# The XML Approach to Implementing Space Link Extension Service Management

**Gregory A. Welz[1], Gerhard Theis[2], Takahiro Yamada[3], Wallace Tai[1]**

*[1] Jet Propulsion Laboratory*
*California Institute of Technology*
*4800 Oak Grove Dr*
*Pasadena, CA 91214*
*USA*

*[2] European Space Agency*
*Space Operations Centre,*
*Robert-Bosch-Strasse 5, 64293 Darmstadt*
*Germany*

*[3] Institute of Space and Astronautical Science*
*3-1-1 Yoshinodai, Sagamihara 229-8510,*
*Japan*
*E-mail: tyamada@pub.isas.ac.jp*

## ABSTRACT

A feasibility study has been conducted at JPL, ESOC, and ISAS to assess the possible applications of the eXtensible Mark-up Language (XML) capabilities to the implementation of the CCSDS Space Link Extension (SLE) Service Management function. The primary objectives of the study are: (a) to help guide the CCSDS Panel 3 in its work on the design of SLE Service Management standard, (b) to establish an approach to service management by which expedient implementation of the related capabilities is possible. If the XML-based approach is deemed feasible, it is our intention to use it on a trial basis for supporting the ISAS MUSES-C mission to be launched in 2002 and the ESA Rosetta mission to be launched in 2003. The study is divided into two parts: (a) a trade-off analysis on the different methods (including XML) of data presentation and data manipulation for the SLE service management, and (b) a prototype task to validate the XML capabilities of representing the schedule requests and configuration parameters in the SLE service package.

The preliminary results of the prototype effort have shown certain prospect of this approach. The chief advantage of the XML-based approach is that it provides an easy means of translating the CCSDS service package represented in XML into center-unique formats for use by existing tools or for display purposes. The commercially available tools for creating service requests in XML format and data mapping have offered a very cost-effective solution to implementation. The prototype task has validated that the current definition of CCSDS service package is sufficient to accommodate the use of XML approach for the exchange of schedule request information between the service user and provider. However, it seems that there exist certain deficiencies in the current specification of the CCSDS service management with respect to the spacecraft telecommunications configuration. These deficiencies will have to be resolved by the CCSDS Panel 3. The XML-based approach has certain limitations too. Chief among them is that the exchange of the service management information between service provider and service user is through the file transfer protocol (FTP) rather than the connection-oriented SLE protocol. This has certain operational ramifications.

Overall, the SLE Service Management, as proposed, is a grand plan allowing high degree of interoperability between the tracking stations and the mission operations centers. As part of this implementation the data that needs to be moved around, shared, and used needs to be defined and represented in useful formats. These should be done and implemented before, or at least as part of the overall system implementation. XML is the ideal candidate to do this. And if in the course of the SLE implementation, the implementation falls behind schedule or is dropped, XML will still serve the community well as a means of sharing data without the full SLE implementation being in place.

## INTRODUCTION

Is XML the right tool for bridging between CCSDS SLE Service Management (SM) implementations and dissimilar implementations? In short yes, XML is a useful tool for conveying the key information between CCSDS implementations and others such as Jet Propulsion Laboratory's (JPL) Deep Space Mission Service (DSMS).

Implementing the full CCSDS SLE Service Management (SM) architecture is a long-term process. However there are many intermediate points that can be achieved that get us closer. One such point is standardizing the data format for conveying the Service Management information. Currently DSMS has a set of unique formats to describe service management information that projects must work with in order to use the DSMS services. These DSMS formats are currently not directly compatible with CCSDS SLE SM formats. However a tool to convert from a CCSDS format Service Package to DSMS compatible form is conceivable. This would enable missions to adopt CCSDS SLE SM tools and formats in advance of DSMS.

The best transition point to bridge between CCSDS formats and DSMS is between the CCSDS Service User and the Service Provider, in this case DSMS. This transition point is illustrated in Fig. 1. The Service user would create CCSDS SLE SM compatible files while DSMS would be able to continue to use the existing formats and tools.

To simplify the transfer of Service Agreements (SA) and service Packages (SP) from the Service User to DSMS a file transfer protocol, such as FTP or e-mail, would be used to send a human readable file. The transfer of the files is easy to implement, but what format should that file be in, and what should it contain are critical questions to resolve before any good implementation can be created. Toward this end a study was performed to resolve the following questions:

- What file format to use: a Parameter Value language (PVL), or eXtensible Markup Language (XML)?
- Can we actually translate from a CCSDS Service Package to a DSMS compatible format?
- Does a CCSDS Service Agreement and Service Package have all of the information needed by DSMS?

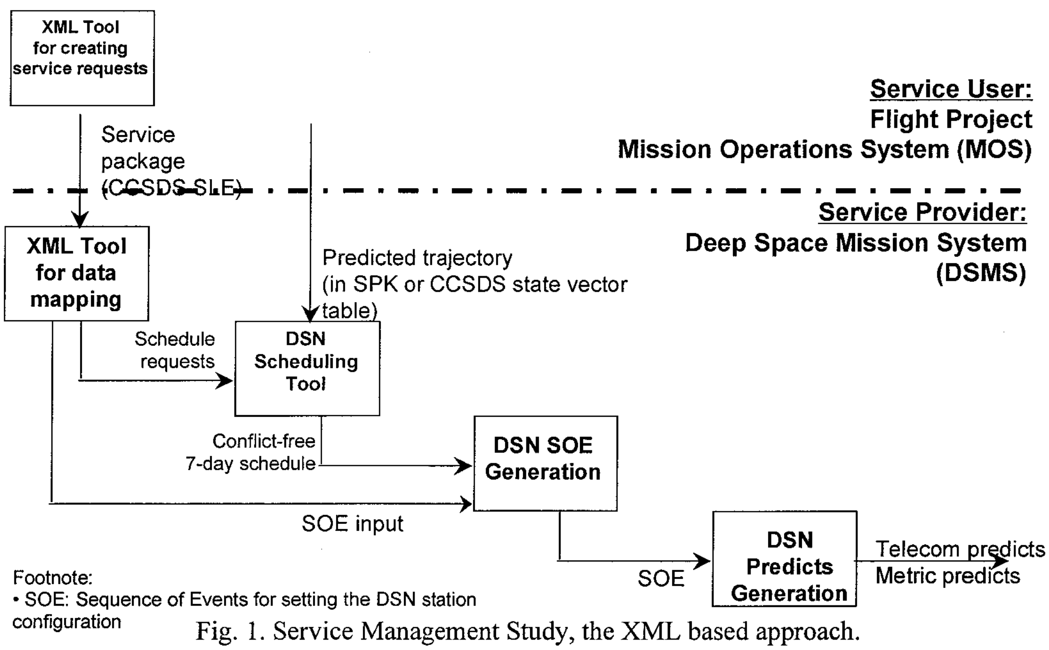


Fig. 1. Service Management Study, the XML based approach.

## ANALYSIS OF XML FOR SLE REPRESENTATION

### Representation goals

To achieve the interoperability goals of the CCSDS SLE SM architecture in the near term, Service Users and Service Providers need to be able to share key SM information, without undergoing major changes to their current systems. The easiest method of doing this is to transmit SA and SP information, normally captured in Managed Objects, in a standardized human readable format. The standardized format permits easy parsing of the files for automated systems, while the human readable format enables reading the files with out any special tools. Of the human readable formats there are few options: PVL and XML being the leading contenders.

Of all the options examined, ASCII text was dropped very early because of two factors. The first is that any representation that is based on positional placement is very prone to errors, and is also very much application specific. The second factor is that any non-positional solution ends up being similar or identical to a PVL or XML like solution, so any issues with those approaches would exist for ASCII text files without any benefit associated with the regularized features of a PVL or the supporting infrastructure of XML..

PVL has many good features, but lags XML in total available capabilities and support beyond the defined application base. PVLs are used in many places, including DSMS and CCSDS [1]. But the actual definitions and supporting tools for each PVL is unique to an application and has very little wide acceptance.

XML [2] was designed for e-commerce and as such has a large public tool set and a standard definition. This provides any application that uses XML a large set of pre-built tools and supporting infrastructure. Any other solution would require the development of comparable tools and infrastructure. As part of XML, XML Style Language (XSL) provides a ready mechanism for converting from one XML format to another, and with the built in support for Java, normally complicated translations can be very easy to implement. Because of all the features that XML provides for representing data it is rapidly gaining acceptance across the CCSDS community. CCSDS SLE would be greatly aided by using XML in the transfer of SM information.

**General comments about CCSDS SLE SM and XML definition**

In the CCSDS SLE SM documentation there are several places where managed data objects are used (such as data bases or data files). In several instances the document discusses using a structured ASCII text file for these objects. . A mapping rule from these data objects to text representations need to be defined and implemented prior to implementing the SLE SM for they will, in part, ensure interoperability among different implementations of SM. Regardless of the implementation choice (ASCII text files, PVL, XML) the data definitions used must be consistent with the SLE architecture or the resultant data files will not support a later SLE implementation. Implementing just the data object files will not replace the full SLE implementation, but it can be used for many organizational interface issues and can aide in the full implementation of the SLE architecture. The regular structure of the SLE SM Managed Objects (MO) that lends itself well to being codified in GDMO, makes it easy to describe the SLE SM MO structure in XML (illustrated in Table 1).

Table 1, Mapping of SLE SM to GDMO and XML Schema constructs.

| **SLE Service Management** | **GDMO** | **XML Schema** |
|---|---|---|
| Managed Object class | MANAGED OBJECT CLASS | `complex` data type |
| | PACKAGE | `group` |
| inheritance | DERIVED FROM | `extension` |
| containment rule | NAME BINDING | nesting of XML schema elements |
| attribute | ATTRIBUTE | `element` with `simpleContent` or `complexContent` |
| | ASN.1 data type | `simpleType` or `complexType` with `facet` as appropriate |

Because of the expected wide use of SLE, and the number of different SLE applications, an application-centric view for sharing data is not reasonable. This means that no single agency can, or should, define the data definitions. Nor should a proprietary solution be used for implementing the data definitions.

**Preliminary work**

Considering the positive reception of XML for presenting information of all kind, several CCSDS member agencies are investigating potential uses of the XML technology for space data. With regard to using XML for the exchange of static SLE Service Management information, the following is available today:

- a draft CCSDS White Book [3] with the formal GDMO specification for all Managed Object classes is in preparation. The actual GDMO specification has already been produced 'manually' as a refinement of the textual description provided in [4] and [5]. Syntactical correctness and internal consistency have been verified by means of a commercial tool.
- rules for mapping instances of GDMO Managed Object classes to XML-documents have been elaborated in the ESA study and are compiled in a draft CCSDS White Book [6];
- a tool - the Transfer Service Ordering Assistant, TSOA - for generation, visualisation and validation of XML documents representing Transfer Service Orders.

## PROTOTYPE EFFORT

Selecting XML to share SM data is far from demonstrating that is a actually feasible, therefore a small prototype effort was organized to see if it was possible. The bounds of the prototype effort were deliberately kept constrained to be able to show the flow of a product and to raise issues or concerns that may result when performing the translations.

### Scenario

The scenario used for the prototype was:

- A Service Agreement (SA) is expected to already exist, the spacecraft description has been defined, a standard configuration exists, and the standard terms and conditions have already been worked out between the Service User and the Service Provider.
- The Service User creates a Service package (SP). This SP is submitted to the DSMS service management organization, acting as the Service Provider. The package identifies support dates, resource needs, and configuration information. This SP would be in an XML format, and be structured along the same lines as the SM MO. For the prototype this product would only be done to a sufficient level of detail to illustrate what information needs to be conveyed, it is not implemented to a complete depth.
- The SP provides the data necessary to help populate the Deep Space Network (DSN) Seven Day Schedule and also create a DSN Sequence Of Events (SOE) keyword file used in the actual operations of the pass. These products are implemented only to the depth to illustrate the feasibility and identify issues, but not to a complete depth.

The Service User could use a simple tool, such as the Transfer Service Ordering Assistant, to create a document that is an XML representation of an SP. This document is stored locally to be eventually transferred to a Service Provider. For the actual file transfer no specific mechanism is prescribed. Possible choices are: an arbitrary File Transfer Protocol, the http protocol, e-mail attachments or simply mailing a diskette. For the actual transfer of XML documents it is sufficient that all parties taking part in the exchange bilaterally agree on a mechanism.

The Service Provider receives and processes the XML file to determine the requested services. This processing may be a simple as viewing the XML file with a COTS browser, or as is the case with DSMS convert the delivered XML file into a format compatible with current DSMS tools.

Code sample (1) presents a fragment of a XML Schema implementation of a service package. An item of special note is the "type="CCSDS:..." This identifies that the type definition is in a separate file, this is advantages since it simplifies re-use of standard elements by placing them is a common location accessible to all.

```
<complexType name=servicePackage>                                          (1)
  <element name="spacecraftTracking"  type="CCSDS:spacecraftTracking"
minOccurs="0"/>
  <element name="eventHandler"        type="CCSDS:eventHandler" minOccurs="0"/>
<!-- eventHandler object captures information that could make up a TMOD DSN SOE
-->
  <element name="notificationLog"         type="CCSDS:notificationLog"/>
  <element name="spaceLinkSession"        type="spaceLinkSession"/>
<!-- Space Link Session came from CCSDS 910.7-W-1.1, I interpreted it as a catch
all for all the RF related information associated with an instance of a service
package -->
  <attribute name="spackmo-id"        type="CDATA"/>
<!-- Assumption, spackmo-id contains as part of its DN the station ID(s) it
applies to.-->
  <attribute name="expir"             type="CCSDS:Timecode"/>
  <attribute name="spack-state">
    <simpleType>
      <restriction base="string">
        <enumeration value="waiting"/>
```

```
      <enumeration value="under validation"/>
      <enumeration value="committed"/>
      <enumeration value="executing"/>
      <enumeration value="done"/>
    </restriction>
  </simpleType>
</attribute>
```

After the XML schemas were developed for each product (SP, 7 Day Schedule, SOE), realistic data was placed into the defined format to check for correctness and completeness. In most cases this was a trivial step to perform, except when it was discovered that translation between two products didn't map well. This required augmenting the elements in one product to carry the additional needed information for the next product. Accordingly, this step revealed some gaps in the SLE SP description that should be filled.

Code sample (2) is a fragment of a populated Service Package, based on the definition from sample (1). The sub-elements of the spaceLinkSession are defined elsewhere.

```
<servicePackage spackmo-id="MC-02A" expir="2005-11-30 22:00"                    (2)
      spack-state="committed">
   < spaceLinkSession mo-id="slsmoid-MC-203" sls-state="validated"
          sls-start-time="2004-11-11 13:00:00" sls-end-time="2004-11-11 22:00">
      <f-401SpaceLinkCarrierPackage slcp-mo-id="f-401SpaceLinkCarrierPackage01">
         <f-401SymbolStreamPackage-mo-id>SymbolStreamPackage01</f-
401SymbolStreamPackage-mo-id>
         <f-tc-session-prod>
          <f-tc-session-prod-mo-id>F-tc-session-prodL01</f-tc-session-prod-mo-
id>
          <bit-format>NRZ-L</bit-format>
          <carrier-use>remnant</carrier-use>
          <frequency>7156.53</frequency>
          <iq-channel>NA</iq-channel>
          <modulation-index>700</modulation-index>
```

The final step was to the translation script/algorithm to show that it was possible to transfer easily from one product to the next. This was performed only for the translation from a SP to the DSN Seven Day Schedule, this was adequate to identify issues and to demonstrate the ease of the translation. The SP to SOE was performed manually, to verify that it was possible, but a script wasn't generated to do this step.

Code sample (3) is a fragment of a short XSL script which combined with a Java program, called from within the XSL script, will extract from the Service Package XML file the information needed to populate a DSN seven day schedule. For this example, it is assumed that a Service Agreement was on file, and defines some DSN specific items needed for the schedule that are not explicitly called out in the Service Package.

```
<xsl:template match="serviceAgreements">                                       (3)
<SIMULATED_XML_SCHEDULE>
  <Case><xsl:value of select="Passed-In-Case"/></Case>
  <Schedule><xsl:value-of select="Schedule-file"/></Schedule>
  <Range_Start><xsl:value-of select="Period-start"/></Range_Start>
  <Range_End><xsl:value-of select="Period-end"/></Range_End>
  <ScheduleItems>
  <xsl:template match="servicePackage">
    <xsl:template match=".[@spack-state='committed']
      <xsl:template match="*[@sls-stop-time &lt;= {serviceAgreements/Period-End}
                    and *@sls-stop-time > {serviceAgreements/Period-start}]"/>
```

**Findings on mapping SLE SM MO to DSMS formats**

Exploring the mapping of SLE SE MO to the DSMS formats proved in general very easy. However, there were areas where the current CCSDS SLE SM definitions proved incomplete. As an example of this the current CCSDS SLE SM definitions could identify when a session is to start and finish, and can even identify the types of resources that maybe needed to support a pass, so there is sufficient information for scheduling supports. But the SM definition is currently not detailed enough to capture the actual activities that are to happen within a session (i.e., the actual timing, duration, and configuration details of the individual activities that need to occur to support a spacecraft pass). This makes the translation from a SP to a sequence of events describing an individual pass very difficult. Some early suggestions to Panel 3 include the addition of start and end times to lower level managed objects, such as data rate (which would change during a pass, or to indicate different tracking modes and when they change). Admittedly some of the requested changes can be simulated with in the existing constructs, but it rapidly becomes a convoluted product that ceases to be easily followed.

The maturity and level of detail of the CCSDS SLE SM definition contributed to the challenge in mapping SP to the SOE. Many parts of a SOE can be mapped to a SP, but there are still some SOE keywords that don't have an equivalent CCSDS object/attribute. Some of these items are peculiar (SHEMT) to the DSN, and others are descriptive of a link, which is not currently addressed within the CCSDS documentation. But there are also some items that need to be addressed in a SP that currently doesn't exist in the CCSDS SLE SM definition.

In theory it seems reasonable to map the XML structures to the SLE SM Managed Objects (MO), in reality the mapping is a little off. The MOs maintain knowledge beyond what is obvious in the containment relations and attributes, which is all the initial mapping captured. To include the additional knowledge, such as relationships across MOs and some of the more subtle containment information, the XML structures need to include information that isn't in the original MO that it is mapped to, or we need to assume that some of the relationship information is captured in the "mo-id" as part of the Distinguished Name.

## FUTURE OF SLE AND XML

Several space missions intend using SLE Transfer Services in mid term future (INTEGRAL, ROSETTA, MARS-EXPRESS, MUSES-C). Analysis of their operation scenarios shows that their service management requirements are far below the level of complexity which could be supported by an SLE Service Management Interface as it is defined today [SLE-SM]. Consequently, concerns are voiced: expense and risk of implementations appear not justified by the mid term requirements of projects; Complex specific solutions may be more appropriate. In response to such concerns, an approach towards a simple initial service management interface is outlined and it is shown how XML could be employed.

With regard to service management, mid term mission scenarios appear to share the following characteristics:

- Details of space link configurations, production and service provision can be fixed a long time ahead of the actual operation.

Hence, instead of being prepared for attribute value setting in real-time, it would be sufficient to prepare several spacecraft support MOs, each capturing a spacecraft operation mode, i.e. a fixed set-up of attribute values. Long before the actual support operations would become due, the Complex may perform equipment level tests in order to confirm feasibility and plausibility of the configuration. If a desired configuration or schedule turns out to be problematic, a new spacecraft support Managed Object could be negotiated.

- Operational patterns during subsequent passes of a spacecraft are largely identical;

In preparation of a pass, mission management would request the activation of a given spacecraft operation mode, i.e. it would select the respective spacecraft support Managed Object and determine at which time it's execution shall start. Sufficiently ahead of due time, Complex Management would configure ground station and other equipment as described by attribute values in the selected spacecraft support MO.

- Adjustments of space link processing or service production are very rare and can be scheduled sufficient time ahead of their due time.

Consequently all operations and behaviour aspects of interface objects which relate to real time interactions between Complex and mission can be left aside. Instead it is sufficient to provide a mechanism, which allows to announce the necessary adjustments days or hours ahead of due time and to overwrite a few parameters in a specific spacecraft support table.

- Reporting is required only some time after a pass.

As debriefing information is not required in real time, all attribute reading and the complicated notification mechanisms of the current Recommendation can be ignored. It is sufficient to compile debriefing information after completion of pass and eventually transfer this report to the customer;

## CONCLUSIONS

It has been shown that the XML covers the static aspects of the full SLE Service Management Specification.

XML's capability ensures the generation of valid Service Packages, their visualization by commercial browsers and enables their processing by XML-aware service management applications.

XML is an excellent choice for transferring CCSDS SLE SM information from a service user to a service provider, such as DSMS. XML brings many tools and features that enable easy standardization, re-use, and translation. The actual translation from a CCSDS format to DSMS formats was a little challenging, but in the end achievable, especially if the CCSDS SLE SM standards get extended to address the holes found during the study.

## ACKNOWLEDGEMENTS

The research described in this paper was carried out at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration.

## REFERENCES

[1] CCSDS-641.0.B-2, Parameter Value Language Specification, Issue 2. May 1999 (http://www.ccsds.org/documents/pdf/CCSDS-641.0-B-2.pdf)
[2] Extensible Markup Language (XML) 1.0 W3C Recommendation, 6.October 2000, see: http://www.w3.org/TR/REC-xml
[3] Space Link Extension – Service Management – Managed Object Formal Specification. CCSDS 910.6-W-x.x, Draft White Book, in work
[4] Space Link Extension – Service Management Specification. CCSDS 910.5-R-2, Red Book, August 2001, (http://ccsds.org/red_books.html)
[5] Space Link Extension – Service Management – Space Link Physical Layer Managed Objects Specification, CCSDS 910.7-R-1, Red Book, August 2001.
[6] Space Link Extension – Service Management – Managed Objects Mapping on XML CCSDS 910.xyz-W-1.0, Draft White Book, March 2001, L.Bednarz