sensor-Alarm-Message

Superclass Alarm-Message

Definition An alarm indicates a special-condition to an operator so that they can take an appropriate action It is triggered by an io-point reading that either exceeds a specified limit for a continuous io-point, or for a discrete io-point by a reading whose value is configured to be an alarm state

6 2 5 1 1 Object Class Sensor-Range-Alarm-Message

Superclass Sensor-Alarm-Message

Definition A range-alarm is an alarm caused by exceeding low or high alarm limits
ts

6 2 5 1 2 Object Class Sensor-ROC-Alarm-Message

Superclass Sensor-Alarm-Message

Definition A ROC-alarm is an alarm caused by rate-of-change alarm limits

6 2 5 2 Object Class Communications-Alarm-Message

Superclass Alarm-Message

Definition Communication alarms indicate problems with communication between the io-station and the sensors

6 2 5 2 1 Object Class **ACROMAG-Alarm-Message**

Superclass Communications-Alarm-Message

Definition Prints WHITE message when GSI status changes

6 2 5 2 2 Object Class **Station-Error-Message**

Superclass Communicaitons-Alarm-Message

Definition Prints WHITE message when station status changes

6 2 5 2 3 Object Class **Module-Error-Message**

Superclass Communications-Alarm-Message

Definition Prints WHITE message that sensor reading is out of range

6 2 5 2 4 Object Class **Panalm-Alarm-Message**

Superclass Communicaitons-Alarm-Message

Definition Prints WHITE message when communications error initiated from Panalarm device driver

6 2 5 2 5 Object Class **ENRAF-Message**

Superclass Communications-Alarm-Message

Definition Prints WHITE message when communications error initiated from ENRAF device driver

6 2 5 2 6 Object Class **Westronics-Alarm-Message**

Superclass Communications-Alarm-Message

Definition Currently not used

6 2 5 3 Object Class **Processing-Status-Message**

Superclass Alarm-Message

Definition Prints BLUE message when anyone of the five processing flags changes state

6 2 6 Object Class Display-Icon

Superclass Object [G2 Built]

Definition Display-Icon includes farm-icons, tank-icons, and external-alarm-group-icons

6 2 6 1 Object Class Tank-Icon

Superclass Display-Icon

Definition A tank icon is the visual representation (plan view) of the tank alarm group status that the user can see on the screen

6 2 6 1 1 Object Class **Double-Shell-Tank-Icon**

Superclass Tank-Icon

Definition A double-shell tank icon is the visual representation (plan view) used to represent the aggregate alarm-group-status for all sensors for a double-shell tank

6 2 6 1 2 Object Class **Single-Shell-Tank-Icon**

Superclass Tank-Icon

Definition A single-shell tank icon is the visual representation (plan view) used to represent the aggregate alarm-group-status for all sensors for a single-shell tank

6 2 6 1 2 1 Object Class **Small-Single-Shell-Tank-Icon**

Superclass Single-Shell-Tank-Icon

Definition A Small-Single-Shell-Tank-Icon is the visual representation (plan view) used to represent the aggregate alarm-group status for all sensors for a Small-Single-Shell-Tank

6 2 6 2 Object Class External-Alarm-Group-Icon

Superclass Display-Icon

Definition An alarm group that is not associated with the 200 Area Tank Farms

6 2 6 2 1 Object Class **K-Basin-Icon**

Superclass External-Alarm-Group-Icon

Definition An icon used to represent the aggregate alarm-group-status of the K-Basins monitored system

6 2 6 3 Object Class Farm-Icon

Superclass Display-Icon

Definition **A Farm-Icon represents alarm groups within the tank farm but not directly associated to a tank**

6 2 6 3 1 Object Class **Sub-Farm-Icon**

Superclass Farm Icon

Definition A Sub-Farm-Icon is a smaller set of alarm groups within the tank farm but not directly associated to a tank

6 2 6 3 1 1 Object Class **Sub-X by –Y-Panel-Icon**

Superclass Sub-Farm-Icon

Definition A sub-X-by-Y-Panel-Icon represents a X-by-Y portion of an annunciator panel board

6 2 6 4 Object Class **Between-Farm-Icon**

Superclass Display-Icon

Definition Display alarm groups that are between tank farms This alarm group maybe an intermediate layer of alarms

6 2 6 4 1 Object Class **Sub-Between-Farm-Icon**

Superclass Between-Farm-Icon

Definition Display alarm groups that are between tank farms This alarm group maybe an intermediate layer of alarms

6 2 7 Object Class **Alarm-Group**

Superclass Display-Icon

Defintion Any alarm group

6 2 7 1 Object Class **Tank**

Superclass Alarm-Group

Defintion Intermediate alarm group

6 2 7 1 1 Object Class **Single-Shell-Tank-Alarm-Group**

Superclass Tank-Alarm-Group

Definition An alarm group associated with a single shell tank

6 2 7 1 2 Object Class **Double-Shell-Tank-Alarm-Group**

Superclass Tank-Alarm-Group

Defintion An alarm group associated with a single shell tank

6 2 7 2 Object Class Monitored-System
Superclass Alarm-Group

Defintion An intermediate alarm group

6 2 7 2 1 Object Class Monitored-System-4-by-4-Panel-Alarm-Group
Superclass Monitored-System-Alarm-Group

Defintion An alarm group associated with a 4-by-4 panel

6 2 7 2 2 Object Class Monintored-System-4-by-8-Panel-Alarm-Group

Superclass Monitored-System-Alarm-Group

Defintion An alarm group associated with a 4-by-8-panel

6 2 7 2 3 Object Class K-Basin-Storage-Tank-Alarm-Group
Superclass Monitored-System-Alarm-Group

Defintion An alarm group associated with a K-Basin-Storage-Tank

6 2 7 2 4 Object Class Monitored-System-5-by-5-Panel-Alarm-Group
Superclass Monitored-System-Alarm-Group

Defintion An alarm group associated with a 5-by-5-Panel

6 2 7 2 5 Object Class Monitored-System-2-by-2-Panel-Alarm-Group
Superclass Monitored-System-Alarm-Group

Defintion An alarm group associated with a 2-by-2-Panel

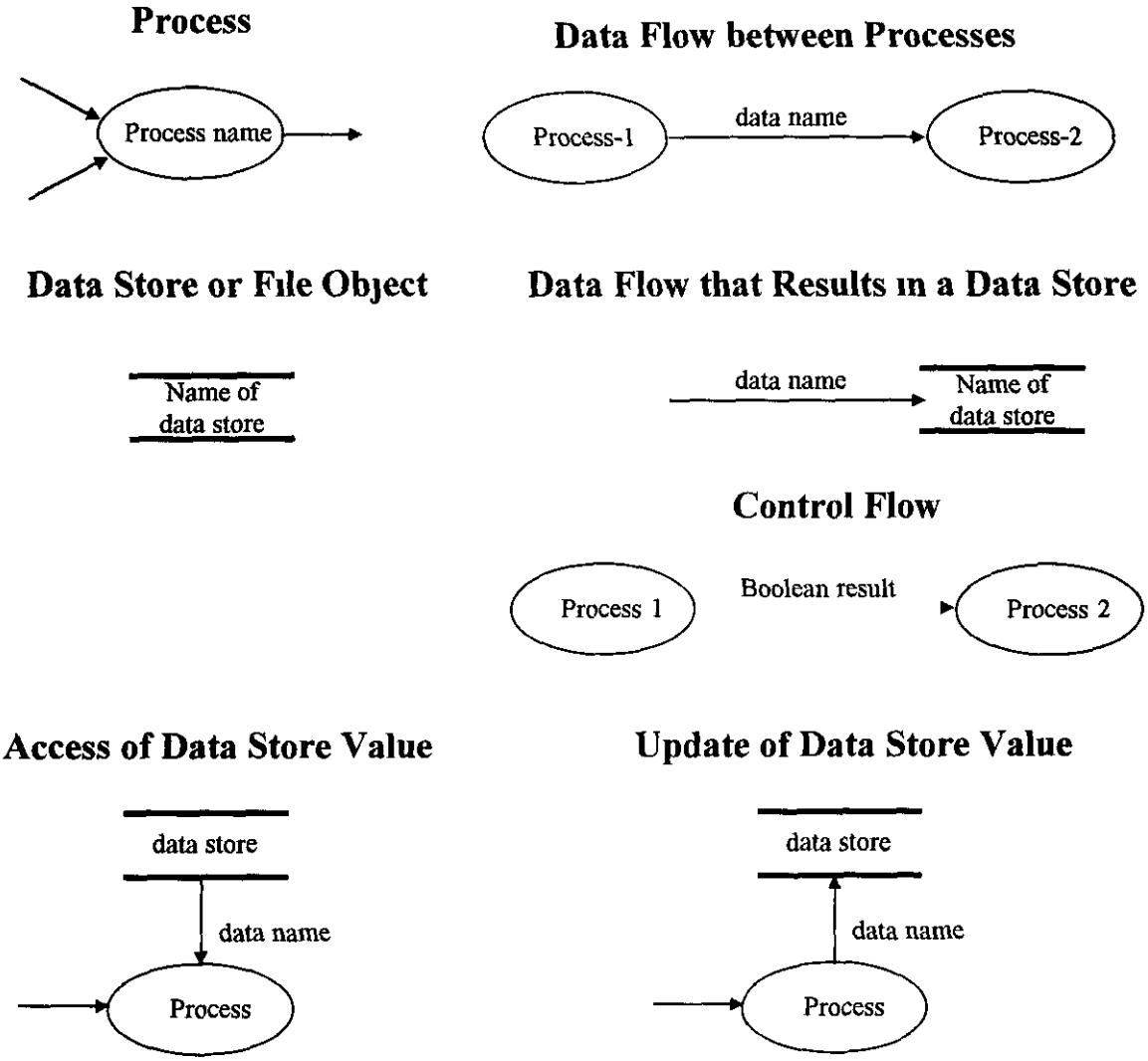
7 AS-BUILT FUNCTIONAL MODELS (DATA FLOW DIAGRAMS)

These models show the processes and data involved to accomplish the various use-cases (functional tasks) TMACS was largely built with versions of G2® that only had object-oriented data structures and no methods Thus the objects turn out to be little more than fancy data-stores Given this scenario, functional models (data-flow-diagrams) are appropriate to show the as-built design of TMACS **Note The data-flow-diagrams assumes that point process, alarm processing and logging are enabled**

7 1 Functional Model Notation

The notation used in the sections to follows is described in the Figure 19 Functional Model (Data Flow Diagram) Notation

Figure 19 Functional Model (Data Flow Diagram) Notation



7 2 Continuous Point Processing Data Flow

The diagrams in this sections shows the data flow for continuous point processing

Figure 20 DFD - Point Processing Continuous-io-point, Instrument-in-range=TRUE

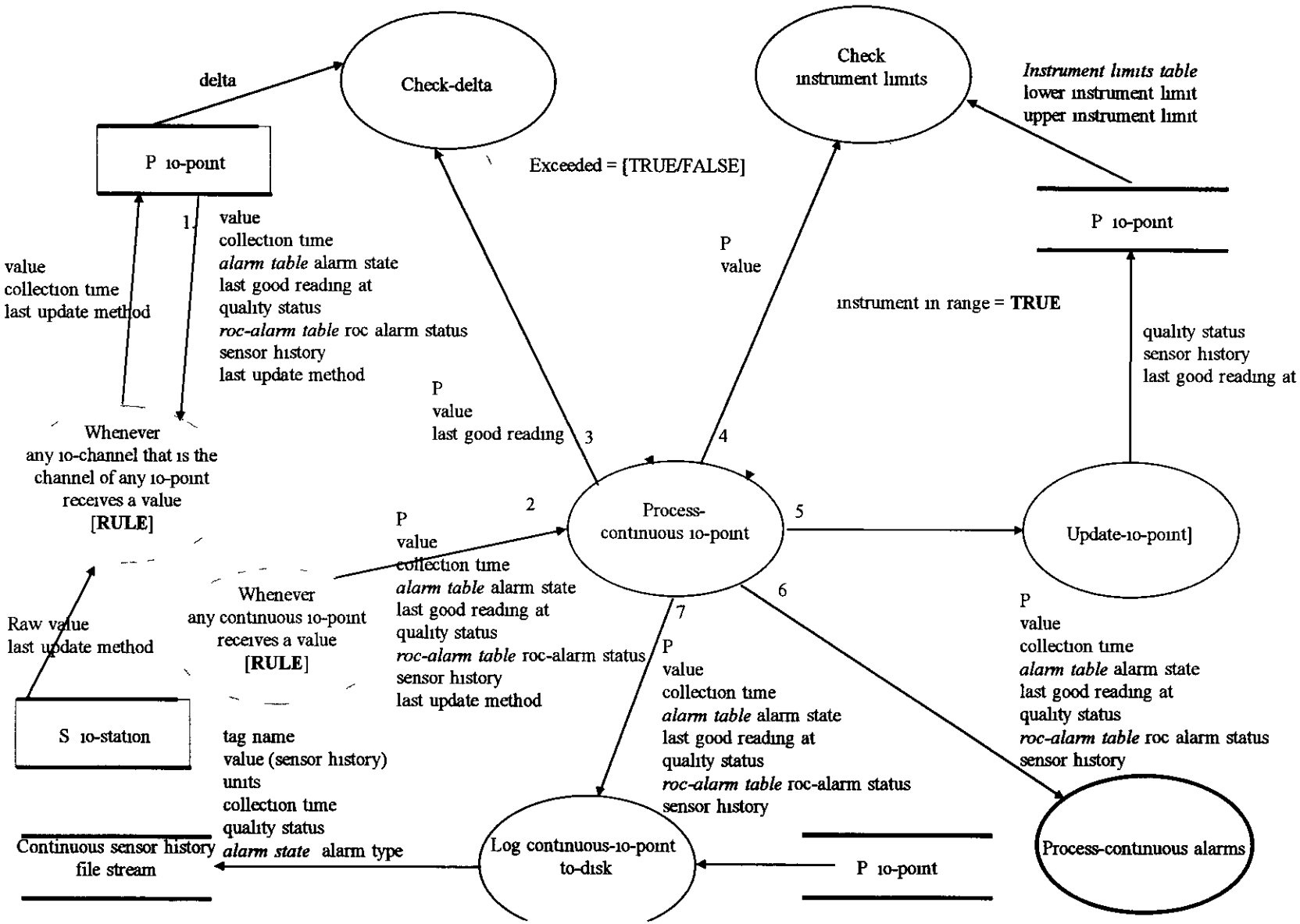


Figure 21 DFD - Point Processing Continuous-io-point, Instrument-in-range=FALSE

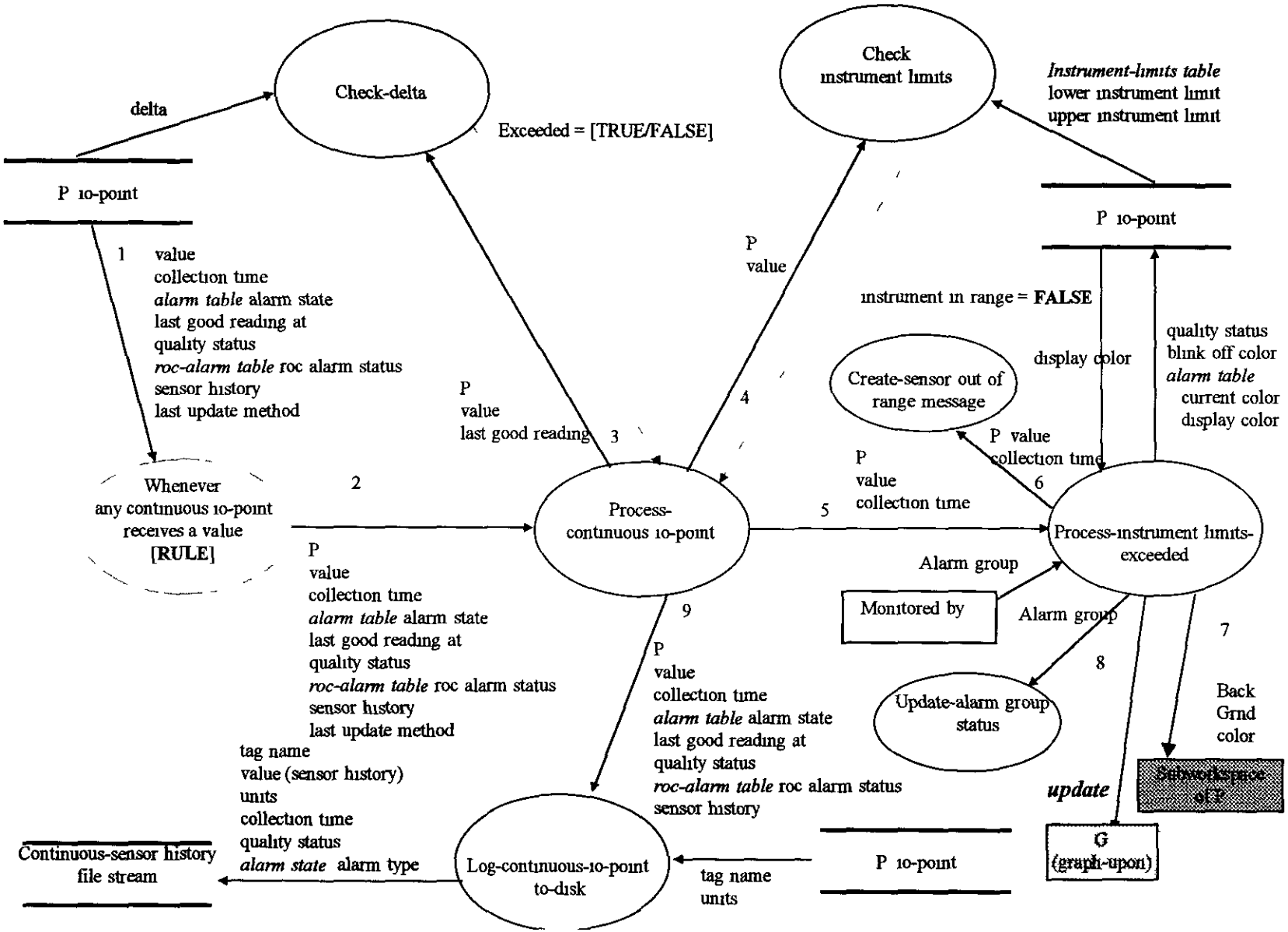


Figure 22 DFD - Point Processing, Continuous-io-point Alarm Processing 1 of 3

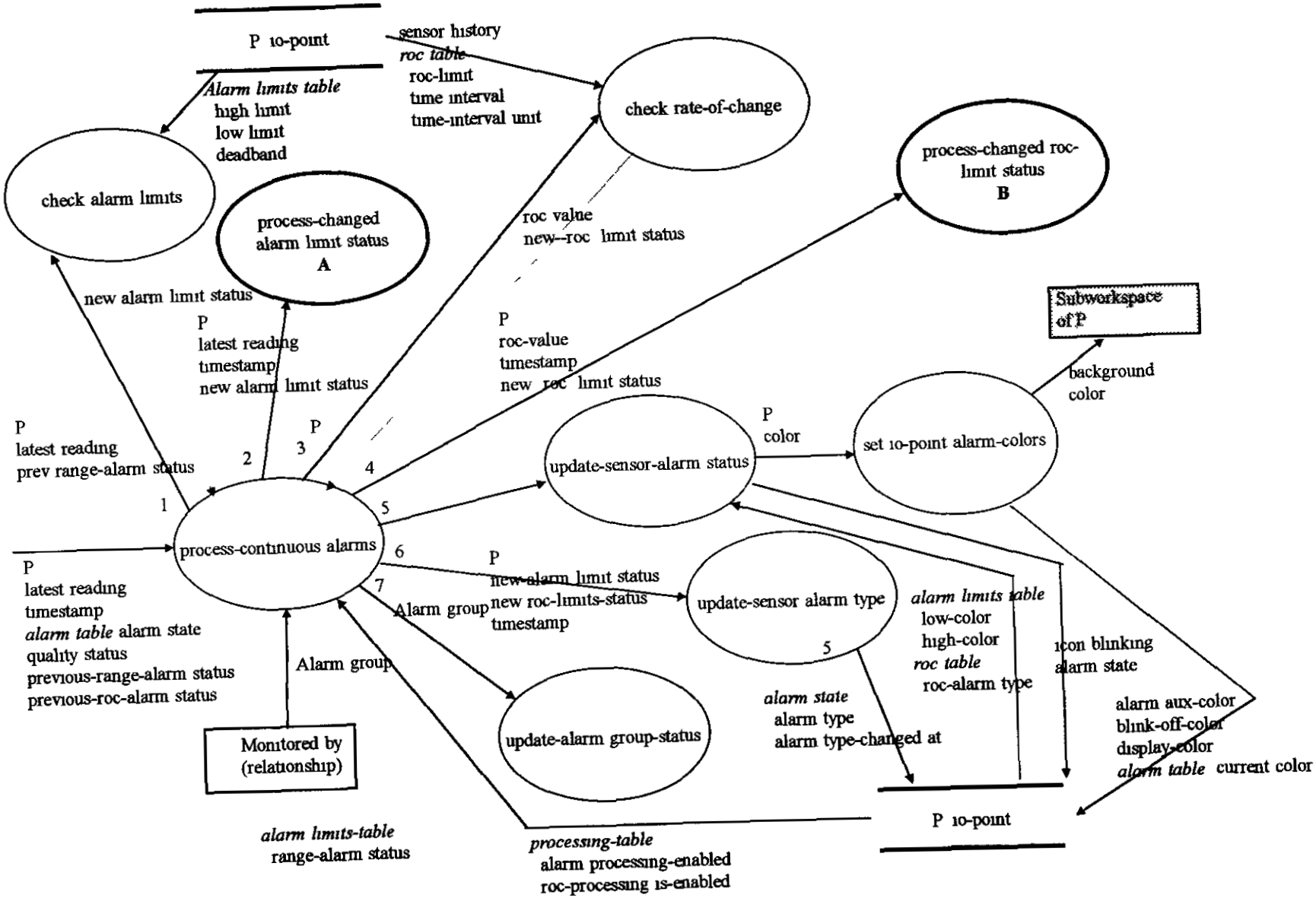
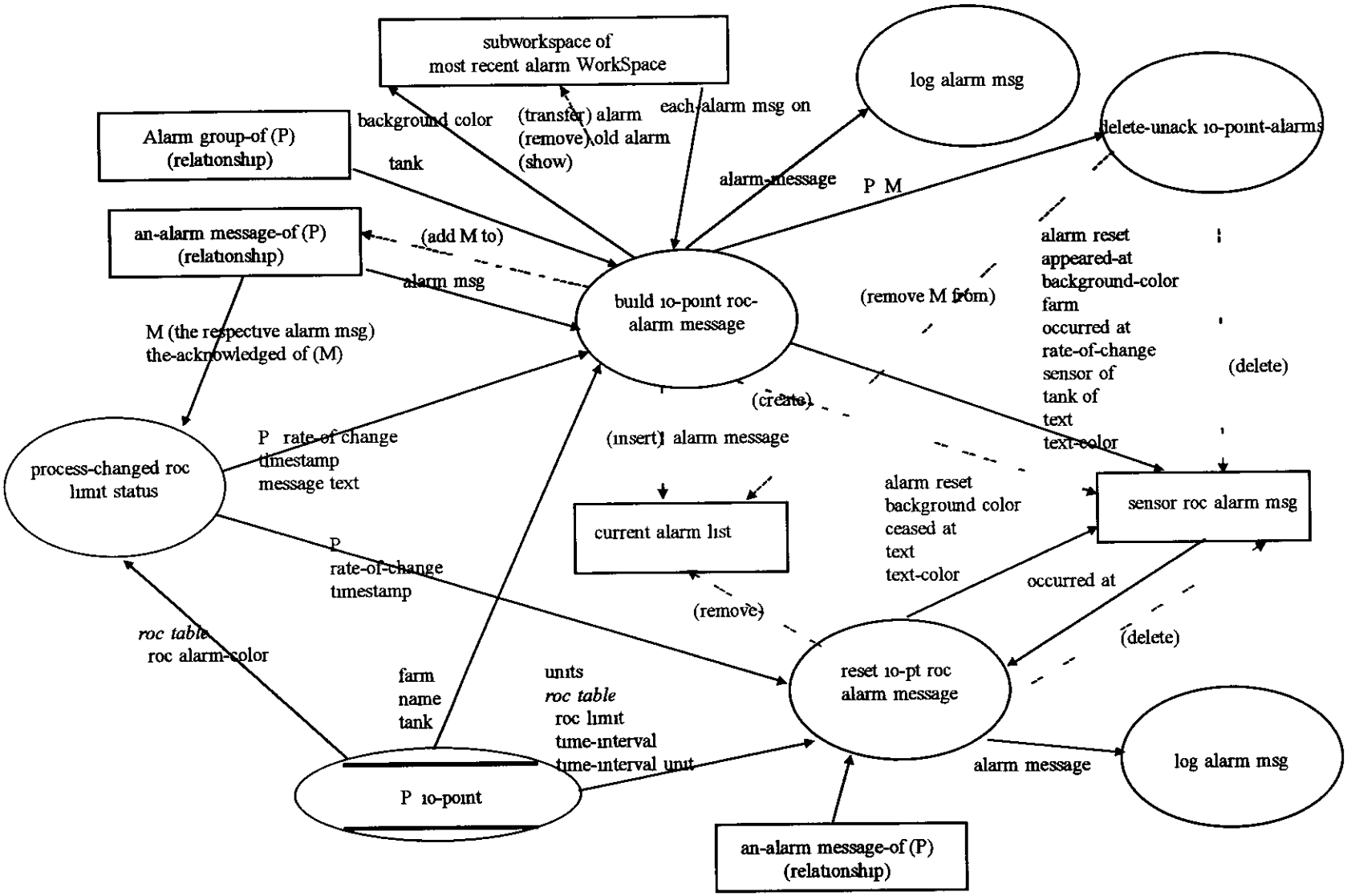


Figure 23 DFD - Point Processing, Continuous-10-point Alarm Processing 2 of 3



7 3 Discrete Point Processing Data Flow

The diagrams in this sections shows the data flow for continuous point processing

Figure 24 DFD - Point Processing, Continuous-io-point Alarm Processing 3 of 3

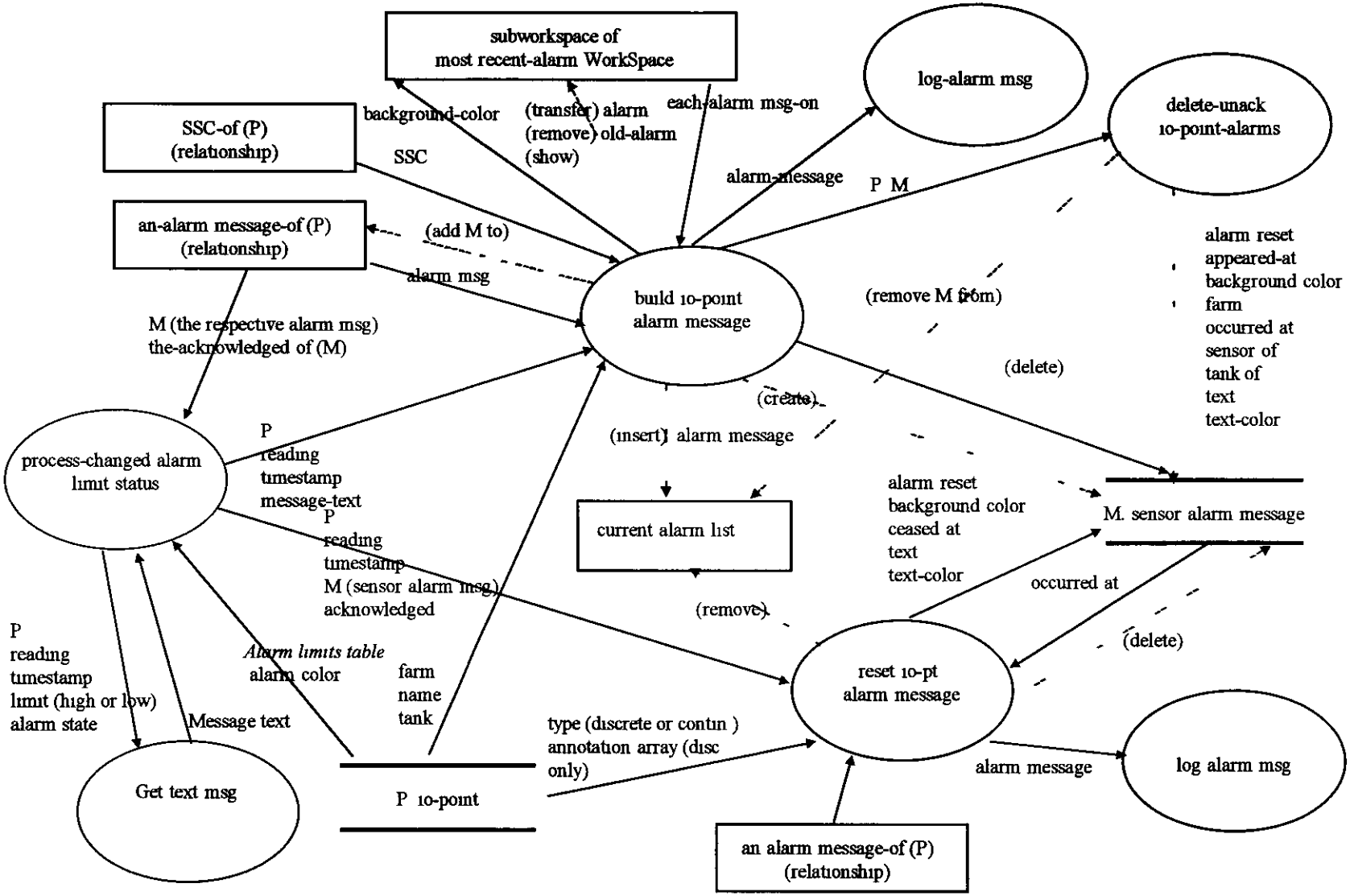


Figure 25 DFD - Point Processing, Discrete-io-point.

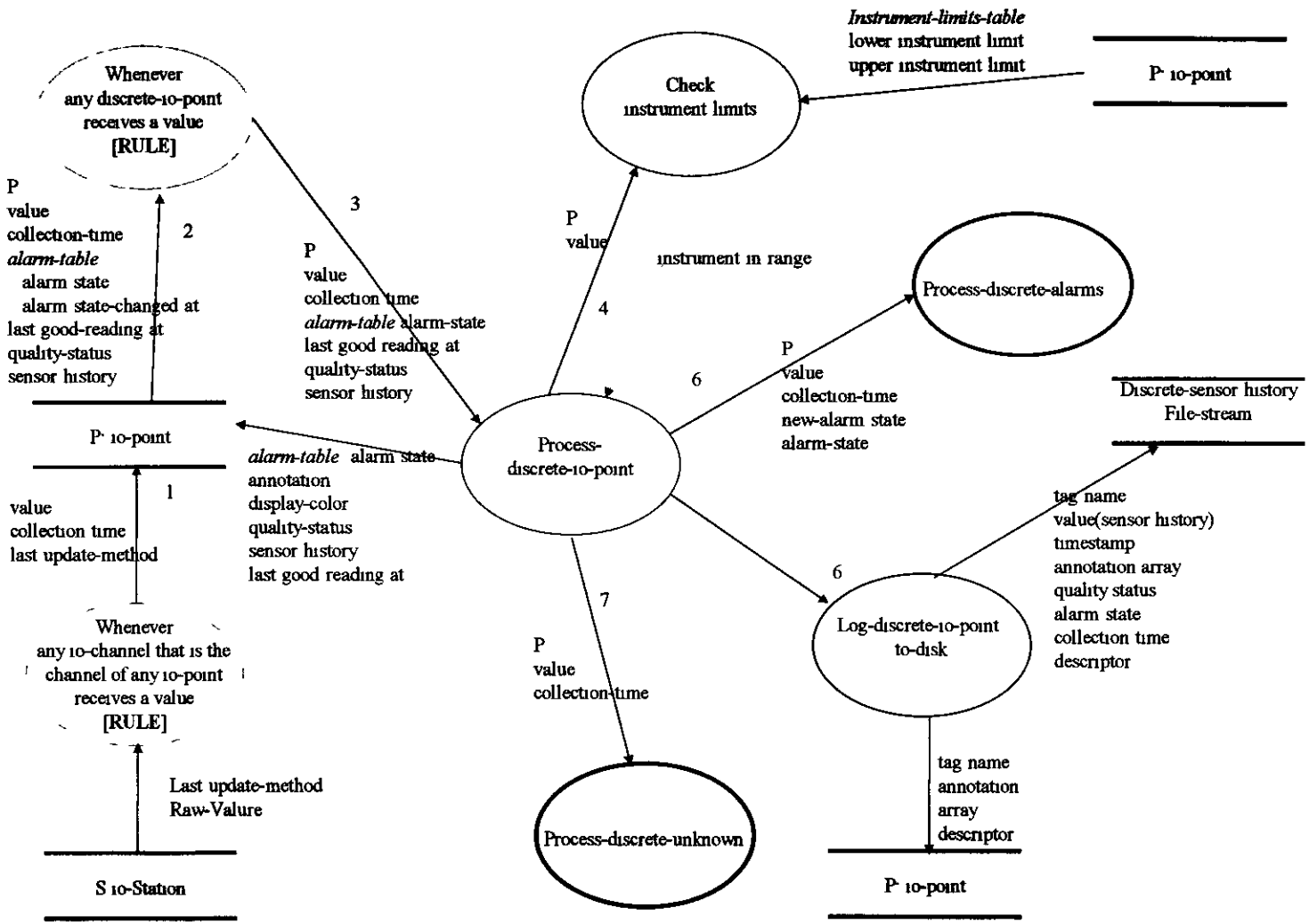


Figure 26 DFD – Process Discrete Alarms

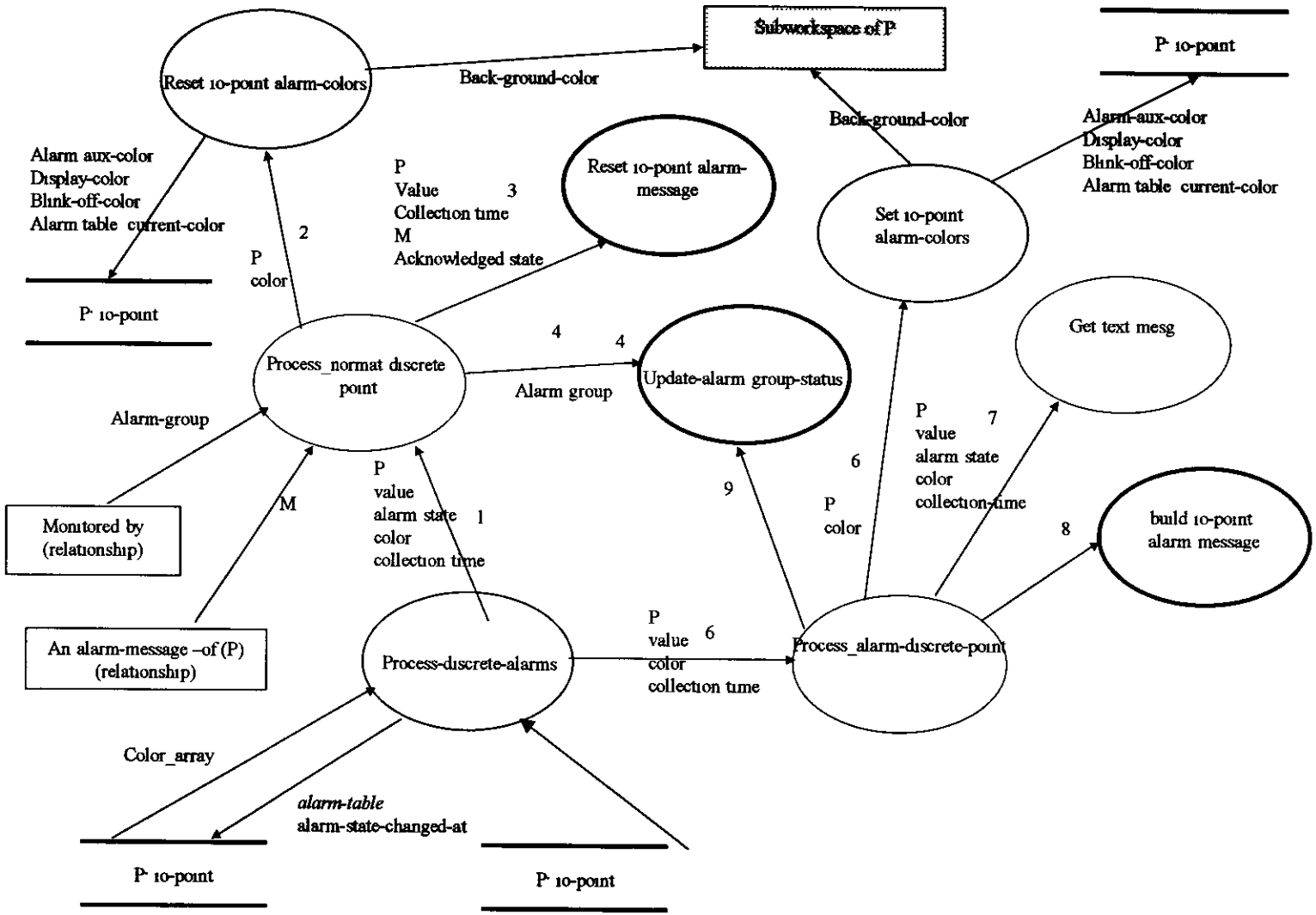


Figure 27 DFD - Reset-IO-Pt-Alarm-Message, Build-IO-Pt-Alarm-Message

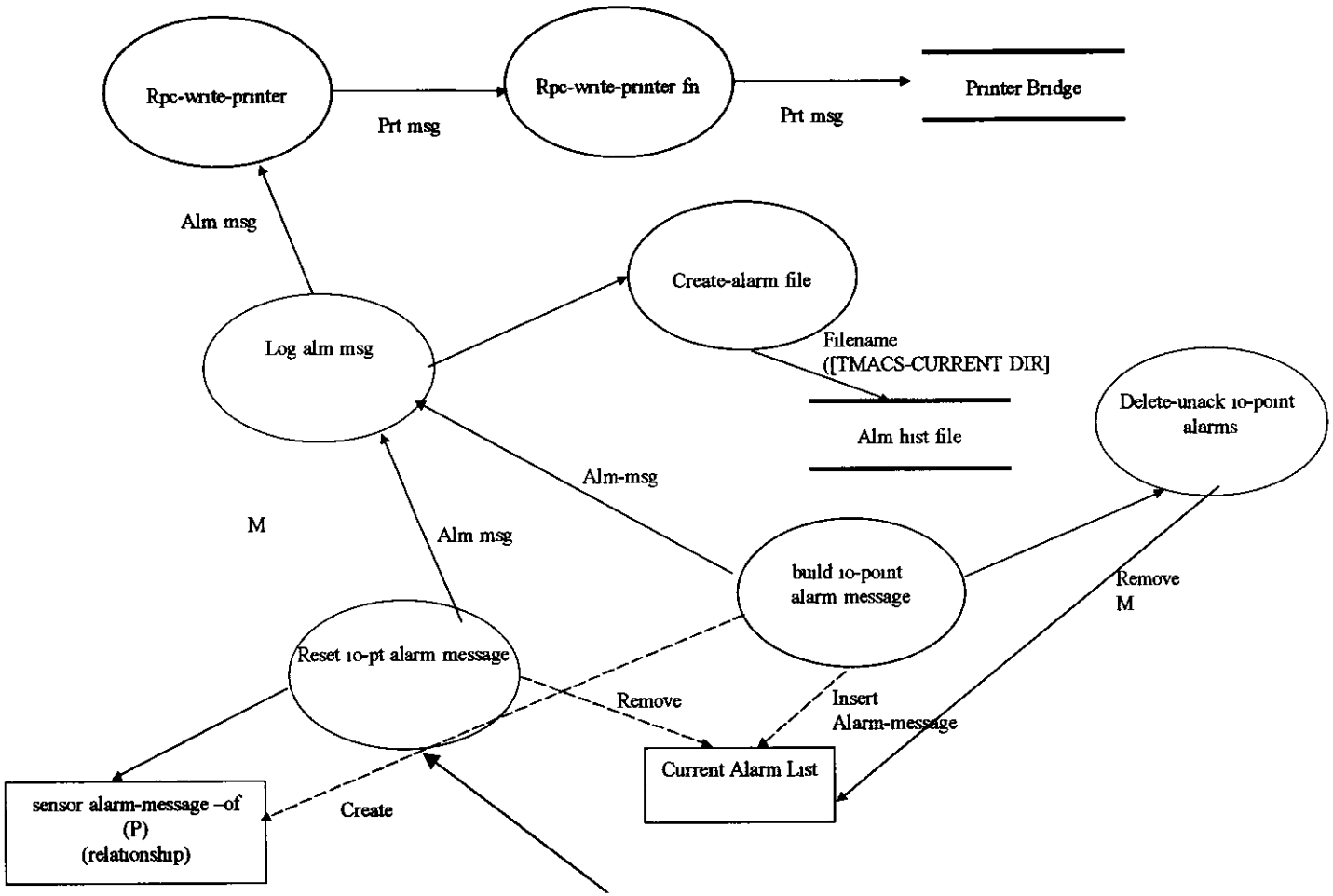


Figure 28 DFD -Update-Structurre-System-Component-Status

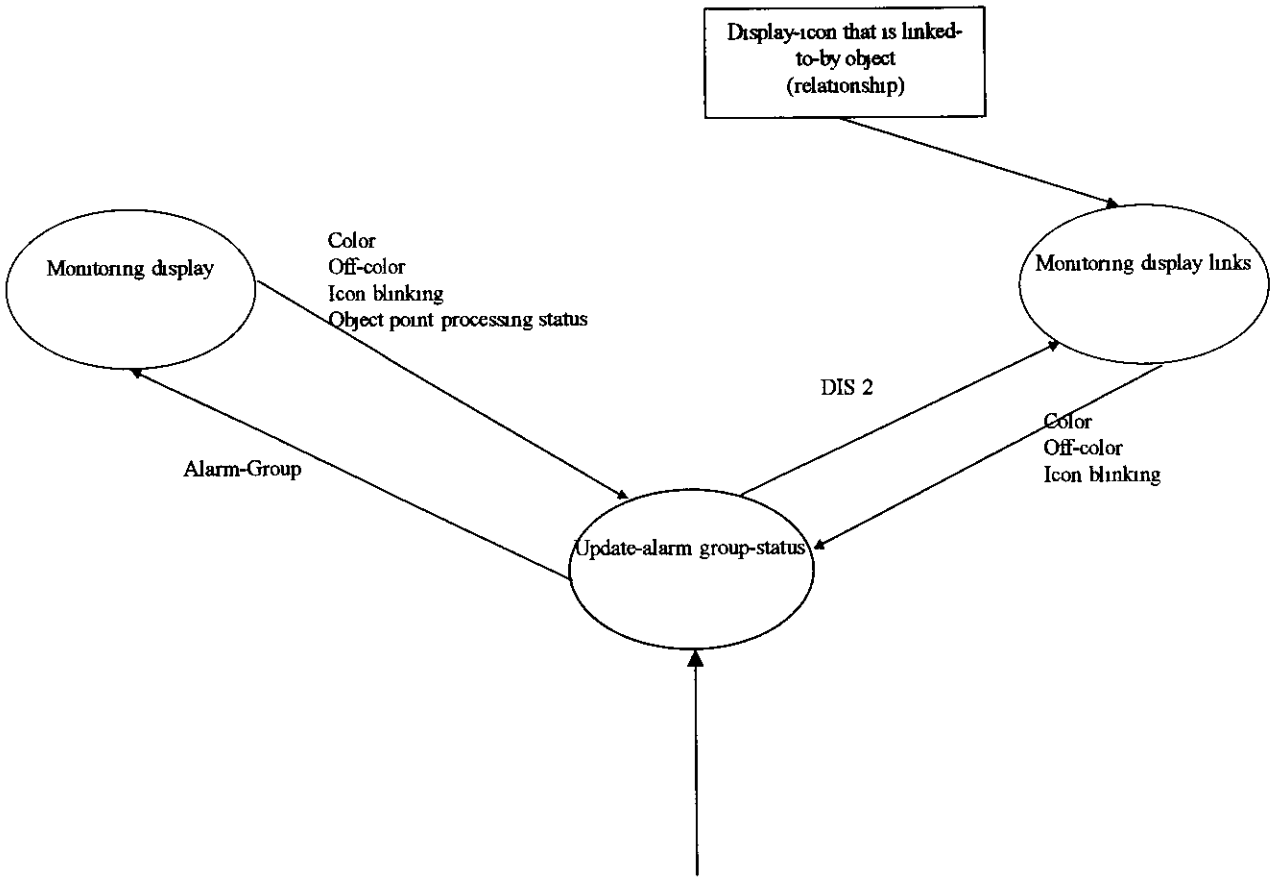
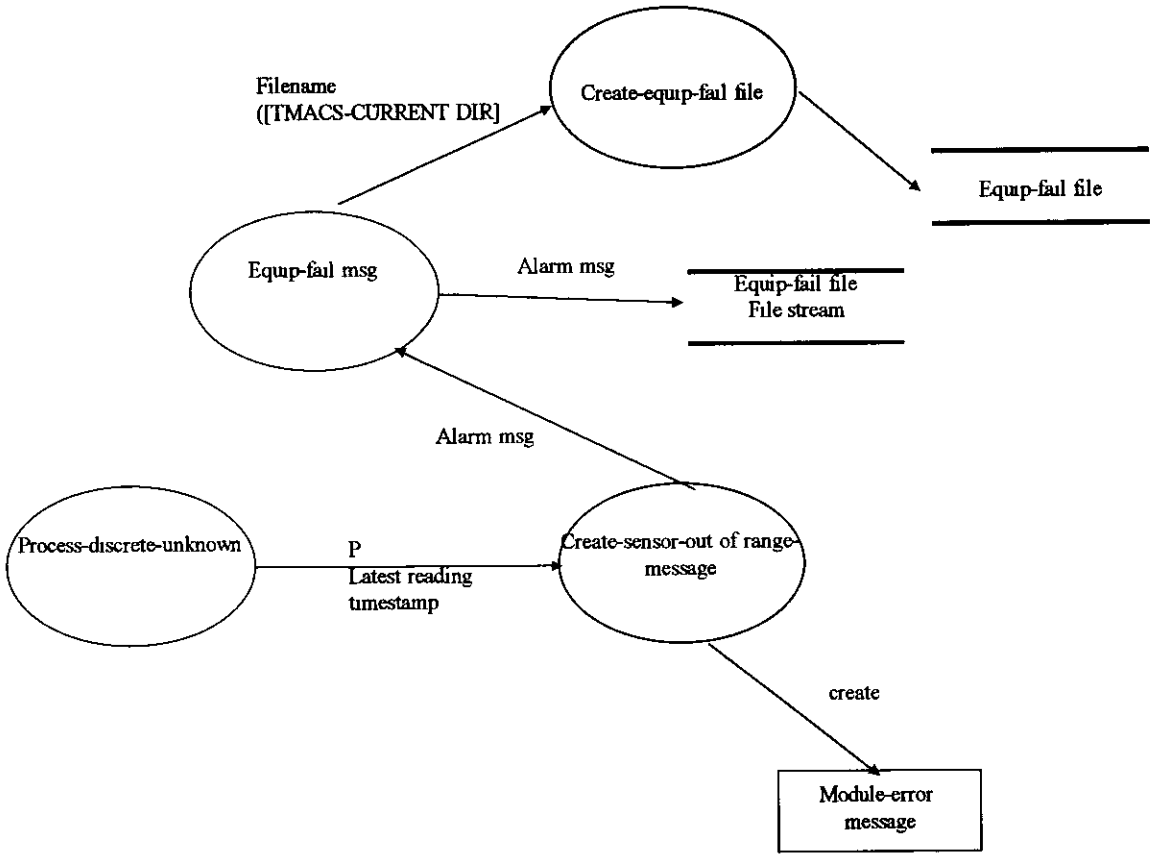


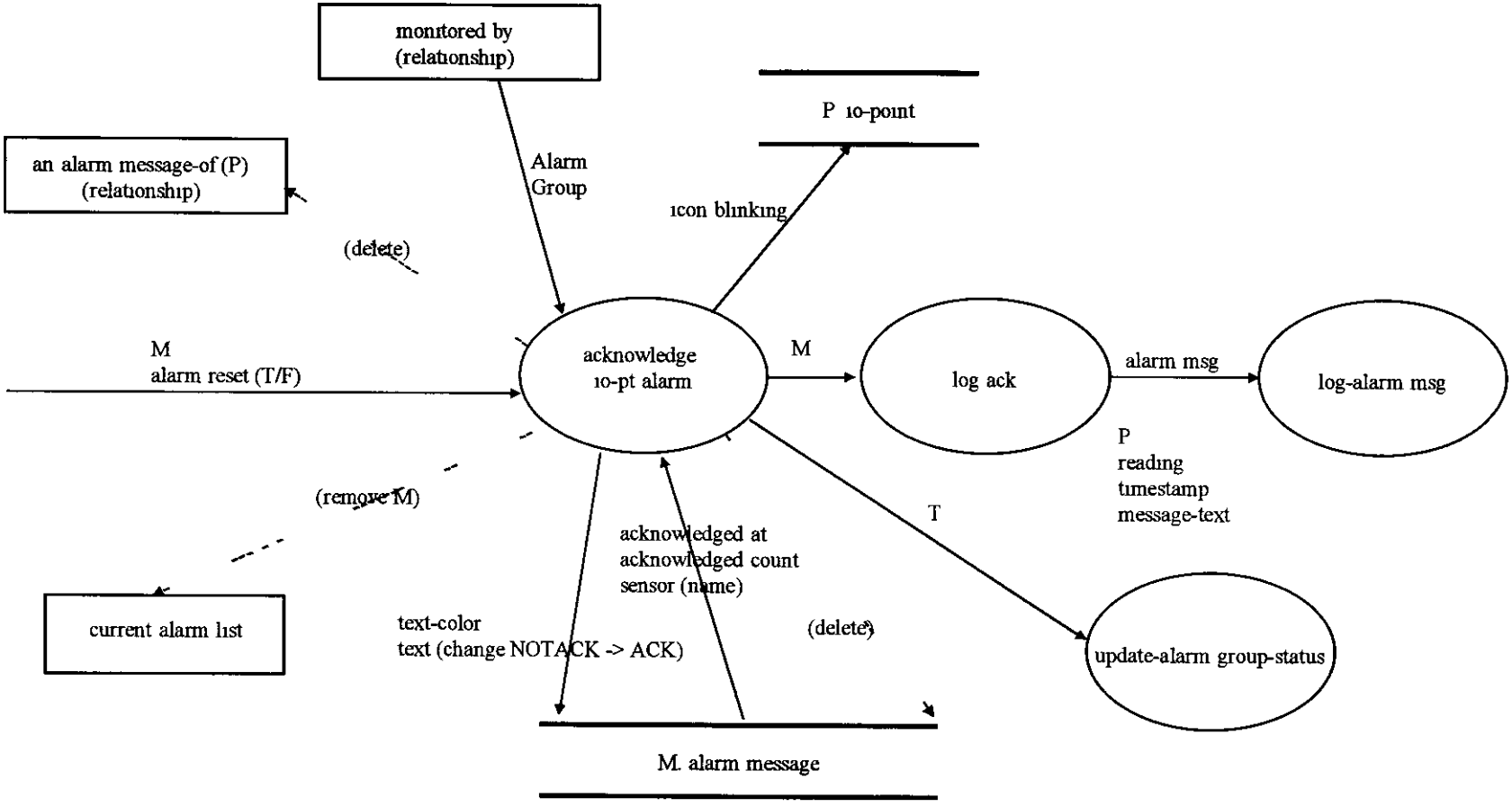
Figure 29 DFD - Process DiscreteUnknown



7 4 Generic Data Flow

This section contains data flow diagrams that are generic to both discrete and continuous io-processing

Figure 30 DFD - Acknowledge Alarm Messages



8 TMACS FILES FORMATS

This section provides the formats for TMACS data files They are ASCII flat files created automatically by TMACS Files for the current day are in located on the production system in the directory "f \BackedUp\TMACSData\Current" Files for yesterday and before are located on the production server in the directory"f \BackedUp\TMACSData\History "

The files are

(1) Alarm History File - used for recording Sensor Alarm events There are 6 different message formats since each alarm, acknowledge, and reset is reported differently for continuous and discrete sensors This file is created in the current directory when alarms occur on TMACS and moved to history at midnight This file will have the date and time stamp of the time the first alarm for that day occurred Described in Sections 8 1 1through 8 1 6

(2) Equipment Failure History File - used for recording I/O device equipment failure events This may include field I/O communication failures as well as sensor readings that exceed the limits of the instrument recording the reading There are several message formats since there are several devices that TMACS communicates with - Acromag I/O Station, Enraf Level Gauge Panalarm Alarm Cabinet, etc This file is created in the current directory when equipment failures occur on TMACS This file will have the date and time stamp of the time the first equipment failure for that day occurred Described in Sections 8 2 1through 8 2 5

(3) Discrete Sensor History File – used for recording the activity of every TMACS discrete sensor Every time a change in any discrete sensor is occurs this value is recorded in this file This file is used in TMACS data recovery operations This file is generated automatically every day at 10 seconds past midnight It may be empty if no discrete sensor values have changed for that day Described in Section 8 3

(4) Continuous Sensor History File - used for recording the activity of every TMACS continuous sensor Every time a reading is taken from the field equipment a determination is made whether or not to record a value for the sensor based on whether it has changed more than delta units from the last recorded reading This file is used in TMACS data recovery operations This file is generated automatically every day at 10 seconds past midnight After creation of the file all sensor values are logged to it automatically whether they have changed outside their delta band or not Described in Section 8 4

(5) Double-Shell Tank Daily File - a snapshot of the current readings for all double-shell tank sensors These data are recorded at approximately 4 a m every day Described in Section 8 5

(6) Single-Shell Tank Daily File - a snapshot of the current readings for all single-shell tank sensors These data are recorded at approximately 4 a m every day Described in Section8 5

(7) Test Tank History File - used for recording the activity of certain TMACS continuous sensors It has the same format as the Continuous Sensor History file (described in Appendix 4) but contains only the output from certain sensors in AN farm This file is created automatically every day at 10 seconds past midnight No readings are written to this file at this time

Data files (1) through (7) are sent to another computer (tfs9) each day for processing The Alarm History, Equipment Failure and Discrete Sensor History files are stored without processing The Continuous Sensor History and Test Tank History files are processed into the RAWDATA database The two Tank Daily files (5 and 6) are processed into SACS Readings with recognized sensor tag names are inserted into the SACS production database

(8) Discrete Shutdown File – used to record the last reading for every TMACS discrete sensor This file is used in TMACS data recovery operations This file is generated automatically every time an orderly shutdown is performed on TMACS Described in Section 0

(9) Continuous Shutdown File - used to record the last reading for every TMACS continuous sensor This file is used in TMACS data recovery operations This file is generated automatically every time an orderly shutdown is performed on TMACS Described in Section 0

(10) Performance Data File - a rule has been added to the TMACS knowledge base to record performance data every hour The data file is created in the TMACS "CURRENT" file and moved at midnight to the "HISTORY" directory It is not transferred to tfs9 Described in Section 8 8

8 1 Alarm History File

8 1 1 Continuous Sensor Alarm Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
ALARM <Alarm-Desc> <Object-Name> **Reading** <Reading> <Units> **Limit** <Limit>
<Ack-Timestamp><State-of-Acknowledgement>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31) MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 – 12) mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Alarm-desc	A designation as to what type of alarm has been detected E g , **HIGH, LOW**
Object-name	A designation of what object the alarm occurred at The format is typically **CC-NNN**, where CC = one or two letter tank farm designation and NNN = 3 digit tank number designation E g **BY-101**
Reading	The current value for the designated sensor It can be either an integer or floating point number
Units	The engineering units the sensor reading is expressed in This can be Fahrenheit for a temperature sensor, Inches of Water Gauge for a pressure sensor, etc E g , **deg F**, or **INWG**
Limit	The alarm limit that was exceeded The format is in the same units as the reading E g 70 0
Ack-Timestamp	The date and time the message was acknowledged by the TMACS user The format is the same as the message timestamp and this only appears if alarm acknowledged
State-of-Acknowledgement	The current acknowledgement state of this alarm The alarm can be **NOT ACKNOWLEDGED** or **ACKNOWLEDGED**

8 1 2 Continuous Sensor Alarm Reset Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
ALARM <Alarm-Desc> <Object-Name> **Reading** <Reading> <Units> **Limit** <Limit>
<State-of-Acknowledgement>
<Reset-Timestamp> **ALARM RESET Reading** <Reset-Reading> <Reset-Units>
Duration <Duration>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >** where HH is the hour (0 – 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Alarm-desc	A designation as to what type of alarm has been detected E g **HIGH, LOW**
Object-name	A designation of what object the alarm occurred at The format is typically **CC-NNN**, where CC = one or two letter tank farm designation and NNN = 3 digit tank number designation E g **BY-101**
Reading	The value for the designated sensor when alarm was generated It can be either an integer or floating point number
Units	The engineering units the sensor reading is expressed in This can be Fahrenheit for a temperature sensor, Inches of Water Gauge for a pressure sensor, etc E g , **deg F**, or **INWG**
Limit	The alarm limit that was exceeded It can be either an integer or a floating point number The units are the same as the Reading
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged, then this value is NOT ACKNOWLEDGED If the alarm was acknowledged, the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

Field Descriptor	Definition
Reset-Timestamp	The date and time the sensor reading reset (changed back to normal range) The format is the same as the message timestamp
Reset-Reading	The value for the designated sensor when the alarm reset It can be either an integer or floating point number
Reset-Units	The engineering units the sensor reading is expressed in
Duration	The length of time the sensor was in alarm That is length of time between alarm and alarm reset events

8 1 3 Continuous Sensor Alarm Acknowledge Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
ALARM <Alarm-Desc> <Object-Name> **Reading** <Reading> <Units> **Limit** <Limit>
<Ack-Timestamp> **ALARM ACKNOWLEDGED**
<Reset-Timestamp> **ALARM RESET Reading** <Reset-Reading> <Reset-Units>
Duration <Duration>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 – 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Alarm-desc	A designation as to what type of alarm has been detected E g , **HIGH, LOW**
Object-name	A designation of what object the alarm occurred at The format is typically **CC-NNN**, where CC = one or two letter tank farm designation and NNN = 3 digit tank number designation E g , **BY-101**
Reading	The value for the designated sensor when alarm was generated It can be either an integer or floating point number
Units	The engineering units for the sensor reading
Limit	The alarm limit that was exceeded It can be either an integer or a floating point number The units are the same as the Reading
Ack-Timestamp	The date and time the alarm was acknowledged
The data elements below are present only if the alarm has reset before the operator acknowledged it	
Reset-Timestamp	The date and time the sensor reading reset (changed back to normal range) The format is the same as the message timestamp
Reset-Reading	The value for the designated sensor when the alarm reset It can be either an integer or floating point number

Field Descriptor	Definition
Reset-Units	The engineering units the sensor reading is expressed in
Duration	The length of time the sensor was in alarm That is length of time between alarm and alarm reset events

8 1 4 Discrete Sensor Alarm Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
<Sensor-Desc> <Discrete-state>
<Ack-Timestamp> <State-of-Acknowledgement>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 - 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Sensor-desc	A description of the sensor E g , **SY-101, SHMS-J, HIGH HYDROGEN**
Discrete-state	The state of the discrete sensor E g , **OPEN, CLOSED**
Ack-Timestamp	The date and time the message was acknowledged by the TMACS user The format of the data is the same as the Message Timestamp described elsewhere This field only appears if the message has been acknowledged
State-of-Acknowledgement	The current acknowledgement state of this alarm The alarm can be **NOT ACKNOWLEDGED** or **ACKNOWLEDGED**

8 1 5 Discrete Sensor Alarm Reset Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
<Sensor-Desc> <Discrete-state>
<State-of-Acknowledgement>
<Reset-Timestamp> **ALARM RESET** <Discrete-Reset-State>
Duration <Duration>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31) MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 - 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Sensor-desc	A description of the sensor (an attribute of all TMACS I/O points) E g , **SY-101, SHMS-J, HIGH HYDROGEN**
Discrete-state	The state of the discrete sensor E g ***OPEN CLOSED***
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged then this value is NOT ACKNOWLEDGED If the alarm was acknowledged, the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED
Reset-Timestamp	The date and time the sensor reading reset (changed back to non-alarm value) The format is the same as the message timestamp
Discrete-Reset-State	The state of the discrete sensor when the alarm reset
Duration	The length of time the sensor was in alarm That is length of time between alarm and alarm reset events

8 1 6 Discrete Sensor Alarm Acknowledge Message

File Name Format *almhst_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-Timestamp> **TAG NAME** <Tag-Name>
<Sensor-Desc> <Discrete-state>
<Ack-Timestamp> **ALARM ACKNOWLEDGED**
<Reset-Timestamp> **ALARM RESET** <Discrete-Reset-State>
Duration <Duration>

Field Descriptor	Definition
Message-Timestamp	The date and time the original message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 - 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g **12 Jan 95 11 43 23 a m**
Tag-Name	The sensor reporting activity The format is described in the TMACS I/O Termination Point Listing supporting document
Sensor-desc	A description of the sensor E g , **SY-101, SHMS-J, HIGH HYDROGEN**
Discrete-state	The state of the discrete sensor E g , **OPEN CLOSED**
Ack-Timestamp	The date and time the alarm was acknowledged by the operator
The data elements below are present only if the alarm has reset before the operator acknowledged it	
Reset-Timestamp	The date and time the sensor reading reset (changed back to non-alarm value) The format is the same as the message timestamp
Discrete-Reset-State	The state of the discrete sensor when the alarm reset
Duration	The length of time the sensor was in alarm That is length of time between alarm and alarm reset events

8 2 Equipment Fail File

8 2 1 Acromag Equipment Fail Message

File Name Format *equip_fail_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

```
==========================================================
<Message-TimeStamp>        DATA ACQUISITION EQUIPMENT
  Port <Port-Name> Station-ID  <Station #> Channel  <Channel #>
  <Message>
<State-of-Acknowledgement>
==========================================================
```

Field Descriptor	Definition
Message-TimeStamp	The date and time the message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >** where HH is the hour (0 - 12) mm are the minutes (00 - 59) SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Port-Name	A designation of which serial port a communication or equipment failure has occurred at
Station #	A designation of what Acromag Station the equipment failure occurred at This can be any integer between 0 and 254 (decimal)
Channel #	A designation of what Acromag Channel the equipment failure occurred on This can be any integer between 0 and 34 (decimal)
Message	The text of the equipment failure message (of varying length)
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged, then this value is NOT ACKNOWLEDGED If the alarm was acknowledged, the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

8 2 2 Printer Equipment Fail Message

File Name Format *equip_fail_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-TimeStamp> **DATA ACQUISITION EQUIPMENT**
Port <Port-Name> **Station-ID** <Station #> **Channel** <Channel #>
<Message>
<State-of-Acknowledgement>

Field Descriptor	Definition
Message-TimeStamp	The date and time the message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31) MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 - 12) mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Port-Name	A designation of which serial port a communication or equipment failure has occurred at
Station #	Not Applicable
Channel #	Not Applicable
Message	The text of the equipment failure message (of varying length)
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged, then this value is NOT ACKNOWLEDGED If the alarm was acknowledged, the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

8 2 3 Panalarm Equipment Fail Message

File Name Format *equip_fail_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-TimeStamp> **DATA ACQUISITION EQUIPMENT**
Port <Port-Name> **Station-ID** <Line #> **Channel** <Cabinet Addr>
<Message>
<State-of-Acknowledgement>

Field Descriptor	Definition
Message-TimeStamp	The date and time the message was generated by TMACS The format of the date is **DD MMM YY** where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >**, where HH is the hour (0 - 12), mm are the minutes (00 – 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Port-Name	A designation of which serial port a communication or equipment failure has occurred at
Line #	A designation of which line on the code switch the failure occurred on This can be any integer between 0 and N (where N is the number of lines on code switch)
Cabinet-Addr	A designation of which Panalarm Cabinet the equipment failure occurred on This can be any integer between 0 and 7 (decimal)
Message	The text of the equipment failure message (of varying length)
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged, then this value is NOT ACKNOWLEDGED If the alarm was acknowledged, the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

8 2 4 ENRAF Equipment Fail Message

File Name Format *equip_fail_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-TimeStamp> **DATA ACQUISITION EQUIPMENT**
Port <Port-Name> **LINE** < COS-Addr> **CIU** <CIU-Addr> **Gauge** <Gauge-Addr>
<Message>
<State-of-Acknowledgement>

Field Descriptor	Definition
Message-TimeStamp	The date and time the message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >** where HH is the hour (0 - 12) mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g **12 Jan 95 11 43 23 a m**
Port-Name	A designation of which serial port a communication or equipment failure has occurred at
COS-Addr	A designation of which line on the code operated switch the equipment failure occurred on This can be any integer between 0 and 7 (decimal)
CIU-Addr	A designation of which Enraf CIU device the equipment failure occurred on This can be any integer between 0 and 9 (decimal)
Gauge-Addr	A designation of what Enraf Gauge device the equipment failure occurred on This can be any integer between 0 and 99 (decimal)
Message	The text of the equipment failure message (of varying length)
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged then this value is NOT ACKNOWLEDGED If the alarm was acknowledged the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

8 2 5 Westronics Equipment Fail Message

File Name Format *equip_fail_YYYY_MMDD_HHmm dat*

Record Format with Field Name Descriptors

<Message-TimeStamp> **DATA ACQUISITION EQUIPMENT**
Port <Port-Name> **Station-ID** <RTU #> **Channel** < Register #>
<Message>
<State-of-Acknowledgement>

Field Descriptor	Definition
Message-TimeStamp	The date and time the message was generated by TMACS The format of the date is **DD MMM YY**, where DD is the day (1-31), MMM is the month (Jan thru Dec) and YY is the year (95 - 99) The format for the time is **HH mm SS <a m /p m >** where HH is the hour (0 - 12), mm are the minutes (00 - 59), SS are the seconds (00 - 59) followed by a designation for either morning or afternoon E g , **12 Jan 95 11 43 23 a m**
Port-Name	A designation of which serial port a communication or equipment failure has occurred at
RTU #	A designation of what Westronic multiplexer the equipment failure occurred at This can be any integer between 1 and 255 (decimal)
Register #	A designation of what Acromag Channel the equipment failure occurred on This can be any integer between 10 and 109 (decimal)
Message	The text of the equipment failure message (of varying length)
State-of-Acknowledgement	The current acknowledgement state of this alarm If the alarm was not acknowledged then this value is NOT ACKNOWLEDGED If the alarm was acknowledged the this value is the <Ack-Timestamp> ALARM ACKNOWLEDGED

8 3 Discrete Sensor History Message

File Name Format *discrete_sensor_history_YYYY_MMDD ascii*

Record Format with Field Name Descriptors

<STN> | <LGR-date> | <LGR-time> | <Value> | <Annot> | <Q-status> | <A-type> | <C–date> | <C-time> | <S-Desc>

The delimiter is the vertical bar, ASCII character 124

Field Descriptor	Definition
STN	Sensor Tag Name is the identifier from the TMACS I/O Termination Point Listing document for this sensor, e g , **SY101-NSH-R07A-K-10-01**
LGR-date	Last Good Reading Date is the date when the sensor quality status, Q-status, was "**GOOD**" The format is MM-DD-YYYY with no leading zeroes, e g , **12-7-1994**
LGR-time	Last Good Reading Time is the time when the sensor quality status Q-status, was "**GOOD**" The format is HH MM SS, no leading zeroes and in military time 0 - 23 hours e g , **15 7 23**
Value	Value is the current or last good reading value for the discrete sensor in the form of any valid integer e g , **1, -1, 9**
Annotation	Annotation is a description of the sensor value meaning e g , **OPEN, CLOSED, CLOSING, OPENING**
Q-status	Quality Status is the indication of the state of the sensor value either good (the current sensor value is known) or unknown (the current sensor value is unknown), e g , **UNKNOWN** or **GOOD**
A-type	Alarm Type is the indicator of what type of alarm the sensor value is in, no-alarm, alarm-low or alarm-high
C-date	Collection Date is when the sensor value was received from the I/O device bridge The format is MM-DD-YYYY, (MM - month, DD - day YY - year with no leading zeroes before single digit months or days), e g **12-3-1994**
C-time	Collection Time is when the sensor value was received from the I/O device bridge The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours, e g , **18 9 2**
S-desc	Sensor Descriptor is a text description of the sensor, which can vary from a length of 1 to 40 characters, e g **SY103, SHMS-J, HIGH HYDROGEN ALARM**

8 4 Continuous Sensor History Message

File Name Format *continuous_sensor_history_YYYY_MMDD ascii*

Record Format with Field Name Descriptors 9 data fields

<STN> | <LGR-date> | <LGR-time> | <Value> | <Units> | <Q-status> | <A-type> | <C-date> | <C-time>

The delimiter is the vertical bar ASCII character 124

Field Descriptor	Definition
STN	Sensor Tag Name is the identifier from the TMACS I/O Termination Point Listing document for this sensor, e g , **SY101-TI-R17B-01**
LGR-date	Last Good Reading Date is the date when the sensor quality status, Q-status, was **"GOOD"** The format is MM-DD-YY with no leading zeroes), e g , **12-7-94**
LGR-time	Last Good Reading Time is the time when the sensor quality status, Q-status, was **"GOOD"** The format is HH MM SS, no leading zeroes and in military time 0 - 23 hours, e g , **15 7 23**
Value	Value is the current or last good reading value for the sensor in the form of any valid floating point number, e g , **-23 33, 90 0, 32767 0**
Units	Units is the engineering units the sensor value is reported in, e g , **F, CFM, %H2**
Q-status	Quality Status is the indication of the state of the sensor value, either good (the current sensor value is known) or unknown (the current sensor value is unknown), e g , **UNKNOWN** or **GOOD**
A-type	Alarm Type is the indicator of what type of alarm the sensor value is in, no-alarm, alarm-low or alarm-high
C-date	Collection Date is when the sensor value was received from the I/O device bridge The format is MM-DD-YY (MM - month, DD - day, YY - year with no leading zeroes before single digit months or days) e g , **12-3-94**
C-time	Collection Time is when the sensor value was received from the I/O device bridge The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours, e g **18 9 2**

8 5 Double-Shell And Single-Shell Tank Daily FILES

File Name Format *dst_data_YYYY_MMDD_HHmm rdbms* for double-shell tanks
sst_data_YYYY_MMDD_HHmm rdbms for single-shell tanks

Record Format with Field Name Descriptors

<R-date> | <R-time> | <STN> | <Value> | <Units> | <Q-status> | <A-type> | <LGR-date> | <LGR-time>

The delimiter is the vertical bar, ASCII character 124

Field Descriptor	Definition
R-date	Reporting Date is the date at which the sensor value was giving a reading as given in the Value field The format is MM-DD-YY (MM - month, DD - day, YY - year with no leading zeroes before single digit months or days), e g , **12-3-94**
R-time	Reporting Time is the time at which the sensor value was giving a reading as given in the Value field The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours e g , **18 9 2**
STN	Sensor Tag Name is the identifier from the TMACS I/O Termination Point Listing document for this sensor, e g , **SY101-TI-R17B-01**
Value	Value is the current or last good reading value for the sensor in the form of any valid floating point number, e g , **-23 33, 90 0, 32767 0**
Units	Units is the engineering units the sensor value is reported in, e g , **F, CFM, %H2**
Q-status	Quality Status is the indication of the state of the sensor value either good (the current sensor value is known) or unknown (the current sensor value is unknown), e g , **UNKNOWN** or **GOOD**
A-type	Alarm Type is the indicator of what type of alarm the sensor value is in, no-alarm, alarm-low or alarm-high
LGR-date	Last Good Reading Date is the date when the sensor quality status, Q-status, was **"GOOD"** The format is MM-DD-YY with no leading zeroes), e g , **12-7-94**
LGR-time	Last Good Reading Time is the time when the sensor quality status, Q-status, was **"GOOD"** The format is HH MM SS, no leading zeroes and in military time 0 - 23 hours, e g , **15 7 23**

8 6 Discrete Shutdown Message

File Name Format *discrete_shutdown_YYYY_MMDD ascii*

Record Format with Field Name Descriptors

<STN> | <LGR-date> | <LGR-time> | <Value> | <Annot> | <Q-status> | <A-type> | <C–date> | <C-time> | <S-Desc>

The delimiter is the vertical bar, ASCII character 124

Field Descriptor	Definition
STN	Sensor Tag Name is the identifier from the TMACS I/O Termination Point Listing document for this sensor, e g , **SY101-NSH-R07A-K-10-01**
LGR-date	Last Good Reading Date is the date when the sensor quality status, Q-status, was **"GOOD"** The format is MM-DD-YYYY with no leading zeroes, e g , **12-7-1994**
LGR-time	Last Good Reading Time is the time when the sensor quality status, Q-status, was **"GOOD"** The format is HH MM SS no leading zeroes and in military time 0 - 23 hours, e g , **15 7 23**
Value	Value is the current or last good reading value for the discrete sensor in the form of any valid integer, e g , **1, -1, 9**
Annotation	Annotation is a description of the sensor value meaning e g , **OPEN, CLOSED, CLOSING, OPENING**
Q-status	Quality Status is the indication of the state of the sensor value, either good (the current sensor value is known) or unknown (the current sensor value is unknown), e g , **UNKNOWN** or **GOOD**
A-type	Alarm Type is the indicator of what type of alarm the sensor value is in, no-alarm, alarm-low or alarm-high
C-date	Collection Date is when the sensor value was received from the I/O device bridge The format is MM-DD-YYYY, (MM - month, DD - day, YY - year with no leading zeroes before single digit months or days) e g , **12-3-1994**
C-time	Collection Time is when the sensor value was received from the I/O device bridge The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours, e g , **18 9 2**
S-desc	Sensor Descriptor is a text description of the sensor, which can vary from a length of 1 to 40 characters e g , **SY103, SHMS-J, HIGH HYDROGEN ALARM**

8 7 Continuous Shutdown Message

File Name Format *continuous_shutdown_YYYY_MMDD ascii*

Record Format with Field Name Descriptors 9 data fields

<STN> | <LGR-date> | <LGR-time> | <Value> | <Units> | <Q-status> | <A-type> | <C-date> | <C-time>

The delimiter is the vertical bar, ASCII character 124

Field Descriptor	Definition
STN	Sensor Tag Name is the identifier from the TMACS I/O Termination Point Listing document for this sensor, e g , **SY101-TI-R17B-01**
LGR-date	Last Good Reading Date is the date when the sensor quality status, Q-status, was "**GOOD**" The format is MM-DD-YY with no leading zeroes), e g , **12-7-94**
LGR-time	Last Good Reading Time is the time when the sensor quality status Q-status, was "**GOOD**" The format is HH MM SS no leading zeroes and in military time 0 - 23 hours, e g , **15 7 23**
Value	Value is the current or last good reading value for the sensor in the form of any valid floating point number, e g , **-23 33, 90 0, 32767 0**
Units	Units is the engineering units the sensor value is reported in, e g , **F, CFM, %H2**
Q-status	Quality Status is the indication of the state of the sensor value, either good (the current sensor value is known) or unknown (the current sensor value is unknown), e g , **UNKNOWN** or **GOOD**
A-type	Alarm Type is the indicator of what type of alarm the sensor value is in, no-alarm, alarm-low or alarm-high
C-date	Collection Date is when the sensor value was received from the I/O device bridge The format is MM-DD-YY (MM – month, DD - day, YY - year with no leading zeroes before single digit months or days), e g , **12-3-94**
C-time	Collection Time is when the sensor value was received from the I/O device bridge The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours, e g , **18 9 2**

8 8 Performance Data Message

File Name Format *perf_data_YYYY_MMDD dat*

Record Format with Field Name Descriptors

<L-date> , <L-time> , <Cur_Cpu>, <Avg_Cpu>, <Mem_Usage>, <Reg1_Mem>, <Reg2_Mem>, <Item_Count>

Field Descriptor	Definition
L-date	Logging Date is the date at which the performance data was logged to the file The format is MM-DD-YY, (MM - month, DD - day YY - year with no leading zeroes before single digit months or days), e g , **12-3-94**
L-time	Logging Time is the time at which the performance data was logged to the file The format is HH MM SS with no leading zeroes used on single digit hours, minutes or seconds, military time 0 - 23 hours e g , **18 9 2**
Cur_Cpu	Cur_Cpu is the G2 performance meter giving the Current G2 CPU usage in percent E g , **15 9**
Avg_Cpu	Avg_Cpu is the 5 minute average of the G2 current CPU usage meter E g , **23 3**
Mem_Usage	Mem_Usage is the Current G2 memory usage in megabytes E g , **78 5**
Reg1_Mem	Reg1_Mem is the G2 performance meter giving the Current G2 Region 1 memory usage, e g , **56789679**
Reg2_Mem	Reg2_Mem is the G2 performance meter giving the current G2 Region 2 memory usage, e g , **456789**
Item_Count	Item_Count is the G2 performance meter giving the current G2 item count, i e , number of items in the system E g **70311**

DISTRIBUTION SHEET

To	From	Page 1 of 1
Distribution	Process Engineering	Date 12/8/99
Project Title/Work Order		EDT No 627258
TMACS As-Built Design document		ECN No N/A

Name	MSIN	Text With All Attach	Text Only	Attach / Appendix Only	EDT/ECN Only
D A Barnes	R2-12	X			
T L Bennington	S5-15				X
S C Cantrell	R3-43	X			
J G Field	R2-12				X
M J Holm	R2-12	X			
W J Kennedy	S5-07				X
P C Miller	R1-51				X
D A Selle	S5-03				X
P L Smith	S7-01				X
P F Ulk	S4-43	X			
R R Wandling	R1-01	X			