The specification had the following key requirements:

- Meet minimum common business requirements of users but allow for extensibility.
- Specify security, communication and interoperability standards for vending clients and vending servers.
- Be platform independent.
- Utilize existing industry technology standards where available for security, communication and interoperability.
- Support existing prepayment standards such as, STS and CVS (NRS009 Series).
- Ensure industry support and ownership.
- Specified technologies must be supported by generally available development tool sets.
- Specified technologies must utilise existing developer skills.

This culminated in version 1 of the Online Vending Interface specification (NRS009-6-10), also termed “XMLVend”, which was released in February 2004. XMLVend has since undergone five subsequent improvements and the latest version of the specification is XMLVend 2.1, which was released in February 2006. XMLVend version 2.1 is currently envisaged as stable and will be implemented as full service implementation for Online Vending in Eskom.

XMLVend 2.1 – an overview

The following is a brief overview of the XMLVend 2.1 specification.

Use cases and business application domains

Use cases have been used to define and describe the functions supported by the XMLVend specification. In XMLVend 2.1 the use cases have been divided into business application domains. The grouping of the use cases into business application domains eases maintenance of XMLVend, while allowing each domain to develop independent of each other. The business application domains and use case are:

Revenue management:
- Cancel token
- Check batch totals
- Confirm customer details
- Create deposit slip (new)
- Collect free basic electricity (FBE) token
- End batch
- Free issue token
- Meter credit transfer (replacement token)
- Pay account (new)
- Purchase Credit token
- Purchase trial credit token
- Reprint deposit slip
- Reprint end batch
- Reprint transaction (new)
- Start batch
-Vendor statement (new)
- Verify token

Meter management:
- Confirm meter details
- Customer report fault (new)
- Non-meter specific engineering token (new)
- Non-meter specific engineering token (new)
- Update meter key data (new)

The revenue management domain specifies all use cases that have financial implications and the meter domain includes all use cases that are used to manage the prepayment meter configuration. It is envisaged that some XMLVend implementations may only implement the revenue management use cases and some only meter management use cases.

XMLVend 2.1 introduces 4 new use cases in the meter domain:

- Create deposit slip: This use case is used to generate (print) a deposit slip for the Vendor. It is usually implemented by systems that operate in an upfront vending model and therefore do not require credit management through batches.
- Account payments: This use case enables account payments for services other than prepayment tokens, such as debt recovery or service charges (but not limited to).
**Reprint transactions:** This use case was previously called the "Reprint token", it has been expanded to allow the reprint of both credit vend transactions and account payment transactions.

**Vendor statement:** This use case is used to display (print) the vendor’s credit updates and currently available credit. The server configuration determines the number- or period- of entries to be displayed.

**XMLVend 2.1** introduces 4 new use cases in the revenue domain:

- **Customer report fault:** This use case is used to make customer fault reporting functionality available at the vending point.
- **Meter specific engineering token:** This use case is used to request meter specific STS engineering tokens, which are defined in NRS009-6-7. This ensures that meter are managed and configured with the same data as the vending database.
- **Non-meter specific engineering token:** This use case is used to request non-meter specific STS engineering tokens, which are defined in NRS009-6-7.
- **Update meter key data:** This use case is a new (better managed) version of the previous "Key Change Token" use case. This use case only allows a key change to the vending database meter configuration and can enable all vendors to issue risk free key change tokens where required. It also optionally allows for a matching power limit token to be issued with the key change token if required.

A further grouping of use cases, referred to as *Base Use Cases* have also been defined. These use cases do not belong to any specific business application domain but are used support the use cases in the revenue and meter management domains. Currently the only use case defined in this domain is the *Issue Advice* use case.

**Designing the message pairs**

XMLVend is essentially a client - server interface protocol specification, based on the request / response pattern (Figure 1).

Each use case consists of a request and response message to achieve the desired outcome. In XMLVend 2.1 the request / response message pairs were designed and modelled as object diagrams using the unified modelling language (UML). This approach ensured that an object oriented design was adhered to throughout the message design process. The benefits of an object oriented message structure are:

- Improved reusability of objects.
- Custom extensions supported through specialisation.
- Object relationships are encapsulated into the message.
- Directly mapping to XML schema.
- A natural, simple message structure.
- Messages are self describing and easily understood.
- Duplication is significantly reduced (if not eliminated).
- Objects are naturally divided into their respective business application domains.
- Significantly improves maintenance and management of updates to the message structures.

An example of a very simple XMLVend request message, modelled using UML is illustrated in Figure 2. Here an FBE Token Request (FBEReq) is shown to be a specialisation of the Base Vending Request (BaseVendReq). Similar other specialisations may be implemented by developers to extend the functionality for XMLVend in a controlled and compatible manner.

**Implementing the message pairs**

XMLVend specifies the open industry standard XML / Web Services as the method to realise the use case message pairs. XMLVend, has also adopted the Web Service Interoperability (WS-I) organisation guidelines for the implementation of web services to ensure interoperability.

**Web Services provide the following** (and as illustrated in Figure 3):

- Platform-independent format language for structured data exchange. This is achieved through the use of eXtensible mark-up language (XML).
- A way of describing the structure of the XML data being exchanged. This is achieved through the use of XML Schema. The XMLVend schemas form the core of the XMLVend protocol and are mapped directly from the use case UML diagrams into their respective business application domains.
- A standard method of packaging the data for transmission over the communications network. This is achieved through the use of SOAP
- A way for the Web services to describe their public interface to clients. This is achieved through the use of Web Services Description Language (WSDL). The XMLVend WSDLs are also a major deliverable of the specification and is also mapped into the respective business application domains.

**Implementation protocols**

<table>
<thead>
<tr>
<th>Protocol</th>
</tr>
</thead>
<tbody>
<tr>
<td>XML 1.0 (second edition)</td>
</tr>
<tr>
<td>SOAP 1.1</td>
</tr>
<tr>
<td>HTTP 1.1</td>
</tr>
<tr>
<td>Compression algorithm - GZIP</td>
</tr>
<tr>
<td>Security protocol - SSL/TLS</td>
</tr>
<tr>
<td>Network communications protocol - TCP/IP</td>
</tr>
<tr>
<td>Ethernet GPRS POTS Dial-up X25 leased lines</td>
</tr>
</tbody>
</table>

**Fig. 2:** FBE Request Message Object Diagram.

**Fig. 3:** XMLVend - Implementation Protocol Stack.
A standard method of transporting the data across the network. This is achieved through the use of hypertext transport protocol (HTTP) and TCP/IP.

XMLVend further makes use of industry standard compression algorithm, GZIP, to optimise the size of the message pairs. Secure Socket Layer (SSL) combined with public key certificates provides client/server authentication, message encryption and integrity.

Two implementation models have been specified for XMLVend. Implementation model 1, sometimes referred to as the "Normal Vendor" model, illustrates that the customer interacts with an XMLVend client (through an operator) that communicates directly with the XMLVend server. This model may be applied to (but not limited to) implementations, where vendors currently using offline Credit Dispensing Units (CDUs) are upgraded to online vending clients.

Implementation model 2, illustrates, that the customer interacts with a terminal that communicates using a proprietary protocol to an XMLVend client. This client in turn communicates with the XMLVend server.

This model may be applied to vendors who currently have a footprint of terminals and would like to add prepayment electricity vending capabilities to their terminals.

XMLVend pilot and implementation

XMLVend has been extensively piloted and tested during Eskom’s successful Online Vending Pilot project. Several lessons that were learnt and shortcomings identified during the pilot, have been improved in XMLVend 2.1. Eskom has also tested numerous vending client suppliers for compliance to XMLVend, which will enable Eskom to significantly increase its current vending footprint and therefore customer service.

Eskom is currently developing its implementation of the full Online Vending Project where it will make use of XMLVend 2.1 to interface to its vending clients.

Conclusion

XMLVend 2.1 has matured into a robust and stable specification for online vending and has become a key enabler for standardised, secure, interoperable and extensible Online Vending Systems.

For more information and to download the specification visit the specification website, http://www.nrs.eskom.co.za/xmlvend/