Electronic Information Environment (EIE)

Service Specification Document/Interface Control Document

ISSCF/PBC Data Exchange

Dead Message Service

(Canada and Industry)

For Industry partners who have been contracted to participate in an In-Service-Support phase of a Weapon System or Platform with Canada Department of National Defence.

EIE Project

document identification	identifier du document
issue date	date de diffusion
09 November 2015	
version	version
Version 2.4	
OPI	BPR
EIE Solution Office	
designator	désignation
EIE Project	
group / division	groupe / division
ADM(IM) / DGEAS	

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

Table of Contents

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

List of Figures

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

1 Introduction

This document establishes an interface between Canada Electronic Data Exchange (EDE) system and Industry responsible for maintenance of a fleet subject to In-Service Support Contracting Framework (ISSCF) and Performance based Contracting (PBC). This interface will be used by Canada EDE and Industry to report **Dead Messages** to each other. A dead message is:

i. A message which the sender (Canada EDE or Industry) determines cannot be successfully transmitted to the intended recipient (Industry or Canada EDE);

ii. A message for which the recipient cannot formulate a valid technical response.

The use of the Dead Message Service allows the sender and intended recipient to be aware of the cause of a message transmission failure to complete its entire lifecycle between a provider and consumer.

On being notified about the cause the provider and the consumer will be able to rectify the cause of the failure and perform the appropriate rectification measures within their organizational boundaries/ systems to ensure there is no net loss of data of a transaction from a message delivery being incomplete.

The business content which was in a message which was deemed to be dead message may be re-transmitted in a new message(s) when appropriate.

To support management of dead messages between Canada EDE and Industry, both systems need to support specific Web Service operations as well as request and response XML schemas as described in this document. The Dead Message service includes operations for Canada EDE and Industry to report acknowledgement and error messages back to each other.

Note: If the Dead Message Service is itself is inoperative, alternate form of electronic communication may be initiated, but the specific model is not in scope for this specification.

1.1 Intended Audience

- Industry Partner System Designers
- Canada EDE Designers
- Industry Testers
- Canada EDE Testers

1.2 References

[Ref. 1] Electronic Information Exchange Service Interaction Model (**Section 5.4.5**)

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

2 Business Information

The Electronic Information Environment (EIE) is the broad architecture of services and the supporting technical and network infrastructure to enable information exchange for ISSCF and PBC. The Dead Message Service is a technical service which is applicable to any business domain and is usable by Canada EDE and Industry partners. In the broad EIE view Canada EDE and Industry are both service providers and service consumers. For example, in the Supply domain, Industry is the nominal service provider for the Part Demand service and Canada EDE is the nominal service consumer of Part Demand. Similarly, Canada EDE is the nominal service provider of the Part Issue service and Industry the consumer.

For any contract both parties (Canada EDE and Industry) are both providers and consumers of services and so to describe the Dead Message service a more generic terminology is required[1]. Also, since the subject is dead ***messages***, it makes sense to refer to the parties in a contract in terms of messages not provider/consumer of services. In the remainder of this document the terms ***message sender*** (equivalent to service consumer) and ***message recipient*** (equivalent to service provider) will be used.

The idea of a contract for every service which imposes responsibilities on both Canada EDE and Industry, each as sender and recipient of messages, is central to understanding the Dead Message Service.

A message may be declared dead by the sender or the recipient. A message is declared dead when a message sender or recipient determines their inability to fulfill the “intent” of a service contract.

The Dead Message service does not send the message that was declared as a dead message; it sends messages about dead messages. In this specification, the messages sent by the Dead Message service are called ***obituary messages***. For example, after some retries a Part Demand service message may be declared dead (i.e., the Part Demand is a dead message). The Dead Message service will create an obituary message using meta-data from the dead Part Demand Message[2]. Essentially, the obituary is the “business object” of the Dead Message service.

Within Canada, maintenance business processes are supported by two types of information systems, known generically as:

- Canada Maintenance Management System (CMMS)
- Canada Supply System (CSS).

Currently both functions are supported within Canada by the Defence Resource Management Information System (DRMIS).

[1] Most EIE service specifications read “Canada does this ….”, “Industry does this …” which lacks symmetry.

[2] This deviates from the naming convention that the ABC Service (e.g. Notification) sends ABC messages. This is warranted as the messages being sent by the Dead Message service are not themselves dead.

2.1 Business Processes

This service is designed to be generic to support handling of dead messages without limitation to particular business processes.

A message may be declared as dead by any EIE service.

The Dead Message service is run on a periodic basis at a period to be determined between Canada and Industry.

2.2 Business Triggers

The business trigger for the Dead Message service is the availability of a message that has been declared as being dead.

More significant are the “triggers” which result in a service declaring a dead message. The following are possible causes of death:

- Infrastructure error(s) and retries all failed;
- Message recipient cannot formulate a valid technical response3;
- Time To Live expires before business response message received.

Over time there may be other causes of death defined. The Dead Message Service has to work whatever the cause of the message being marked as dead.

See Section 4.1 Service Context for further discussion of when a message can be declared dead.

[3] As a general rule, EIE service specifications deliberately make minimal constraints on the fault technical response. In any case where it is impossible for the recipient to formulate a valid technical response the specification of the service which sent the invalid message should be reviewed.

3 Business Constraints

Constraints on *Usage* of the Service

1) Every invocation of a service operation shall be secured using secure credentials such as: Public Key Infrastructure (PKI) Certificate.

2) Every service definition (contract) includes an error operation or explicitly stipulates manual handling of errors. The Dead Message service shall **not** be used to report validation errors in business objects or any business error.

3) If a Recipient has a message which is invalid only in business objects then the contractually defined error reporting service must be used and the message may **not** be declared as dead.

4) The Dead Message service's payload (i.e. the obituary) should **not** carry any business object content from the original message that was marked as being dead. (See note below.)

5) The Dead Message service's payload (i.e. the obituary) will only provide metadata with regards to the original message that was marked as being dead.

Constraints on *Behaviour* of the Service

6) If a Sender or Receiver declares a message dead, it should not assume the other party also knows the message is dead. (See Section 4.4 Discordance Scenarios.)

7) Both parties will authorize invocations of operations of Dead Message service.

8) Dead Message messages will be signed using digital certificates between Canada EDE and Industry. Please see Service Interaction Model [Ref. 1] for details.

9) Industry may attempt to repeat operation invocations in response to technical faults. This behaviour is controlled by parameters for each operation. Please see Service Interaction Model [Ref. 1] for details.

Note:

With respect to constraint number 4, there are three reasons why the obituary message should not contain business objects:

- The Dead Message service is not intended to be used an alternate means of delivery of business objects. It would be incorrect for the recipient of an obituary message to try to recreate a business object from content of the obituary message.

- Attempts to use business objects from an obituary may conflict with any manual actions taken to remediate the prior failure to deliver business objects.

- In the course of determining the cause of death of a message, technical staff/monitoring programs may exchange obituary messages through non-secured channels (e.g., email).

4 Service Use Case

The requirements for the Dead Message service are defined by one use case with several scenarios.

This use case describes the steps in sending the obituary message and technical response.

4.1 Service Context

The Dead Message service may be used in any service context for any of the other EIE services. The context for the Dead Message service is **after** another service has declared a dead message. The context depends on whether a message sender or receiver declared the dead message.

Canada's and Industries EIE services should be operating in a context which includes an automated monitoring program or over-sight by technical staff, or both. The term ***monitoring agent*** is used to refer to this role in the system context. When a message sender or recipient declares a message as dead the following steps occur:

- The sender/recipient may alert their monitoring agent;
- Technical Staff will investigate the under-lying problem. In instances where participation is required between Canada DND and Industry staff the data in the dead message transmission will aid in supporting resolution of the dead message occurrences.
- Detailed Dead Message information is sent according to the pre-defined periods. This can be useful in resolving any missing business objects[4].

4.1.1 Sender Declares Dead Message

The most common context is when the message sender has made the specified number of retry attempts without receiving a technical response. This is detailed in the following diagram, Figure 4-1.

[4] The approach to resolve missing business objects after dead messages is out-of-scope of this specification and should be addressed in the Message Operational Model document for each business domain.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

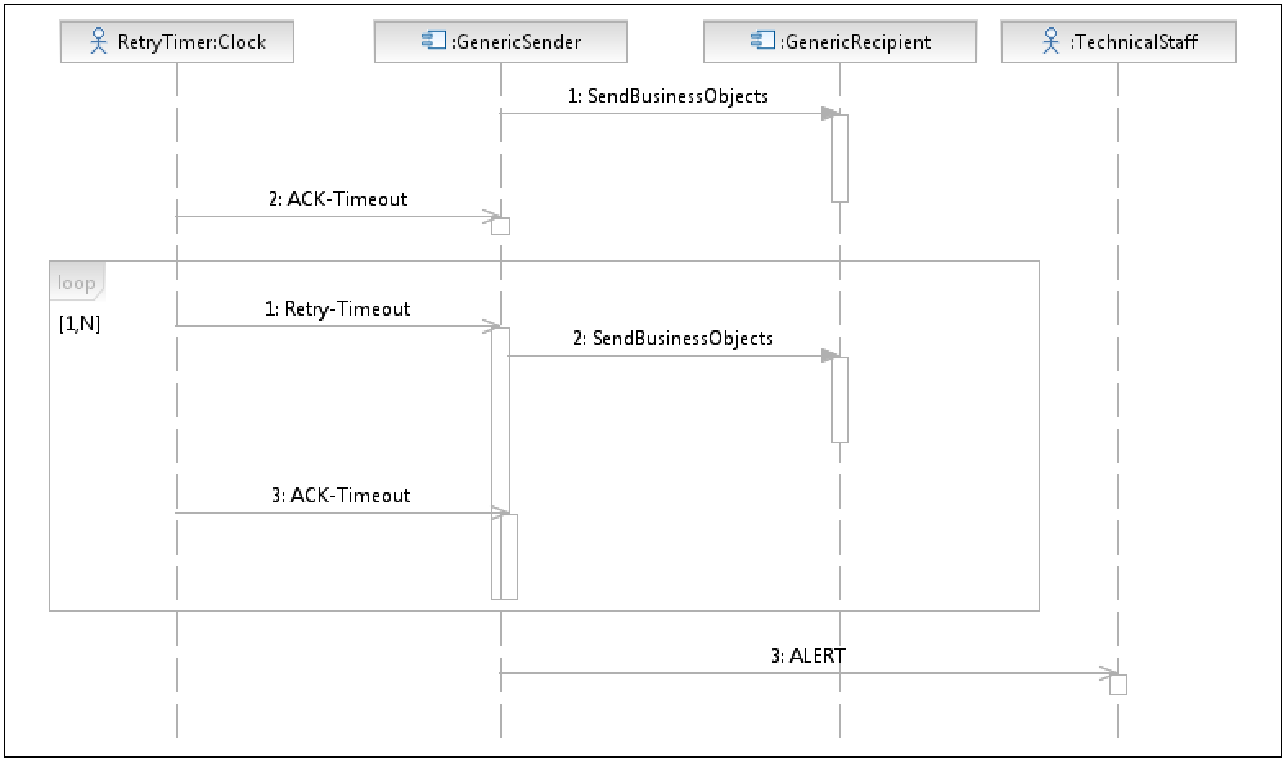

Figure 4-1 Service Context – Sender Declares Dead Message

Service Context Detail	
Scenario	A generic message sender makes multiple attempts to send a message to a generic recipient and does not receive a corresponding technical response.
Pre-Condition	A prior business trigger has occurred and a Generic Sender attempts to send business objects to a recipient. • Number of retries 'N' is known, per individual service specifications. • 'ACK Time' is known, per individual service specifications. • 'Retry Time' is known, per individual service specifications.
Post-Condition	Alert received by message sender monitoring agent. • Undelivered messages are marked as being dead • Monitoring agent will determine if the message should be sent to service recipient.

Steps	1) GenericSender invokes SendBusinessObject()[5] operation. There is no technical response. 2) An ACK timeout occurs and Generic Sender enters retry loop. **For up to "N" retry attempts.** 1) A Retry timeout occurs after the defined retry time interval. 2) GenericSender invokes SendBusinessObject() operation. There is no technical response. 3) An ACK timeout occurs and GenericSender continues or exits the loop. **After "N" attempts.** 3) GenericSender sends an alert message to Sender's monitoring agent.

4.1.2 Recipient Declares Dead Message

There is one identified special case where the Recipient may declare a dead message.

1) The Recipient has an invalid message from the Sender and the Recipient cannot formulate a contractually required fault technical response.
   - In this case the sender will not receive a technical response and ultimately the Sender will also declare the message as dead.

4.2 Successful Request and Technical Response

This is the main or "Happy Day" scenario (Figure 4-2) for the Dead Message Service between a generic message sender and recipient.

The Dead Message service differs from other EIE services mainly in the context in which it is used. Once the Dead Message service is initiated its behaviour follows the same pattern as other services.

Service contracts in EIE rely on the concept of ***custody*** of a message. A message sender is the initial custodian of a message and retains custody until a positive technical response is received from the recipient, if the recipient returns a fault technical response the sender retains custody. Only the custodian of a message may declare it dead. (Section 4.4, 'Discordance Scenarios' describes some potential complications.)

In the following sequence diagrams, note that the DeadMessage service involves interaction between the custodian's Dead Message service and the DeadMessage service of the other partner in the original contract.

[5] A generic operation 'SendBusinessObjects()' is used in scenarios in this document.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

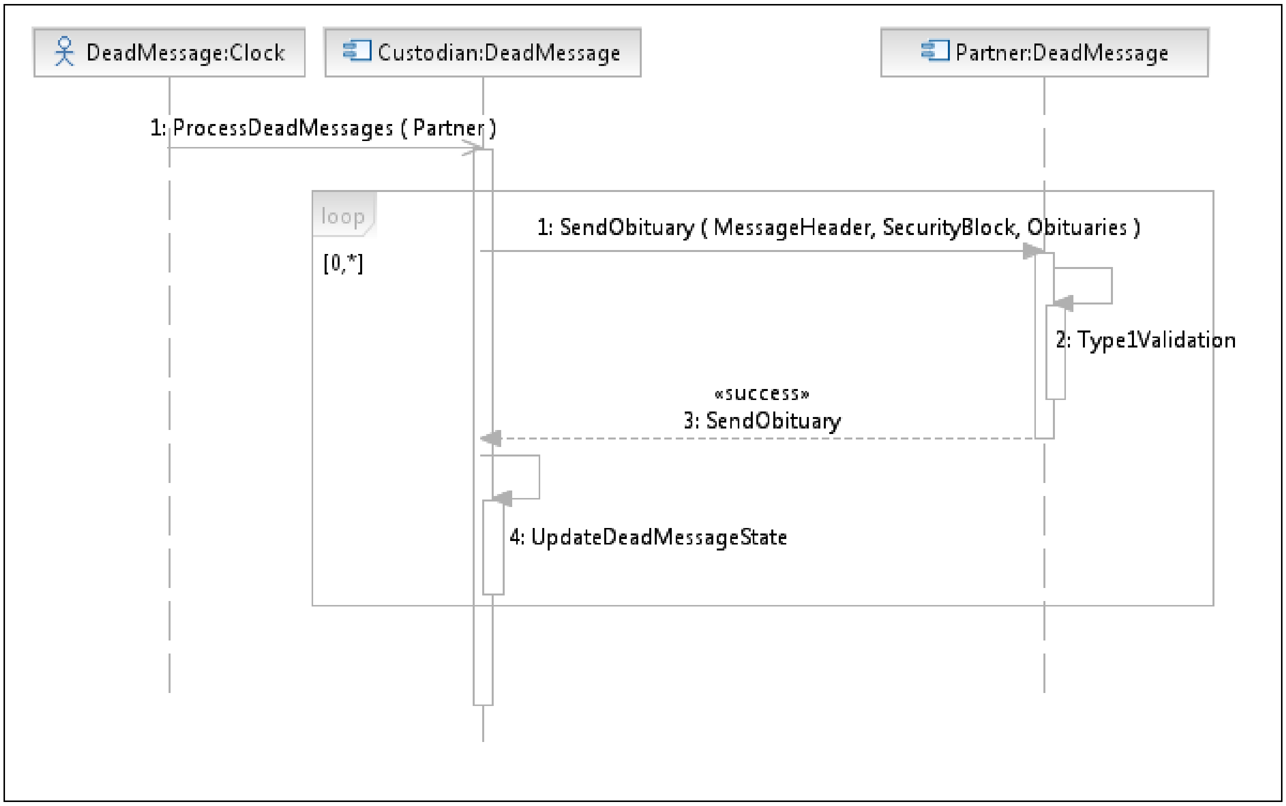
DeadMessage:Clock
Custodian:DeadMessage
Partner:DeadMessage
1: ProcessDeadMessages (Partner)
loop
[0,*]
1: SendObituary (MessageHeader, SecurityBlock, Obituaries)
2: Type1Validation
«success»
3: SendObituary
4: UpdateDeadMessageState

Figure 4-2 Dead Message "Happy Day"

Dead Message Main Flow	
Scenario	One or more dead messages have occurred and investigation of the cause(s) is underway.
Pre-Condition	Dead message states have been preserved.
Post-Condition	Obituary messages arrive at Partner. Dead message states are updated to prevent re-transmission of obituaries.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

Steps	1) Clock triggers custodian's DeadMessage component to process dead messages for a Partner. **For all dead messages for the given partner:** 1) Custodian's DeadMessage component invokes SendObituary() operation on Partner's DeadMessage component - passing one or more obituaries. 2) Partner's DeadMessage component performs Type 1 Validation on the input message. In this scenario there is no error. 3) Partner's DeadMessage component returns a 'success' technical response. 4) Custodian's DeadMessage component updates state of dead message(s).

Each invocation may contain more than one obituary.

4.3 Alternate Scenarios

The following scenarios apply to all uses of Dead Message service.

For the Dead Message service there are no "business" errors. The content of obituary messages is used by monitoring agents and any errors in the content are resolved directly between the partners.

This scenario applies to Type 1 errors - namely those errors detected prior to the Partner's DeadMessage component accepting custody of a message (Figure 4-3).

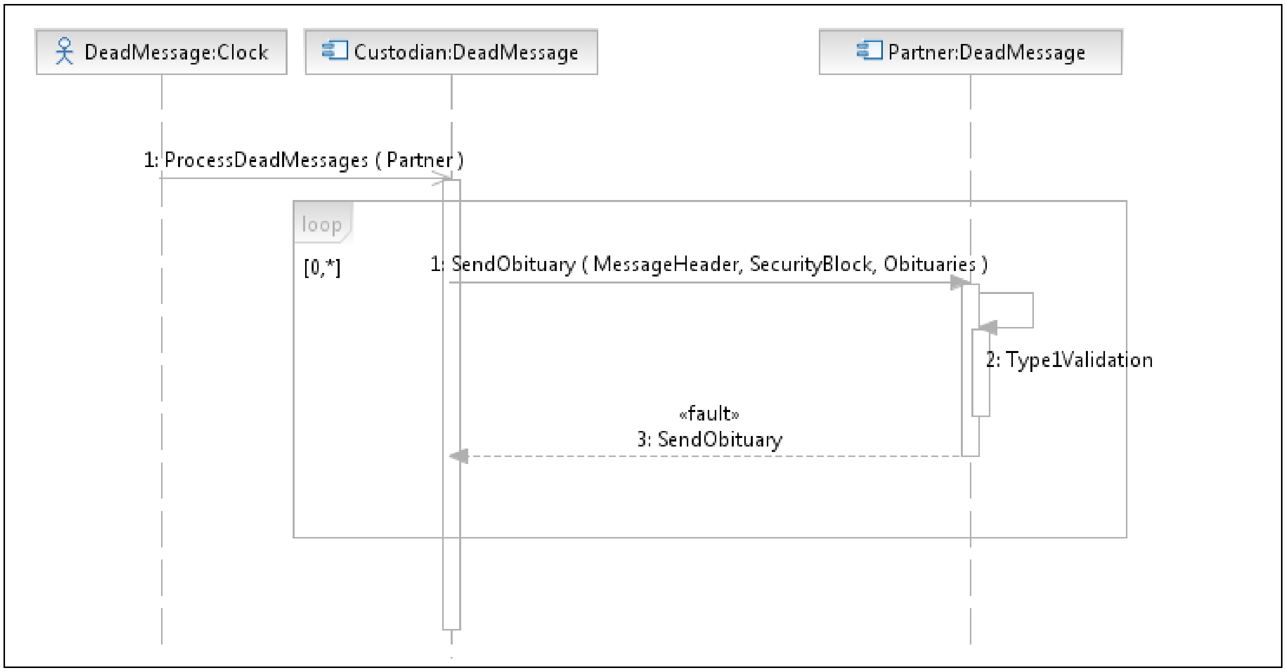

Figure 4-3 Dead Message Type 1 Error

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

Alternate Scenario – Type 1 Error	
Scenario	Type 1 error sending obituary messages.
Pre-Condition	Same as Main Flow.
Post-Condition	Some obituary messages do not arrive at Partner. Dead message states are updated to prevent re-transmission of obituaries. Custodian's technical staff are alerted.
Steps	1) Clock triggers Custodian's DeadMessage component to process dead messages for a Partner. **Loop for all dead messages for the given partner:** 1) Custodian's DeadMessage component invokes SendObituary() operation on Partner's Dead Message component passing one or more obituaries. 2) Partner's DeadMessage component performs Type 1 Validation on the input message. In this scenario there is an error. 3) Partner's DeadMessage component returns a 'fault' technical response. **Custodian's DeadMessage component continues with any remaining dead messages for the given partner.**

Each invocation may contain more than one obituary.

If an invocation of SendObituary() returns a fault:

- The usual re-try mechanism is used;
- Processing continues with other dead messages.

Alternate Scenario 2 (Service unresponsive)	
Scenario	Declarer-side Dead Message does not receive technical response within ACK_TIME_INTERVAL.
Pre-Condition	Same as Main Flow.
Post-Condition	Manual intervention. Custodian's DeadMessage service does **not** mark the obituary message as dead.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

Steps	1) Custodian's DeadMessage invokes SendObituary() operation on Partner's DeadMessage component passing one or more obituaries. 2) Custodian's DeadMessage component does not receive any response within the allowed ACK_TIME_INTERVAL. 3) Custodian's DeadMessage component will retry sending the message up to the defined maximum retry count and/or Time to Live interval. 4) If there is no response, then Custodian's DeadMessage component o Does **not** change the state of the original dead message, the next scheduled run of the DeadMessage service will try again. o Sets the state of the unsuccessful obituary message to permanently failed. There will **not** be an obituary for an obituary.

4.4 Discordance Scenarios

There are certain scenarios where the Sender and Receiver may not have consistent information on the death of a message.

1) A Sender may declare a message as dead after all retry attempts have failed. However, it is possible that on one of the attempts the Recipient actually received the message, the technical response was lost, the Recipient does not know it was lost, and the Recipient considers the message not dead. The recipient may perform business processing on a message the Sender considers dead. This should be detected when the Sender-side obituaries are sent to the Recipient.

In other words, a discordance may arise when both sender and receiver are acting as custodians.

The process for handling a discordance is described in the Service Interaction Model [Ref. 1].

5 Service Description – Dead Message Service

5.1 Service Overview

The Dead Message service presumes each system’s message handling maintains a message state in some form.

The Dead Message service requires interacting web services exposed by Canada EDE and Industry.

Canada EDE and Industry will implement and expose a service and operation which allow them to send each other obituary messages (see Section 7 for message definition). After receipt of the input message, the implementation of the service will return a technical response indicating success or failure in accepting the obituary message.

Message interaction is further described in Electronic Information Exchange Service Interaction Model [Ref. 1].

5.2 Service Properties

Service Property	Description
Enterprise Service Name (Business)	Dead Message Service
Enterprise Service Name (Technical)	DeadMessageService
Purpose	This service supports a business process where Canada EDE and its Partners can inform each other of dead messages.
Business Response Time Interval	Not Applicable
Service Domain	Cross Domain
Business Owner	ADM (IM)
Service Grouping	Cross Domain / Dead Message
Service Provider	Canada EDE and Industry
Target Service Consumers	Canada EDE and Industry
Business Process Supported (now)	Any message-based exchange business process.
Business Process Supported (future)	

Service Property	Description
Business Objective Supported	See Section 2: Business Information.
Expected life time	The full lifecycle of the subject weapons system.

5.3 'SendObituary()' Operation

This operation is used by Canada EDE or Industry to send an obituary message to the other party in a contract.

Any implementation of this operation will perform Type 1 validation on the obituary message and will return a status or fault information to Industry in a technical response. A status of "**success**" is confirmation the other party accepts custody of the obituary message. Any returned fault implies the other party does NOT accept the message and error processing (as per Section 4: Service Use Case/Interaction Model) is performed.

5.4 'SendObituaryAck()' Operation

This operation is optionally used by Canada EDE or Industry to send an acknowledgement message to the other party after internal message processing accepts an obituary message (no Type 2 errors, see Service Interaction Model [Ref. 1]). Any implementation of this operation will perform Type 1 validation on the acknowledgement message and will return a technical response to Industry.

The use of this operation in a "pessimistic" scenario is to be confirmed between Canada and Industry on a per-fleet basis.

5.5 'SendObituaryError()' Operation

This operation is used by Canada EDE or Industry to send an error message to the other party when an error is detected after the initial technical response to 'SendObituary()'. Any implementation of this operation will perform Type 1 validation on the error message and will return a technical response to Industry/Canada.

The use of this operation is expected to be infrequent as handling of dead messages is primarily a manual process.

6 Information Model

This section describes the "business objects" which are used in the Dead Message service.

Note that all date or time values must be in **Coordinated Universal Time (UTC).** If an explicit time zone offset is not provided it is assumed to be zero.

The XML Schema is the authoritative source for purpose of the information exchange.

6.1 Obituary

Figure 6-1 shows the Information Model used in the Dead Message service. In the following *underlined italic* text refers to specific classes.

The *Obituary* class is used to describe metadata about a dead message. Each instance of *Obituary* class contains:

- A *MessageHeader*, which is the header of the dead message.
- A *SecurityBlock*[6], which is the header of the dead message.

Rules:

i. If the message was declared dead by the **sender**, the *MessageHeader* and *SecurityBlock* instances are **mandatory** in the obituary since the sender must have the correct information.

ii. If the message was declared dead by the **recipient,** the *MessageHeader* and *SecurityBlock* instances are **optional** in the obituary since the recipient may not have the correct information.

The *Obituary* class contains one or more instances of *ErrorDetail* to indicate the cause of death.

In addition, *Obituary* class has the following attributes:

- LastAttempt: the time of the last attempt to send the dead message;
- AttemptedURL: the URL which the service was attempting to send to prior to message death;
- AttemptedService: the name of the service which was attempting to send to prior to message death;
- AttemptedOperation: the name of the operation which was attempting to send to prior to message death;
- DeadIdentifier: when the message was declared dead by the recipient and the recipient does not have the header of the dead message then DeadIdentifier is set to some meaningful value-.

[6] Not required if there is no business object included.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

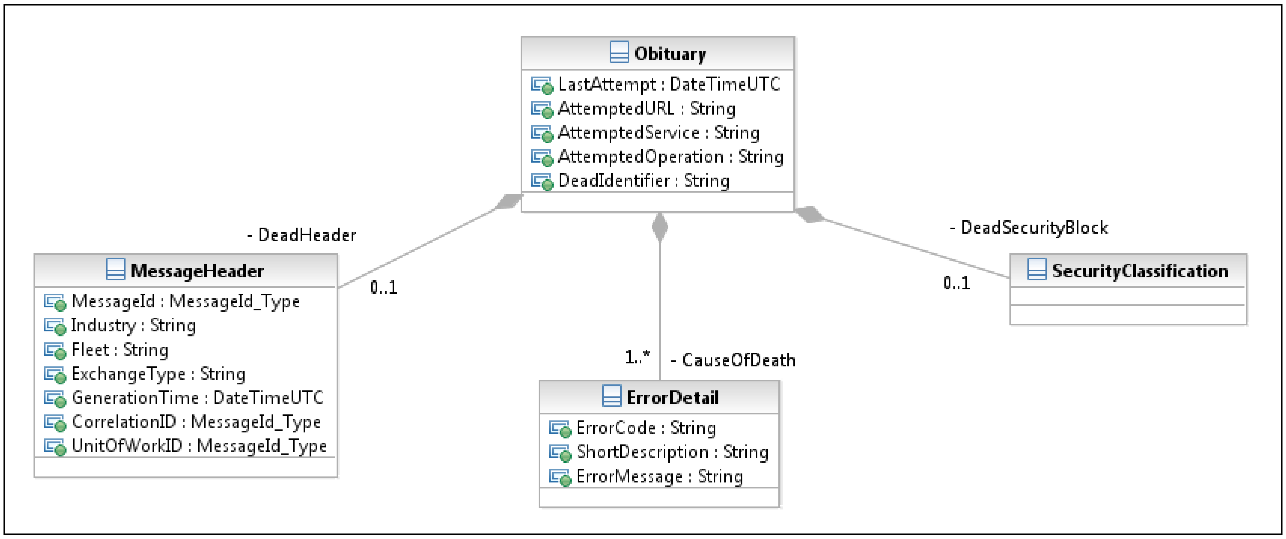

Figure 6-1 Obituary for a dead message

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

7 Operation Message Model

This section describes how the business objects described above (Section 6 Information Model) are aggregated for the purpose of ISSCF/PBC information exchange.

The Exchange Type, Message Id and Generation Timestamp of obituary message uniquely identify a BIZID of an “Obituary” message.

The Dead Message service follows the request/response model and each operation definition includes a distinct input, output and fault message. See Service Interaction Model [Ref. 1] for definition of the common MessageHeader element.

Note that all date or time values must be in **Coordinated Universal Time (UTC).** If an explicit time zone offset is not provided it is assumed to be zero.

7.1 Dead Message Input Body

As shown in Figure 7-1, a Dead Message input body consists of:

- A Message Header;
- A Security Block;
- One or more Obituary objects (and contained objects as in Figure 6-1).

The *InputBody* instance and the *Obituary* instances will of course contain different instances of *MessageHeader* and *SecurityClassification*. See the Annex to this document for a sample xml document to clarify the usage of MessageHeader and SecurityBlock constructs.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

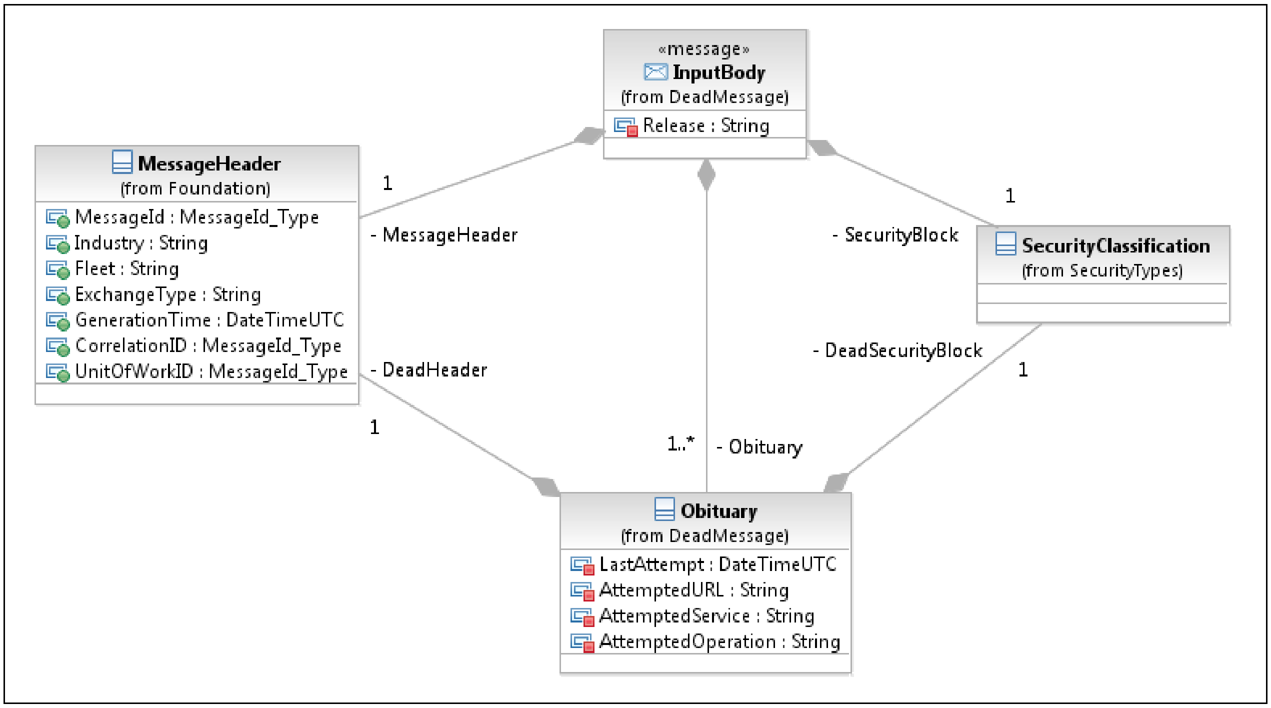

Figure 7-1 DeadMessage Input Body

The MessageHeader UnitofWorkID and Correlation ID are not used as part of the Dead Message service.

The InputBody also contains an attribute “Release” which designates the release of the Dead Message InputBody and the Dead Message service. The “Release” attribute appears in every instance of the InputBody to allow any input body instance to be traced back to the appropriate release. A “Release” attribute appears in all message bodies.

7.2 Dead Message Output Body

The output of the SendObituary() operation is the DeadMessage OutputBody. As shown in Figure 7-2, the output body consists of:

- A Message Header;
- A Custody object.

The DeadMessage OutputBody has no security block. It must not contain any sensitive or protected information.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

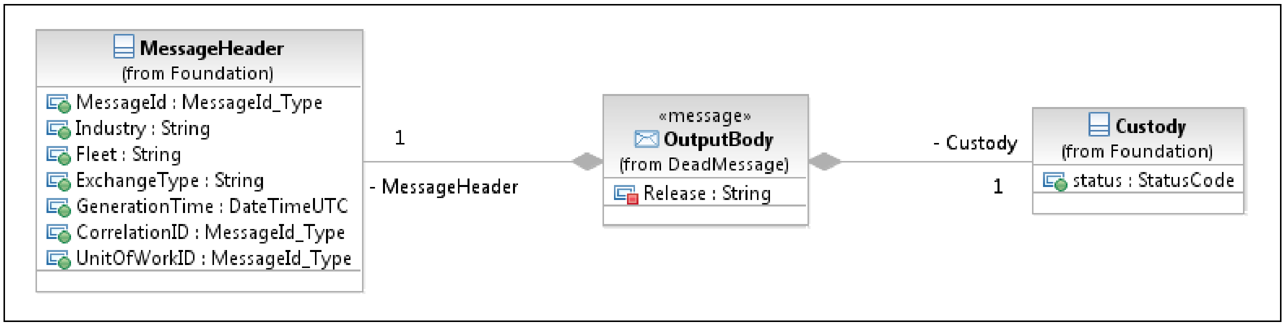

Figure 7-2 DeadMessage Output Body

For a Dead Message OutputBody:

- The MessageHeader Message Id is a new unique value;
- The MessageHeader Correlation ID is set to the MessageID of the Dead Message Input Body;
- The MessageHeader Exchange Type must be set to the Exchange Type of the Dead Message InputBody;
- The value of the Custody status field is “success”.

7.3 Dead Message Fault Body

A fault returned by the SendObituary() operation uses the Dead Message FaultBody element. As shown in Figure 7-3, the DeadMessageFaultBody consists of:

- A Message Header;
- A Security Block;
- A FaultBlock.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

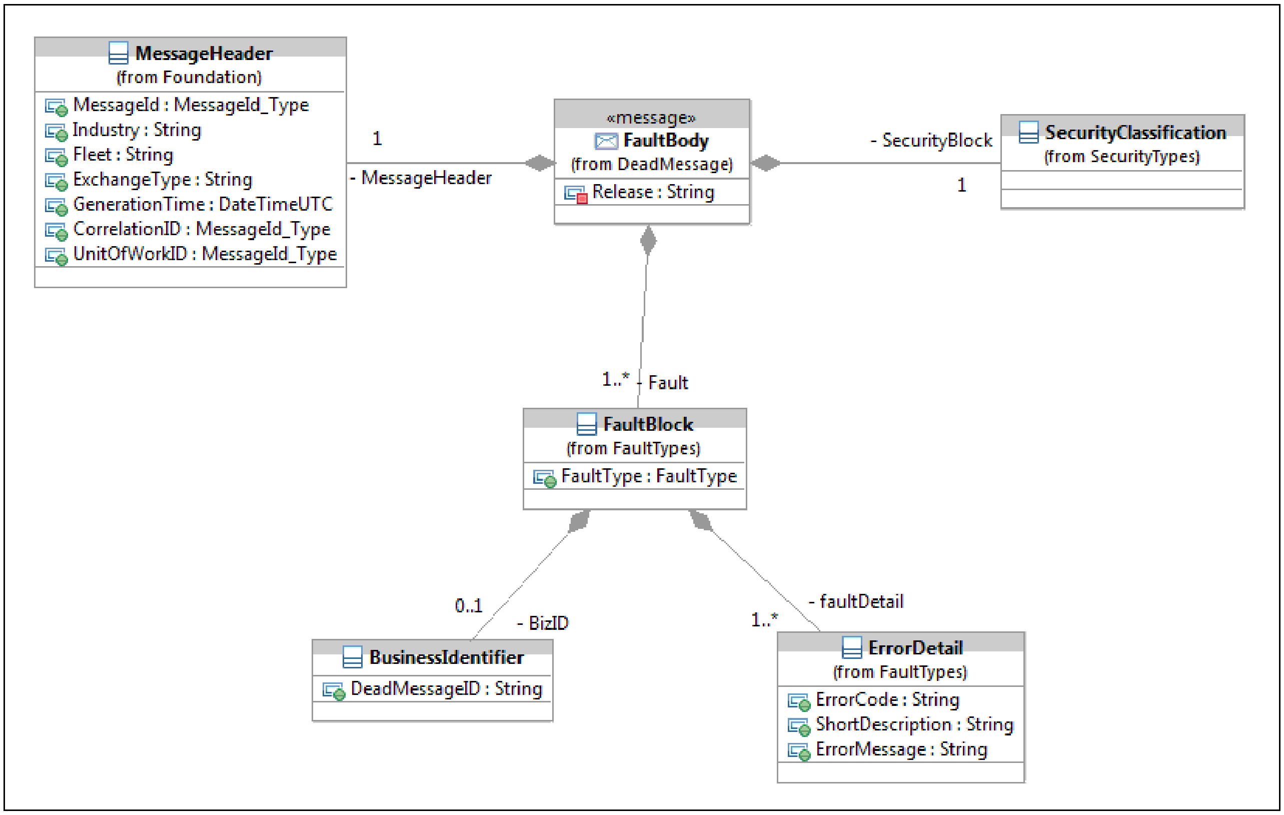

Figure 7-3 DeadMessage Fault Body

MessageHeader is mandatory, but only MessageID and GenerationTime are mandatory within the header. This is for the scenario where the input message is so damaged that the necessary attributes cannot be recovered.

7.4 Dead Message Acknowledgement Input Body

The input to the SendObituaryAck() operation consists of a Message Header, a list of business identifiers, and a "success" status indicating the business objects were accepted (see Figure 7-4).

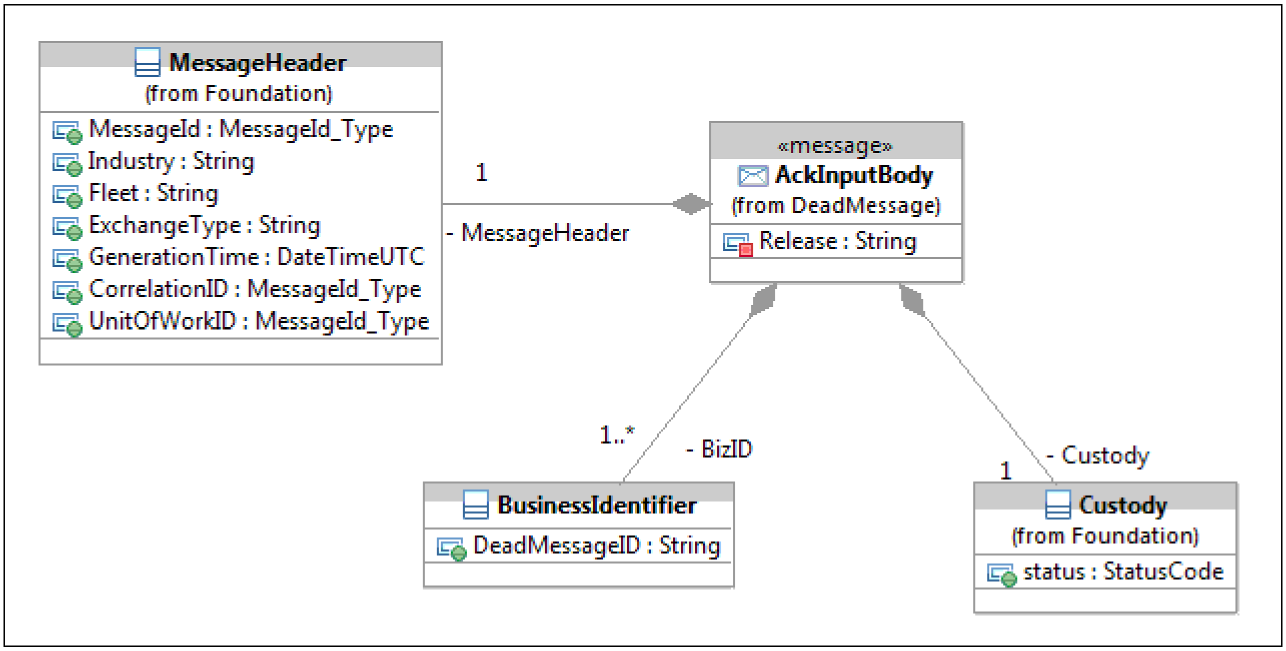

Figure 7-4 Dead Message Acknowledgement Input Body

The Message Header has a new unique Message Id and the Correlation ID is set to the Message Id of the Dead Message Input Body which is being acknowledged.

7.5 Dead Message Error Input Body

The input to the SendObituaryError() operation consists of a Message Header, a Security Block and a list of Error Blocks (Figure 7-5). Each Error Block references a business object (by business identifier) and a list of one or more errors pertaining to the business object.

In the case of dead message, the business payload is essentially metadata about an actual business message that was marked as being "dead". Thus any Error being reported can only use the information available as part of the "Obituary" business object in the dead message to communicate a specific error.

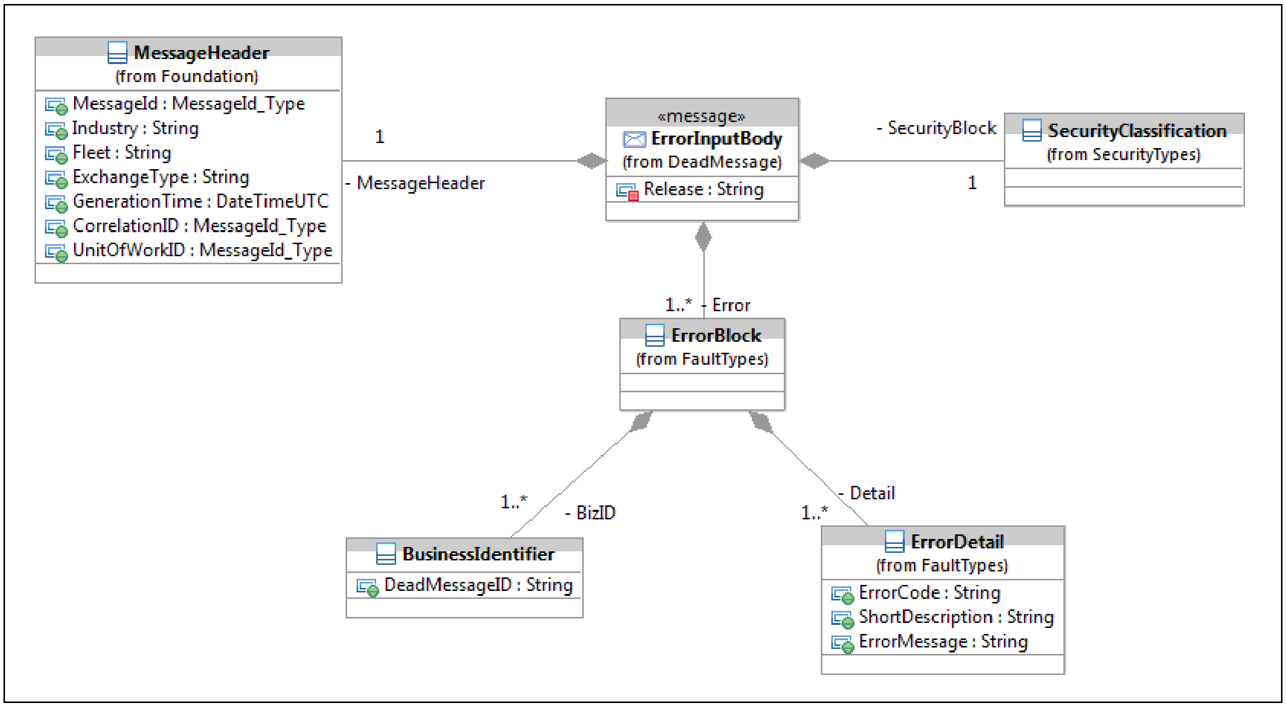

Figure 7-5 Dead Message Fault Input Body

MessageHeader and SecurityClassification are mandatory as in this scenario (after Type 1 validation has passed) the input message is well-formed.

The Message Header has a new unique Message ID.

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

8 Service Operation Details

8.1 Detailed Operation Characteristics – SendObituary()

Canada EDE or Industry will invoke the other party's Dead Message service through this operation. The input will consist of an Obituary InputBody (as above).

Please refer to Service Interaction Model [Ref. 1] for definitions of the terminology used in the non-functional requirements section.

Please refer to Dead Message wsdl files for implementation details.

Detailed Operation Characteristics

Interface Definition	Description
Operation Name	Send Obituary
Operation Technical Name	SendObituary()
Operation Description	This operation is invoked by Canada EDE or Industry to send an obituary to the other party.
Target Operation Provider	Canada EDE and Industry
Target Operation Consumer	Canada EDE and Industry
Properties	*Request/Response* message exchange pattern.
Input Message Definition	Please refer to Operation Message Model Section 7.1 DeadMessage Input Body for details.
Output Message Definition	Please refer to Operation Message Model Section 7.2 DeadMessage Output Body for details.
Fault Definition	Please refer to Section 7.3 DeadMessage Fault Body for details. Please refer to Service Interaction Model [Ref. 1] for Type 1 faults.

Non Functional Requirements

Non Functional Requirements/Technical Details	
Frequency	Periodic basis – Every Hour on the hour(0000, 0100, 02000, .., 2300)
Peak Throughput Time	Expected to be off-peak, e.g. after ZULU 01:00 and before ZULU 11:00 ZULU

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

Non Functional Requirements/Technical Details	
Peak Throughput Volume	In a worst case scenario (e.g. network failure) there could be one invocation of *SendObituary*() for every message from other business domains (e.g., Supply, Maintenance History, etc.)
Payload Size	A single obituary will be ~5K. The number of obituaries per service invocation is variable.
Attachments	None
Attachment Size	N/A
ACK Time Interval	Nominal value is 5 minutes – to be confirmed between Canada and Industry on a per-fleet basis.
Retry Time Interval	Nominal value is 15 minutes – to be confirmed between Canada and Industry on a per-fleet basis.
Number of Retries	Nominal value is 3 retries – to be confirmed between Canada and Industry on a per-fleet basis.
Biz. Response Time Interval	N/A
Time to Live Span	N/A
Service Op Availability	During core processing hours. The specific period will be defined during later phases of service realization 95% available uptime is the goal of the service
Downtime Requirements	The service cannot be used during established maintenance windows, which is currently expected to be for about 2 hours per week. The unavailability window may be accumulated and invoked during major maintenance periods, but ensuring that the overall availability of the service is still maintained.
Dead Message Handling	Alternative communication channel applies to report that this operation is not available when Industry cannot successfully send Dead Message business objects to Canada EDE. See Service Interaction Model [Ref. 1].

8.2 Detailed Operation Characteristics – SendObituaryACK()

Canada EDE or Industry will invoke the other party’s Dead Message service through this operation. The input will consist of an ObituaryACK InputBody (as above).

Note: This is not required to be used unless determined by industry and Canada that the “Dead Message” service operates in a pessimistic model.

Canada will accept an optimistic model in that the successful technical delivery of the message is adequate for this service to enter into a completed state from a message delivery perspective.

Please refer to Service Interaction Model [Ref. 1] for definitions of the terminology used in the non-functional requirements section.

Please refer to Dead Message wsdl files for implementation details.

Detailed Operation Characteristics

Interface Definition	Description
Operation Name	Send Obituary Ack
Operation Technical Name	SendObituaryACK()
Operation Description	This operation is invoked by Canada EDE or Industry to send an acknowledgement of obituaries to the other party.
Target Operation Provider	Canada EDE and Industry
Target Operation Consumer	Canada EDE and Industry
Properties	*Request/Response* message exchange pattern.
Input Message Definition	Please refer to Operation Message Model Section 7.4 DeadMessage ACK Input Body for details.
Output Message Definition	Please refer to Operation Message Model Section 7.2 DeadMessage Output Body for details.
Fault Definition	Please refer to Section 7.3 DeadMessage Fault Body for details. Please refer to Service Interaction Model [Ref. 1] for Type 1 faults.

Non Functional Requirements

Non Functional Requirements/Technical Details	
Frequency	Periodicity dependant on the “SendObituary()” service being invoked
Peak Throughput Time	Same as SendObituary() operation.
Peak Throughput Volume	Same as SendObituary() operation
Payload Size	1kB.
Attachments	None

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

Non Functional Requirements/Technical Details	
Attachment Size	N/A
ACK Time Interval	Nominal value is 5 minutes – to be confirmed between Canada and Industry on a per-fleet basis.
Retry Time Interval	Nominal value is 15 minutes– to be confirmed between Canada and Industry on a per-fleet basis.
Number of Retries	Nominal value is 3 retries – to be confirmed between Canada and Industry on a per-fleet basis.
Biz. Response Time Interval	N/A
Time to Live Span	N/A
Service Op Availability	During core processing hours. The specific period will be defined during later phases of service realization 95% available uptime is the goal of the service
Downtime Requirements	The service cannot be used during established maintenance windows, which is currently expected to be for about 2 hours per week. The unavailability window may be accumulated and invoked during major maintenance periods, but ensuring that the overall availability of the service is still maintained.
Dead Message Handling	Alternative communication channel applies to report that this operation is not available when Industry cannot successfully send Dead Message business objects to Canada EDE. See Service Interaction Model [Ref. 1].

8.3 Detailed Operation Characteristics – SendObituaryError()

Canada EDE or Industry will invoke the other party’s Dead Message service through this operation. The input will consist of an Obituary Error InputBody (as above).

Please refer to Service Interaction Model [Ref. 1] for definitions of the terminology used in the non-functional requirements section.

Please refer to Dead Message wsdl files for implementation details.

Detailed Operation Characteristics

Interface Definition	Description
Operation Name	Send Obituary Error
Operation Technical Name	SendObituaryError()

The information being provided is to illustrate the model that exists for business processes and information exchange within the Performance Based Contracting (PBC) solution for the Department of National Defence. The information is provided to facilitate an understanding of the business architecture and the solution architecture that exist for the PBC program. The content is not intended to reflect the end state specifications for all of the PBC EIE related services.

.

Interface Definition	Description
Operation Description	This operation is invoked by Canada EDE or Industry to send an obituary to the other party.
Target Operation Provider	Canada EDE and Industry
Target Operation Consumer	Canada EDE and Industry
Properties	*Request/Response* message exchange pattern.
Input Message Definition	Please refer to Operation Message Model Section 7.5 DeadMessage ErrorInput Body for details.
Output Message Definition	Please refer to Operation Message Model Section 7.2 DeadMessage Output Body for details.
Fault Definition	Please refer to Section 7.3 DeadMessage Fault Body for details. Please refer to Service Interaction Model [Ref. 1] for Type 1 faults.

Non Functional Requirements

Non Functional Requirements/Technical Details	
Frequency	Dependant on the SendObituary() call frequency
Peak Throughput Time	Same as SendObituary() operation.
Peak Throughput Volume	Same as SendObituary() operation.
Payload Size	1kB.
Attachments	None
Attachment Size	N/A
ACK Time Interval	Nominal value is 5 minutes – to be confirmed between Canada and Industry on a per-fleet basis.
Retry Time Interval	Nominal value is 5 minutes – to be confirmed between Canada and Industry on a per-fleet basis.
Number of Retries	Nominal value is 3 retries – to be confirmed between Canada and Industry on a per-fleet basis.
Biz. Response Time Interval	N/A
Time to Live Span	N/A

Non Functional Requirements/Technical Details	
Service Op Availability	During core processing hours. The specific period will be defined during later phases of service realization 95% available uptime is the goal of the service
Downtime Requirements	The service cannot be used during established maintenance windows, which is currently expected to be for about 2 hours per week. The unavailability window may be accumulated and invoked during major maintenance periods, but ensuring that the overall availability of the service is still maintained.
Dead Message Handling	Alternative communication channel applies to report that this operation is not available when Industry cannot successfully send Dead Message business objects to Canada EDE. See Service Interaction Model [Ref. 1].

8.4 Service Bindings

8.4.1 SOAP Over http

The implementation of this service will use a Simple Object Access Protocol (SOAP) binding with document style messages and http transport.

The business objects (Section 6) are bound to the SOAP Body element. The SOAP Header is used for EIE adopted WS-* standards-based elements (e.g., WS_Security assertions) and, typically, MessageHeader and SecurityMarkings elements. See the Dead Message Service WSDL file for the precise binding.

In this binding the http response is used for each operation’s technical response (i.e., output or fault messages).

8.4.2 SOAP Over JMS

Not currently supported.

9 Definitions, Acronyms, Abbreviations

Term	Description
CM	Configuration Management
CMMS	Canada Maintenance Management System
CSS	Canada Supply System
DND	Department of National Defence
DRMIS	Defence Resource Management Information System
EDE	Electronic Data Exchange
EIE	Electronic Information Environment
EMR	Equipment Master Record
FLOC	Functional LOCation
ISS	In Service Support
ISSCF	In Service Support Contracting Framework
MER	Master Equipment Record
MPN	Manufacturer Part Number
MMR	Material Master Record
NATO	North Atlantic Treaty Organization
NSN	NATO Stock Number
PBC	Performance Based Contracting
SOAP	Simple Object Access Protocol
UTC	Coordinated Universal Time
WS	Weapon System
WSDL	Web Service Definition Language

10 Document History

Revision Number	Description	Date
0.1	Initial Draft	28 November, 2011
2.0	Update based on internal review and added triggers	31 January, 2012
2.1	Clarified some non-functional requirements. Minor wording changes for clarity.	22 March, 2012
2.2	Updated BizID to be concrete definition.	8 January 2013
2.3	Removed PROTECTED-A markings from document and add proviso to page footer.	10 June 2013
2.4	Adapted to reference PBC as well as ISSCF	09 November 2015

11 Annex

This example xml file shows an obituary message (with Message ID “OBITUARY-1234-5678”) which is sent from Industry to Canada to report that a prior Part Demand message (with Message ID “DEMAND-ABCD-EFGH-0123”) was declared dead (by Industry).

This example is for illustrative purposes and no content (URLs, namespace, elements, …) is definitive.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<tns:InputBody tns:PublishDate="2011-12-05" tns:Release="1.0"
xmlns:p="http://eie.isscf.forces.gc.ca/EIE/2012/01/31/Foundation"
xmlns:tns="http://eie.isscf.forces.gc.ca/EIE/2012/01/31/DeadMessage"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://eie.isscf.forces.gc.ca/EIE/2012/01/31/DeadMessage ../DeadMessage.xsd ">
  <p:MessageHeader>
    <p:MessageId>OBITUARY-1234-5678</p:MessageId>
    <p:GenerationTime>2001-12-31T12:00:00</p:GenerationTime>
    <p:ExchangeType>9101</p:ExchangeType>
    <p:Industry>0002</p:Industry>
    <p:Fleet>0002</p:Fleet>
  </p:MessageHeader>
  <p:SecurityBlock>
    <p:Classification>UNCLASSIFIED</p:Classification>
    <p:Caveat>ITAR</p:Caveat>
    <p:Releasable>CAN</p:Releasable>
  </p:SecurityBlock>
  <tns:Obituary>
    <tns:DeadHeader>
    <p:MessageId>DEMAND-ABCD-EFGH-0123</p:MessageId>
    <p:GenerationTime>2011-12-01T12:01:02</p:GenerationTime>
    <p:ExchangeType>3003</p:ExchangeType>
    <p:Industry>0002</p:Industry>
    <p:Fleet>0002</p:Fleet>
    </tns:DeadHeader>
    <tns:DeadSecurityBlock>
    <p:Classification>UNCLASSIFIED</p:Classification>
    <p:Caveat>ITAR</p:Caveat>
   <p:Releasable>CAN</p:Releasable>
</tns:DeadSecurityBlock>
    <tns:LastAttempt>2011-12-01T13:01:02</tns:LastAttempt>
    <tns:AttemptedURL>http://eie.industry.com/eie/supplymateriel/partdemand</tns:AttemptedURL>
    <tns:AttemptedService>PartDemand_Industry_Service</tns:AttemptedService>
    <tns:AttemptedOperation>SendPartDemand</tns:AttemptedOperation>
  </tns:Obituary>

</tns:InputBody>
```