THROUGH LIFE SUPPORT STANDARD

TLSS Data Exchange Specification Development Methodology

LSC REFERENCE: ECSMODTLSS0601.046

EUROSTEP REFERENCE: Eurostep.ESUKPC09.000136

AUTHORS	CHECKED	APPROVED
Brad Harris - Eurostep Colin Tweedie – LSC Group	Rob Bodington – Eurostep Tim Turner – LSC Group	Paul Clark – LSC Group Simon Dick – Eurostep
ISSUE	**DATE**	**CLIENT**
3.0	01 November 2007	TLSD Policy Coordination - TLSS Project Manager

LSC Group Ltd.
Lincoln House
Fradley Park
Lichfield
Staffordshire
WS13 8RZ
United Kingdom

Eurostep Ltd.
Cwttir Lane
St. Asaph
Denbighshire
LL17 0LQ
United Kingdom

Amendment Record

Issue	Date	Summary of changes
1.0	2007-01-29	Initial release
2.0	2007-03-27	Updated following review comments from TES and teleconference with TES on Thursday 15th February.
3.0	2007-11-01	Updated to reflect changes in the DEXlib development environment and to align with the 'Guidance on writing DEXlib TLSS Data Exchange Specifications' document.

Distribution

This document has been distributed to:

Copy No	Media	Location	Title
1	PDF File		TLSD Policy Coordination - TLSS Project Manager

Contents

Figures

1 Introduction

This document defines the TLSS Data Exchange Specification development methodology.

It has been produced through the work undertaken by LSC Group and Eurostep Limited for the UK MoD (TES ILS Policy and Co-ordination Group) under their Task Authorisation Form (TAF) 0601. The requirement was to develop a methodology for developing the Data Exchange Specifications for the Through Life Support Standard (TLSS).

This deliverable is one of four from this project, as stated in Eurostep TAF 06 01 and LSC TAF 06 01, the other three being:

- The overall Project Report [1]
- TLSS Data Exchange Specification Development Infrastructure and Tool URD [2]
- Example completed TLSS Data Exchange Specifications that can be exchanged in accordance with ISO 10303-239 (PLCS).

1.1 Background

The components of TLSS are described in the following diagram:

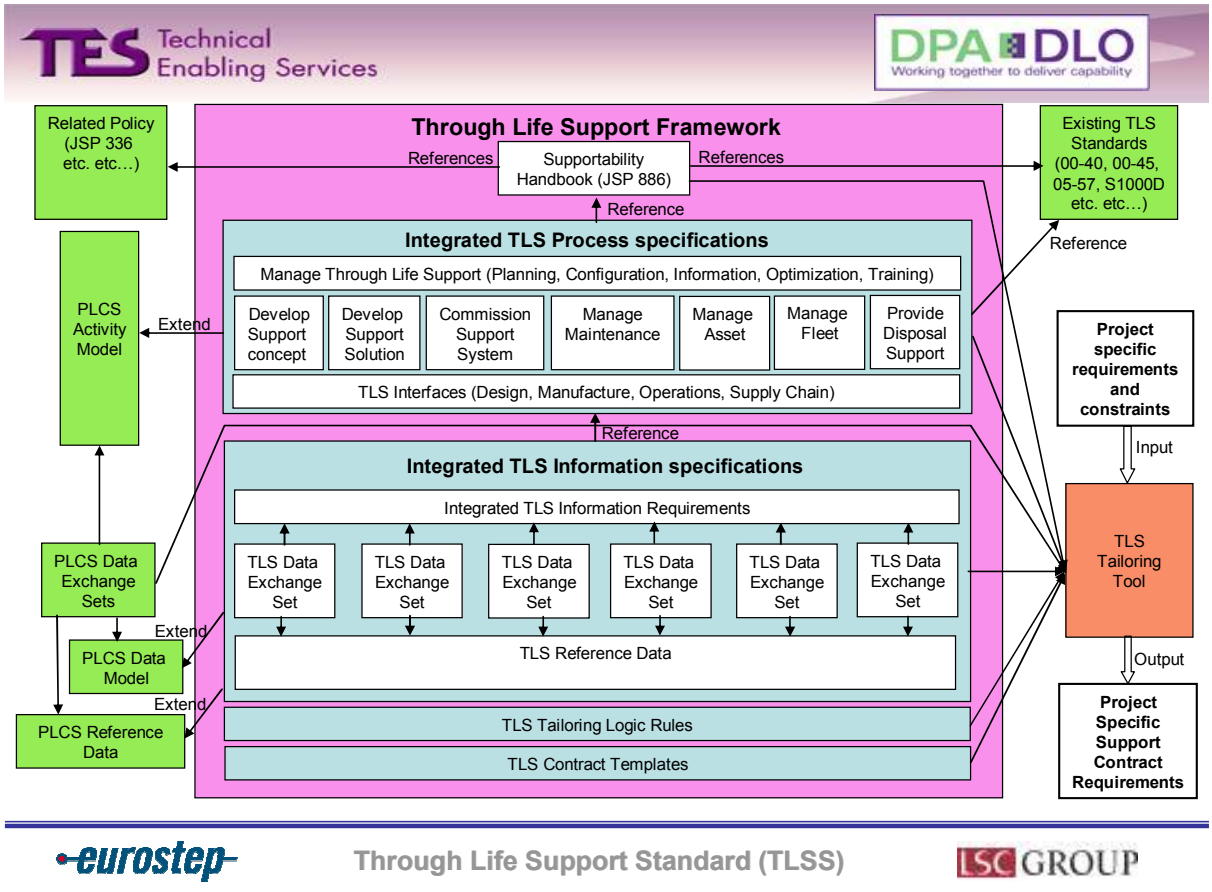

Figure 1: The TLSS Architecture

The key output from the application of TLSS will be a set of project specific requirements (in terms of process requirements and data exchange requirements), that can be used in the generation of through life support contracts that can be placed by one organization on another – typically from MoD to Prime Contractor. The core components of the TLSS architecture that enable this output, are the through life support process model and the associated information that is required to be exchanged between those processes.

Project specific tailoring, according to the methodology defined in [3] , will select applicable processes and their corresponding input and output information flows. These information flows constitute a set of data exchange requirements.

In order to achieve coherence and consistency of all such data exchange requirements, an integrated set of TLSS Business Objects that satisfy those requirements, needs to be defined.

In order to be able to exchange those TLSS Business Objects in a coherent and consistent manner between the systems that produce and use such information, TLSS Data Exchange Specifications based on ISO 10303-239 (PLCS) need to be defined.

1.2 Objective

The objective of this document is to define the methodology for developing TLSS Data Exchange Specifications in a coherent and consistent manner.

1.3 Structure of the document

The remainder of this document is structured to define:

1. The nature of the key components of the TLSS architecture that are used in applying the methodology.
2. The methodology itself, in terms of:
    a. How to identify and select which process model information flows are potential TLSS Data Exchange Specifications. These may be individual information flows, or groups of information flows that are exchanged in a typical data exchange contract. See section 3.2.1
    b. How to define the information content of each information flow selected, in terms of TLSS Business Objects. See section 3.2.2
    c. How to link the TLSS Business Objects to the TLSS information requirements that they satisfy. See section 3.2.3
    d. How to document TLSS DEXlib Data Exchange Specifications that define an implementation mapping of the TLSS Business Objects to the ISO 10303-239 (PLCS) data model. See section 3.2.4[1]

[1] Note that it is recognized that the scope of TLSS is broader than that covered by ISO 10303-239 and therefore that some supply chain and document data exchange requirements will need to be satisfied by other standards such as OAGIS and ASD S1000D. For the time being however, this document concentrates on those data exchange requirements that can be satisfied by ISO 10303-239.

NOTE: The requirements for tools necessary for developing TLSS Data Exchange Specifications are described in detail in the TLSS Data Exchange Specification Development Infrastructure and Tool URD [2] .

2 Key Concepts

2.1 Introduction

This section describes the key components of the TLSS architecture that are used when employing the TLSS DEX Development Methodology, which is described in Section 3.

2.2 The TLSS Business Process Model Processes

The TLSS business process model captures the TLS processes independent of the organizations that carry them out. The purpose of the process model is to represent the sequence of the TLS processes throughout the CADMID[2] cycle. Some processes are decomposed hierarchically into a set of sub-processes.

The model has been developed in the Business Process Modeling Notation (BPMN), part of the Object Management group (OMG) standard and related to the Unified Modeling Language (UML).

2.3 The TLSS Business Process Model Information flows

The TLSS business process model information flows represent the information that is used by and generated by the TLS business processes and / or external business processes. Each information flow comprises:

- the set of information that is required to be exchanged between the processes.
- the identification of both the source and destination processes.

In the same way that the TLSS processes may be decomposed hierarchically, so may the information flows that are exchanged between them. The higher level information flows are aggregations of the lower level flows[3] (rather than being abstractions of them).

TLSS information flows are defined as instances of named UML classes and are linked to processes by directional associations.

These UML classes are defined in terms that define the scope of the information that needs to be exchanged, but they do not define that information in sufficient detail to enable implementation using ISO 10303-239 (PLCS).

2.4 Integrated TLSS Information Requirements

The integrated TLSS information requirements have been derived from current standards (such as DEF STAN 00-60 and AECMA S2000M) and system specifications (such as UMMS and DRACAS) that are used by the MoD for various aspects of through life support.

[2] See Glossary

[3] Some may be at the same level

These information requirements represent a logical view of through life support information. They do not represent a transactional / message / information exchange view of that information in the way that the process model information flows do.

The information requirements are defined in a systems engineering requirements database (the TLSS Information Requirements Repository) whose structure is specified by ISO 10303-233.

2.5 TLSS Business objects

In order to be able to define TLSS Data Exchange Specifications based on ISO 10303-239 that can be implemented by software systems, it is necessary to specify the information structure / content of TLSS process model information flows in sufficient detail that enables them to be mapped to the PLCS data model, either directly, or via existing Templates[4]. This requires that the reader should be familiar with the TLSS Process Model, UML modelling, the PLCS data model and the PLCS Templates.

The TLSS construct for achieving this is the Business Object. This is an information object that represents the information that is used within a business, such as the information that is exchanged in MoD business transactions.

A TLSS process model information flow comprises one or more TLSS business objects. This set of business objects is represented by a UML Composite Structure (diagram). Example diagrams are shown in Figure 2 and Figure 3.

A TLSS business object is used by one or more TLSS process model information flows.

A TLSS business object may be derived directly from and/or satisfy:

1. A single TLSS information requirement.
2. A group of TLSS information requirements.
3. The class that defines a TLSS process model information flow.

Each TLSS business object is represented by a UML class and may have one or more attributes as well as one or more relationships to other TLSS business objects. Attributes and relationships may be mandatory or optional.[5]

Where the business object is used in an information flow, and where there are process specific constraints that apply to that business object, for example, where it is known that certain optional elements (attributes or relationships) of the business object need to be mandatory within this process context, then this can be specified by making an instance of the class on the composite structure diagram and showing which optional elements are required.

However, for flexibility, it will often be the case that it is better not to define such constraints within the TLSS Data Exchange Specification, but to allow the project specific tailoring process to define them, by the tailoring in of required optional elements, as specified in [3] .

NOTE: The consequence of organizing the TLSS information classes as described above, is that TLSS requires that the two types of UML classes are distinguished - using UML stereotypes for:

1. The information flow classes.
2. The business object classes.

[4] Templates are described in detail in sections 3.3.3 and 3.3.4.

[5] The way in which Business Objects are documented in DEXlib is referenced in Section 3.3.5.and [5].

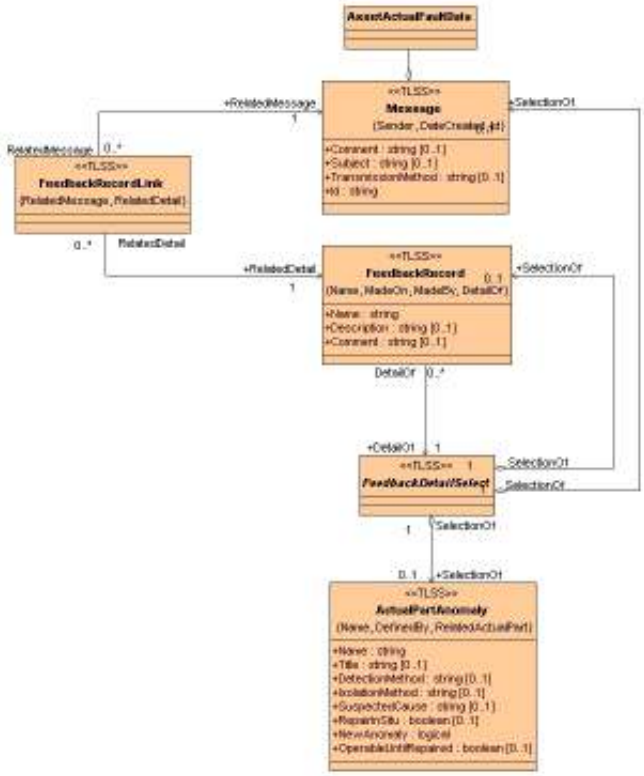

Figure 2: Feedback business objects represented as a class diagram

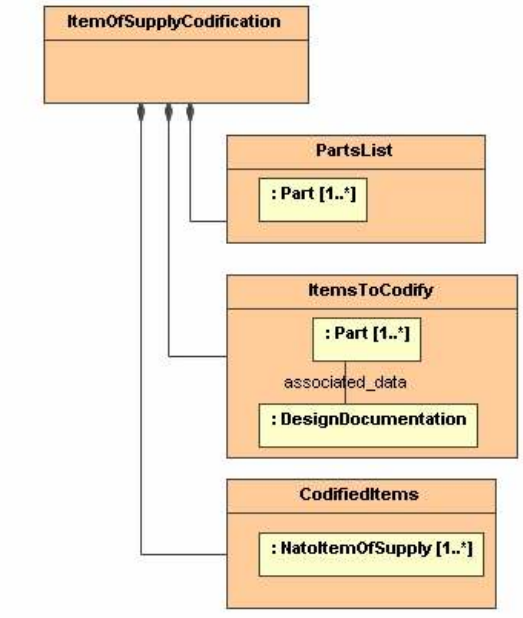

Figure 3: Codification Business objects represented as a composite structure diagram

2.6 PLCS Data Model, Templates and Reference Data

The implementation layer of TLSS Data Exchange Specifications is defined by a mapping from the TLSS business objects to the ISO 10303-239 (PLCS) data model, templates and reference data

through TLSS Business Templates, as described in section 3.3.4. Implementation of TLSS data exchanges will then take the form of ISO 10303-21 or ISO 10303-28 file exchanges between systems

3 TLSS Data Exchange Specification Development Methodology

3.1 Overall approach

The overall approach to the development of TLSS Data Exchange Specifications is to:

1. Analyse the requirements top-down, from the TLSS process model and information flows down to the PLCS implementation layer.
2. Build the DEXlib documentation of the specification bottom-up, from the PLCS data model and templates to the process model requirements.

Consequently, the TLSS business objects and their DEXlib counterparts, act as the focus for bringing the two strands together.

3.2 Top-down analysis

The relationship between the concepts described in Section 2 is represented in Figure 4. In the diagram the stages of the methodology are defined by stages 1 to 4. In this methodology, the TLSS processes are represented by the yellow rectangles labelled as 'Process A', 'Process B' and 'Process C' respectively. These are sequential processes that have information flows labelled: 'InformationFlowA' and 'InformationFlowB'. The process for analysing and selecting the information flows is described in more detail in section 3.2.1.

Once the information flows are identified, there is a need to determine the corresponding business information that will be exchanged (contractually) between MoD and its industrial contractors, subcontractors and suppliers. The key objective of identifying the business objects is to ensure that MoD terminology and the right level of detail of the information is captured. The process of identifying the business objects is described in more detail in section 3.2.2.

Having identified the business objects, these are then related to the information requirements (stage 3) and the PLCS data model (stage 4). These processes are described in more detail in sections 3.2.3 and 3.2.4 respectively.

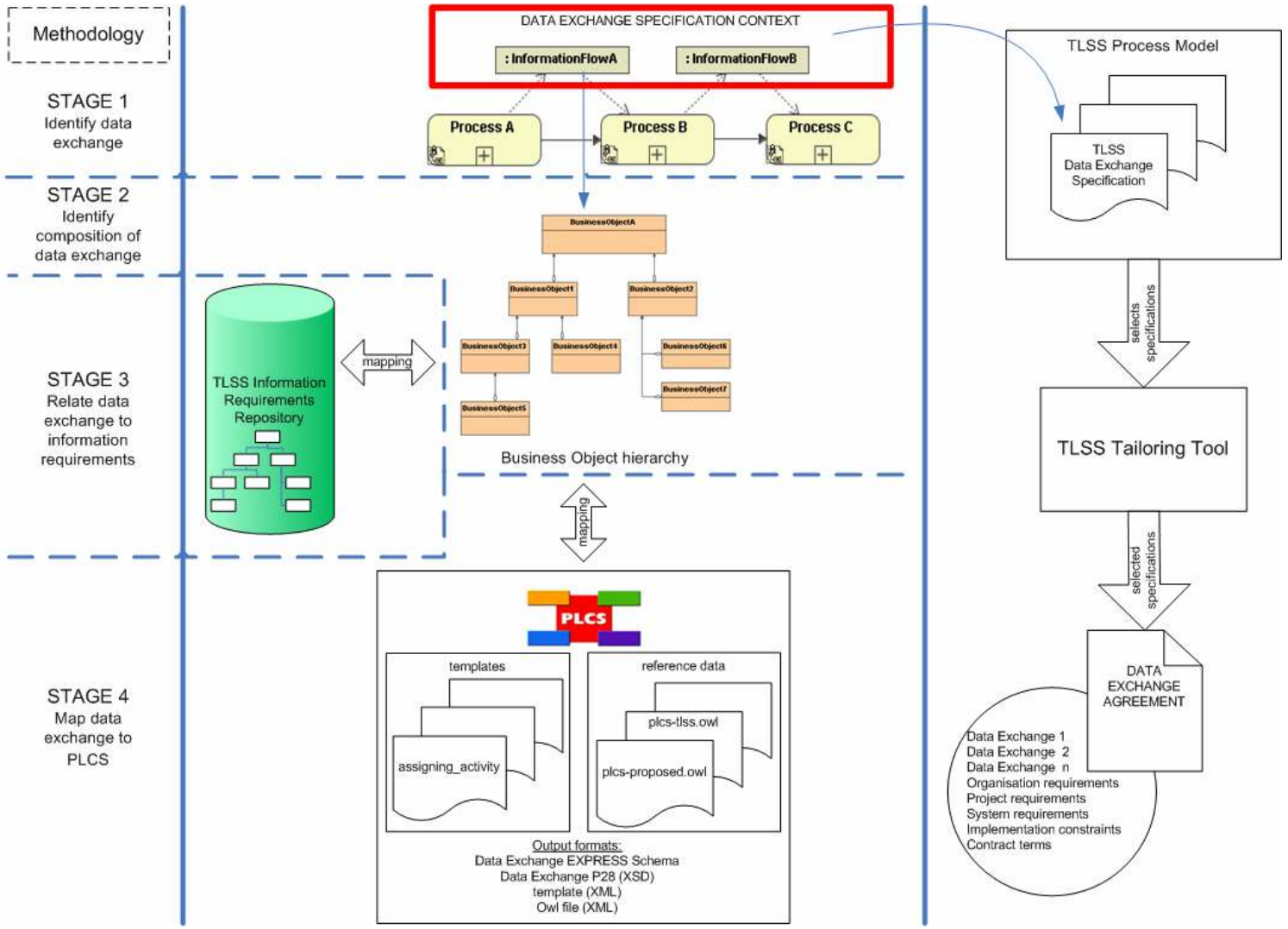

Figure 4: Methodology stages and relationships between the TLSS and PLCS components

In summary, the data exchange development process is defined in 4 stages:

1. Identification;
2. Detailed composition;
3. Relating to the TLSS information requirements;
4. Mapping to ISO 10303 AP239 (PLCS).

This process is illustrated in Figure 5.

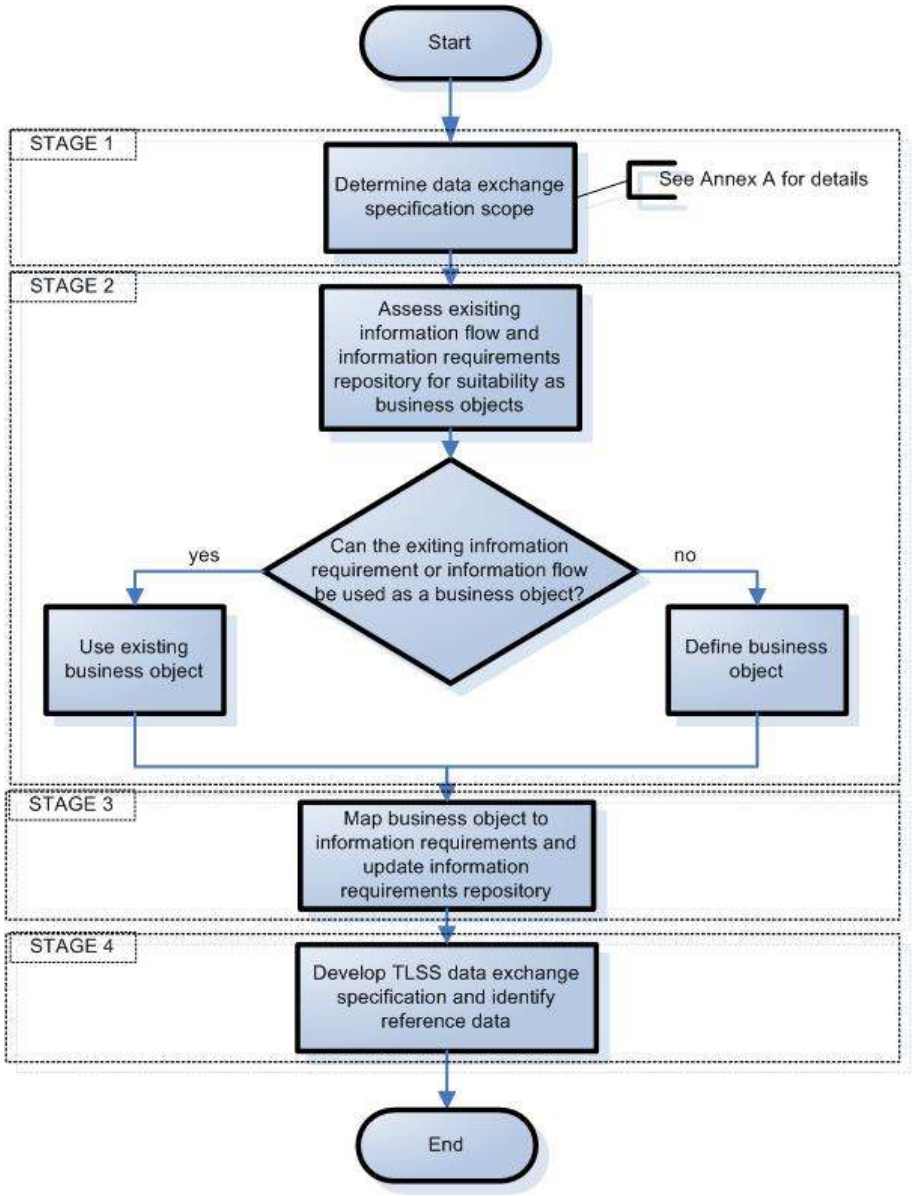

Figure 5: Data Exchange Specification development workflow

3.2.1 STAGE 1 – Identify the requirement for a TLSS Data Exchange[6]

Objective: Identify requirements for a TLSS Data Exchange Specification

Input: TLSS Process Model (format: MagicDraw UML model).

Method: Utilize the decision flow chart and record the results in the data exchange identification spreadsheet. Apply domain knowledge in order to group identified information flows for the data exchange. (Refer to Annex A).

Output: Data exchange scoping document.
This document identifies the selected information flow(s), and their associated processes, and records the rationale for selecting and grouping them in the Data Exchange Specification. Scope statements that explicitly define data that is both in scope and out of scope should be added where possible.

3.2.2 STAGE 2 – Identify the composition of the TLSS Data Exchange

Objective: Identify the TLSS business objects that constitute the information flow(s) of the identified TLSS Data Exchange Specification and define them in sufficient detail to enable mapping to the PLCS data model and templates.

Inputs: TLSS Process Model (format: MagicDraw UML file).

Data exchange scoping document (from STAGE 1).

Method: With respect to the data exchange scoping document, apply domain knowledge and knowledge of the TLSS information requirements to identify the required business objects, including their attributes and relationships with other business objects.

When new business objects are required, these shall be defined within the context of the existing business objects that have been derived from the process model information flows, to ensure that duplicate, redundant and inconsistent business objects are not proliferated[7].

Output: UML composite structure diagram that defines the composition of the data exchange in terms of the required business object classes and / or instances.

3.2.3 STAGE 3 – Relate the TLSS Data Exchange business objects to the TLSS information requirements

Objective: Relate the TLSS business objects to the TLSS information requirement(s) that they satisfy.

Input: TLSS data exchange UML composite structure diagram.

Method: Analyse the TLSS information requirements repository to determine if there are matches with the business object and its attributes and relationships. If there are, document the relationship from the business object to the relevant requirements in the repository. If not, register that a change is required to the TLSS information requirements repository.

---

[6] See reference [6]

[7] Note: During the period October – December 2007, an integrated business object model will be created under the TLSS / LCIA convergence work and the result will be known as the Common Business Information Structure (CBIS). It will proceed in parallel with the development of the first set of TLSS DEXs (Product information related DEXs needed for ACA) and there will be a two way exchange of information between the two projects to prevent divergence.

Once complete, add hyperlinks from the UML elements to the TLSS Information Requirements Repository to aid navigation between the two environments.
(Note that the hyper linking mechanism described above is subject to verification as the TLSS architecture matures. Refer to the data exchange development URD [2] for a definition of the requirement.)

Output: Record of TLSS business objects related to TLSS information requirements.

Record of need to update the TLSS information requirements repository.

3.2.4 STAGE 4 – Document the TLSS Data Exchange in DEXlib.

Objective: To document the TLSS Data Exchange Specification as a DEXlib Business DEX.

Input: Data exchange scoping document from Stage 1.

UML composite structure diagram from Stage 2.

Method: Develop a TLSS business Data Exchange Specification utilising the documentation and instantiation template facilities in DEXlib.

Once complete, add hyperlinks from the UML elements to the DEXlib elements to aid navigation between the two environments.

(Note that the hyper linking mechanism described above is subject to verification as the TLSS architecture matures. Refer to the data exchange development URD [2] for a definition of the requirement.)

Outputs: TLSS business Data Exchange Specification.

3.3 Bottom-up production of DEXlib documentation

3.3.1 Approach

TLSS Data Exchange Specification documentation can be built in a bottom-up manner in the following order:

- PLCS templates
- Business templates, including the Business object that they define.
- Business data exchange specification

These can be seen as components that can be related together to form the data exchange specification. These relationships are shown in Figure 6.

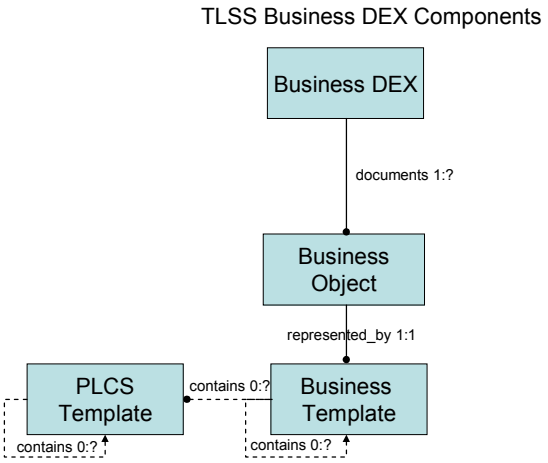

Figure 6: Relations between the Business DEX Components

Note: The overall relationships between these components and those of the TLSS architecture are illustrated further in Figure 7 below, although for documentation purposes, each Business Object is defined within its corresponding Business Template, as described in Section 3.3.4 below.

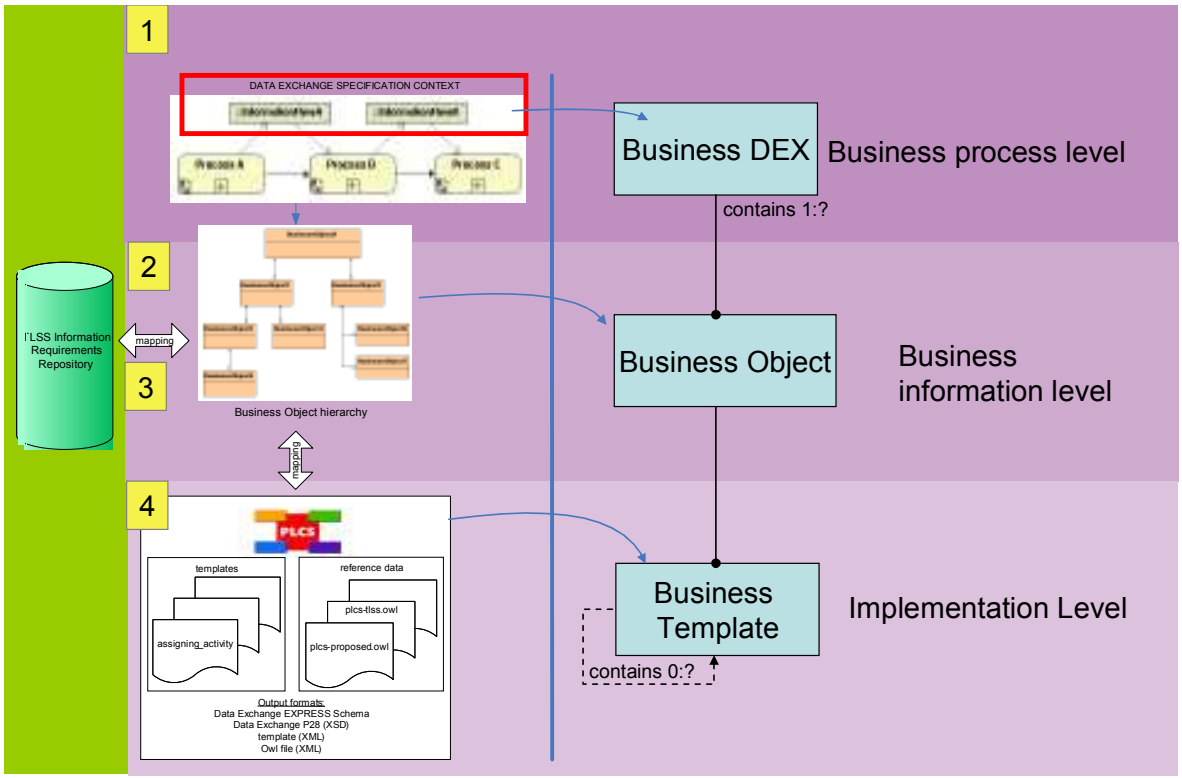

Figure 7: Relationships of TLSS Components as used in the OASIS Architecture

3.3.2 Workflow

The following diagram illustrates the overall workflow for developing TLSS Data Exchange Specifications. Note that there are review and approval cycles defined for each DEX and for each required template, as described in [5] and [4].

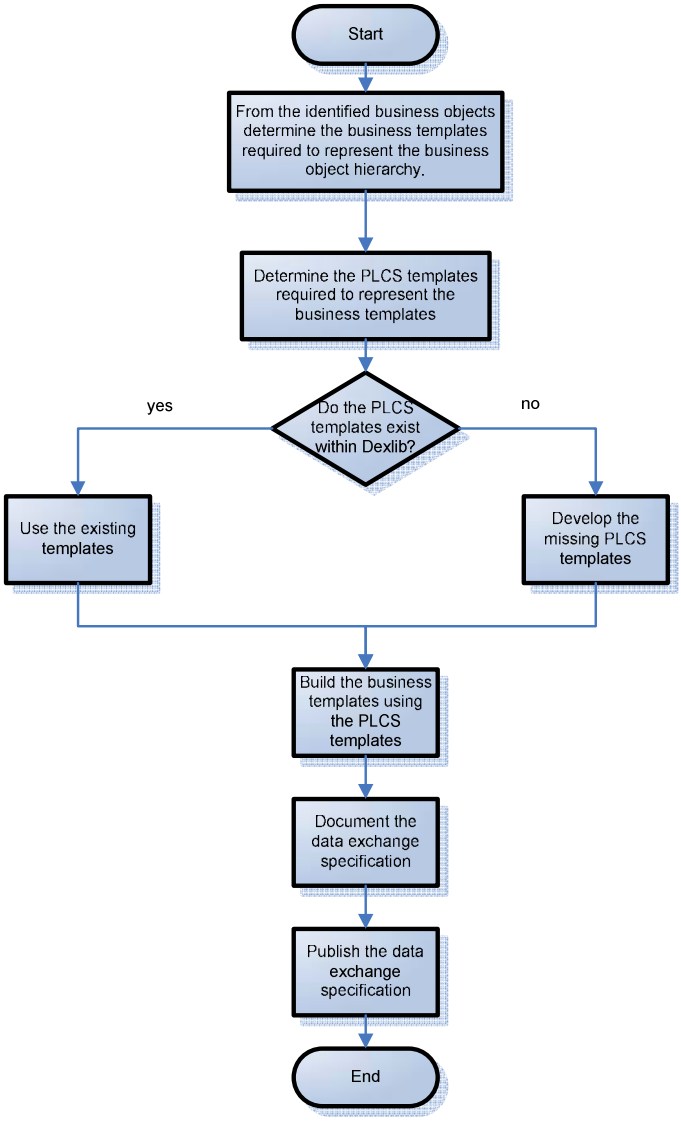

Figure 8: Workflow for developing business objects in DEXlib

3.3.3 PLCS templates

Any construct required by the TLSS Data Exchange Specification that will be generically useful in other contexts, should be developed as a PLCS Template. These are either developed based on existing templates or are new developments.

The structure and content of PLCS Templates, together with the process for developing them, is documented in detail in the DEXlib help pages.[8]

[8] These are currently undergoing review and update.

3.3.4 Business templates

Any construct required by the TLSS Data Exchange Specification that will not be generically useful in other contexts, should be developed as a Business Template. These are either developed based on existing PLCS Templates and/or Business Templates, or they are new developments.

The structure and content of Business Templates, is documented in detail in the DEXlib help pages.[9]

When designing templates, as many of the requirements as possible should be satisfied by the generic PLCS templates. Where a relevant generic PLCS-template does not exist, thought should be given to creating one before generating TLSS-specific templates. Business templates should be built based on PLCS templates as far as possible. In some cases they specialize the PLCS templates and in some cases they combine/wrap a number of PLCS templates into a new one. Business templates have a similar structure to the PLCS templates; however they represent a domain specific business view. The description section should include the business use of the template.

Business Templates are the mechanism by which each TLSS business object is mapped to the PLCS Templates, entities, relationships and attributes.

Specific guidance on writing TLSS Business Templates is defined in [4].

3.3.5 Business DEX

Each TLSS Data Exchange Specification should be documented as a DEXlib Business DEX.

Business DEXs are broadly similar to OASIS DEXs [10].

Specific Guidelines for developing TLSS DEX's can be found in [5].

[9] Note that OASIS DEXLib Help files are currently undergoing review and update.

[10] See http://www.plcs-resources.org/plcs/dexlib/dex_index.htm and click on the info menu item

The final DEXlib document will therefore be structured as follows:

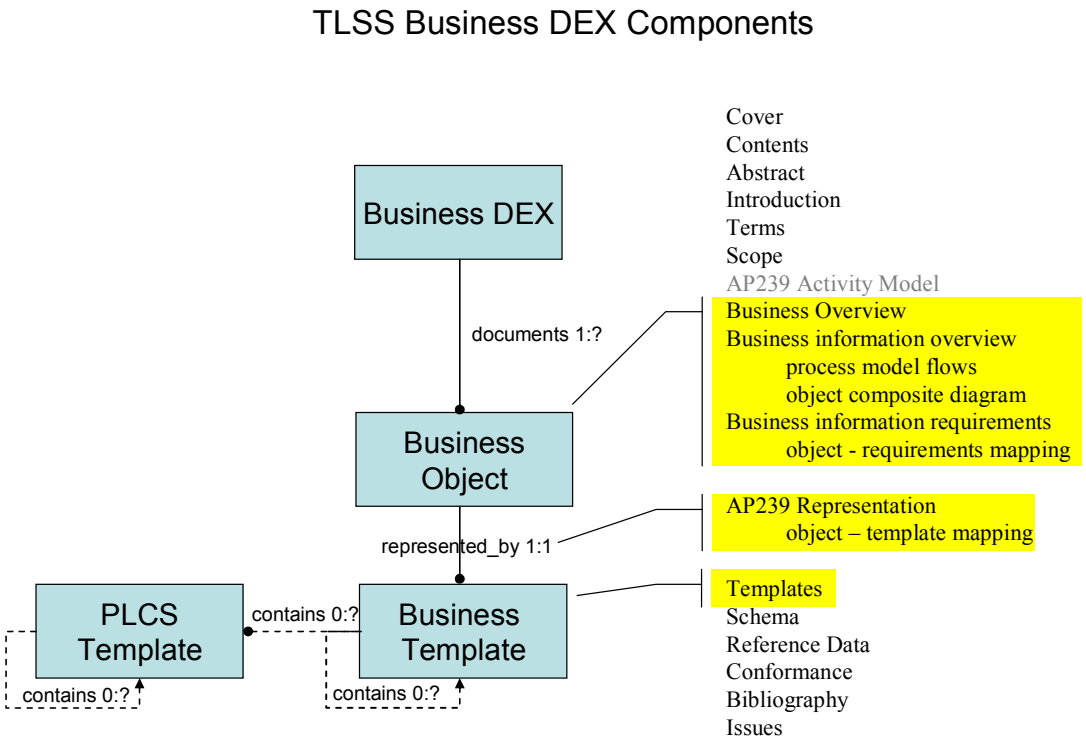

Figure 9: The form of the DEXlib Business DEX document structure

4 Document References

[1] TLSS Data Exchange Specification Development Project Report

[2] TLSS Data Exchange Specification Development Infrastructure and Tool URD.

[3] TLSS Tailoring Methodology

[4] Guidance on writing DEXLib TLSS Business Templates

[5] Guidance on writing DEXLib TLSS Data Exchange Specifications.

[6] TLSS Data Exchange Identification Project Report

Annex A. Guidelines: Identifying the requirement for a TLSS data exchange

A.1 General

Stage one identifies the data classes that will be the basis for the data exchange specifications. The TLSS process model is the starting point for this stage and every instance of the use of a data class has been identified and a spreadsheet populated with this information. Each line on this spreadsheet identifies the data class and the process with which it is associated; if one instance of a data class is associated with more than one process there will be separate lines for each data class/process association.

A.2 Identification process

The spreadsheet provides a means of ensuring that no processes and data flows can be ignored, however it is necessary to understand the logic used in deciding which of these need to be considered as part of a data exchange.

- understand the function of each process
- identify the process information flows that are:
  - input;
  - output;
  - inter process.
- understand the information content of each:
  - read the natural language definition
  - understand the source & target processes
  - focus on the information needs of the target process

A.3 Identification logic (see Figure 10)

Each element data class occurrence listed in the populated spreadsheet is subjected to a logical process which takes the form of the following questions:

Question	Rationale
1. Is the flow an information object?	Physical objects cannot be a Data Exchange Specification.
2. Is the flow largely/formally structured?	Unstructured information cannot be a Data Exchange Specification
3. Is the flow likely to be exchanged across a contract boundary?	Internal movement of information does not require a Data Exchange Specification
4. Is the flow likely to be passed between IT systems?	The Data Exchange Specifications are selected primarily on the need to communicate between IT systems.
5. Is the info flow likely to be exchanged in a higher aggregation of info exchange?	This question establishes whether the data class can be incorporated with others to form a consolidated Data Exchange Specification
Prioritization criteria: How frequent is the flow likely to be?	Data Exchange Specification development for low frequency data exchanges would not be cost-effective. Judgement to be used with regard to the value of "frequency" to be used. This judgement is to be recorded with the decision.

To be included as a candidate for inclusion in a Data Exchange Specification the answers to all these questions will be yes, with the exception that, if the answer to [Q5] is negative, there may well be a need for the data exchange at a lower level (e.g. failure feedback and request for codification).

The data exchange identification decision flow chart (Figure 10) is used in the context of each data class occurrence on a business process diagram within the TLSS Process Model. The 13 parts of the TLSS process model will each be analyzed by a team led by the data modeller responsible for that processes.

Data Class Occurrence Classification Logic

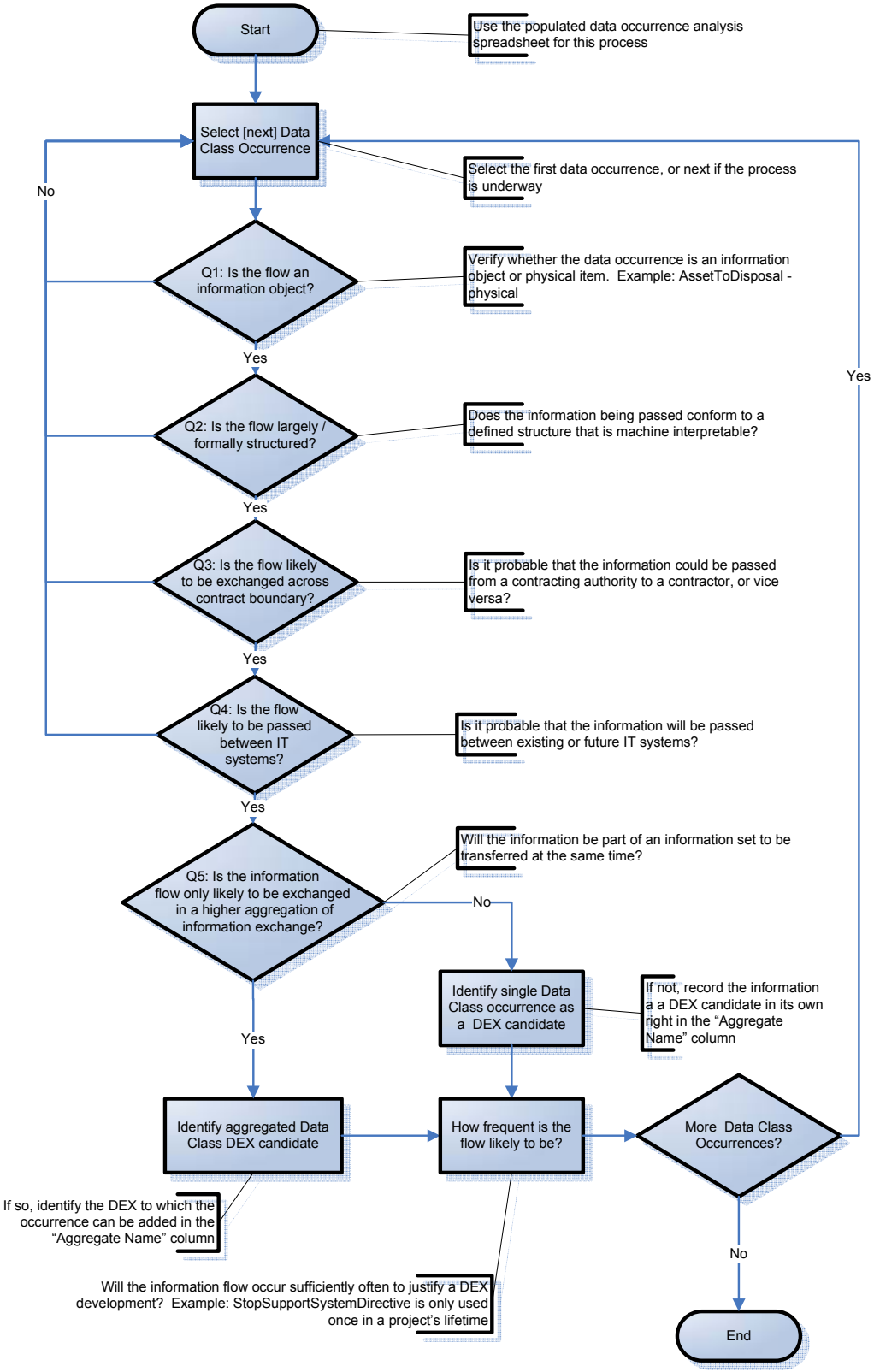

Figure 10: Data exchange identification decision flow chart

Annex B. Glossary

BPMN	Business Process Modelling Notation, an OMG standard for representing business processes. See "Business Process Modeling Notation (BPMN) Specification", OMG Final Adopted Specification, dtc/06-02-01, February 2006[11].
CADMID	Project life cycle phases used by UK MoD: concept, assessment, development, manufacture, in-service and disposal (or termination in the case of a service, hence CADMIT).
DEXlib	The **d**ata **ex**change **lib**rary provides information concerning the OASIS PLCS Data EXchange Specifications and related technology. DEX Specifications establish structured data exchange and sharing to support complex engineered assets throughout their total life cycle. These Data Exchange Specifications are based upon the ISO 10303 (STEP) Application Protocol 239 (Product Life Cycle Support).
ISO	International Organization for Standardization.
PLCS	ISO 10303-239, "Product life cycle support", published as a first edition International Standard in October 2005.
TES	Technical Enabling Services, a UK MoD organization seeking to provide coherent technical support and advice to DPA and DLO staff. This organization is the sponsor of this project.
TLSS	Through Life Support Standard, a proposed replacement for Def Stan 00-60, designed to support the development of requirements for support-related contracts for any stage of the CADMID life cycle.
SOM	Support Options Matrix, a UK MoD tool to show configurations for the allocation of responsibilities between MoD and Industry for various aspects of in-service support, depending on the top-level form of required contract.
UML	Unified Modeling Language. See "Information technology – Open Distributed Processing – Unified Modeling Language (UML) – Version 1.4.2", ISO/IEC 19501:2005

[11] <http://www.omg.org/docs/dtc/06-02-01.pdf>