Implementation of TASE.2, MMS and APLI Protocols

Novak Savić[1], Miroslav Popović[2], Branislav Atlagić[3], Miodrag Temerinac[4]

Abstract – **This paper describes the implementation of Association Control Service Element/Presentation Layer, Manufacturing Message Specification, and Telecontrol Application Service Element 2 protocols as well as the protocols themselves. Besides implementation of those protocols, the paper presents an approach used for testing and verification. Telecontrol Application Service Element 2 protocol is unitized in 9 (nine) conformance building blocks. This implementation includes building blocks: 1, 2, 5, 7 and 9, corresponding MMS conformance building blocks and appropriate APLI services.**

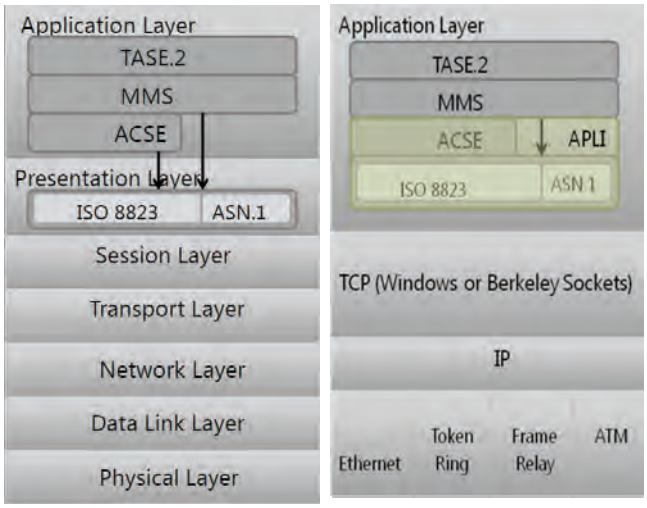
Keywords - **Telecontrol Application Service Element 2, Manufacturing Message Specification, Association Control Service Element/Presentation Layer.**

I. Introduction

APLI (Association Control Service Element/Presentation Layer Interface) is used for establishing and maintaining logical connection (association) between communication points. MMS (Manufacturing Message Specification) specifies a way to emulate, or model, the capabilities and functionality of a programmable device, and the means to manipulate this model. TASE.2 (Telecontrol Application Service Element 2) is the application layer standard that represents extension to MMS protocol. The purpose of TASE.2 protocol is to regulate communication between Inter Control Centers.

TASE.2 and MMS are able to operate over either an ISO-compliant transport layer or a TCP/IP transport service, as long as ISO layers 5-7 are maintained. MMS uses ACSE (Association Control Service Element [7], [8]) to establish logical connection (association). ASN.1 [10], [11] is used for describing the abstract structure of a protocol as well as object modeling facilities. There is standard C Unix specification for ACSE/Presentation Layer Interface (APLI), which goes on top of the ISO stack. We have taken advantage of another approach by developing APLI over Windows and Berkeley sockets.

Figure 1 shows TASE.2 and MMS protocol over: a) the ISO stack, b) APLI and Windows (Berkeley) sockets.



a) b)

Fig. 1. TASE.2 and MMS protocols over: a) ISO stack, b) Windows (Berkley) socket

A section II, III and IV contains an overview of TASE.2, MMS, and APLI protocol, respectively. Implementation of those protocols is represented in section V, VI and VII. Testing can be found in section VIII. Finally, section IX describes further plans.

II. APLI

The purpose of APLI is establishment and maintaining logical connection (association) between applications and data exchange among those applications. An association is identified by Application References of those applications. Application Reference constitutes of AP-Title and AE-qualifier. APLI offers following services: M-ASSOCIATE (establishing association), M-RELEASE (peaceful association disrupting), M-DATA (data transfer), M-U-ABORT (user forced association disrupting) and M-P-ABORT. All APLI services except M-DATA are confirmed services. Prior to any data exchange between applications, application association must be established.

III. MMS

The Manufacturing Message Specification (MMS) addresses the integration of shop floor devices and the computer systems that control them. MMS defines a structure for the messages required to control and monitor these devices. MMS is composed of following components: Virtual Manufacturing Device, Environment and General

[1]Novak Z. Savić is with the Faculty of Techincal Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, E-mail: savic.novak@dmsgroup.co.yu.

[2]Miroslav Popovic is with the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, E-mail: miroslav.popovic@micronasnit.com.

[3]Branislav Atlagic is with the Faculty of Technical Sciences, Trg Dositeja Obradovića 6, 21000 Novi Sad, Serbia, E-mail: branislav.atlagic@rt-rk.com.

[4]Miodrag Temerinac is with the MicronasNIT, Fruškogorska 11, 21000 Novi Sad, Serbia, E-mail: miodrag.temerinac@micronasnit.com.

Management, Conditioned Service Response, Domain Management, Program Invocation Management, Unit Control, Variable Access, Data Exchange Management, Operator Communication, Event Management, Event Condition, Event Action, Event Enrollment, Event Condition List and Journal Management. Only those protocol objects and services which have been implemented will be described.

The MMS services define the externally visible behavior of an MMS server application process. This behavior is modeled by describing of an entity called Virtual Manufacturing Device (VMD). *Environment and General Management* are concerned with the application association, which refers to establishing and managing connection between two applications. *A Domain* represents a subset of the capabilities of the VMD used for a specific purpose. *The Unit Control* object represents a collection of MMS objects, Domains and Program Invocations that may be loaded and managed as a unit. *The Event Condition* object models that portion of the state information that is concerned with event detection and prioritization. *An Event Action* object is an MMS confirmed service that shall be executed whenever a specified transition of an Event Condition object's state field is detected. *An Event Enrollment* represents a request from an MMS user to be notified about the occurrence of one or more specified transitions of an Event Condition object, or to delay execution of a confirmed MMS-service until the occurrence of one or more specified transitions of an Event Condition object. *The Event Condition List* object shall be used to reference groups of Event Condition objects that are required to be operated on as groups.

IV. TASE.2

TASE.2 is based on the client/server concept. All data transfers originate with a request from one control center (the client) to another control center, which owns and manages the data (the server). Multiple associations may be established from a client to multiple control center servers. Multiple associations may also be established to the same control center for the purpose of providing associations with different Quality of Service (QoS). To provide access control, the server checks each client request to ensure that the particular client has access rights to the data or capability requested. Access control is provided through the use of Bilateral Tables (BLTs) defined for each client/server association. BLTs provide execute, read/write, read only, or no access for each item that can be requested by a client.

TASE.2 services are provided via TASE.2 server objects. There are two basic types of services in TASE.2, called operations and actions. An operation is client-initiated via a request to a server, typically followed by a response from the server. An action is a server-initiated function. Other data and control elements typically exchanged between control centers are defined as “data objects”. TASE.2 Server includes the following types of objects: Association, DataValue, Data Set, Transfer Set, Account, Device, Program and Event. *Association objects* are used to establish an association, or logical connection, between two TASE.2 instances. *Data Value objects* represent values of control center data elements. *Data Set* objects are ordered lists of Data Value objects maintained by a TASE.2 server. This object enables a client to remotely define Data Sets via TASE.2. The establishment of the reporting criteria and the actual transfer of data vales are accomplished using the Transfer Set object. *Transfer Set* objects reside at a TASE.2 server and it is used by a TASE.2 client to establish the actual transfer of data values. *Account* is a generic term applied to a whole class of data objects used to represent information on schedules, accounts, device outages, curves, and other entities used by control centers. *Device objects* represent actual physical devices in the field for the purpose of providing services for a client to control them remotely. *A Program object* provides a client with remote operation of a program at a server site. An Event object represents a system event at a server site, such as a device changing state or the occurrence of a certain data error.

Representations of all 9 (nine) TASE.2 Conformance Building Blocks (CBBs) are shown below.

Block 1 (Basic Services) consists of following objects with appropriate services: Association, Data Value, Data Set Objects, Transfer Set, and Next Transfer Set. Block 2 is used to provide the capability to transfer power system data in more ways than periodic reports. Block 3 provides the possibility for a TASE.2 server to send power system data to a client with fewer bytes than required. Block 4 provides a general message transfer mechanism that also includes the ability to transfer simple text or binary files. Block 5 provides a mechanism for transferring a ‘request to operate a device from one TASE.2 implementation to another. It is composed of Device objects. Block 6 provides a mechanism for a TASE.2 client to perform program control at a server TASE.2 implementation site. It is composed of Program objects. Block 7 adds the Event Enrollment and Event Condition objects. It is composed of Device objects, Event Conditions objects and as well as Event Enrollment objects Block 8 offers the ability to transfer scheduling and accounting information between TASE.2 implementations is a key feature of TASE.2. TASE.2 generalizes this transfer capability to allow any data that is collected on a period basis. The transfer capability is accomplished via the Transfer Account data object. Block 9 provides a TASE.2 client with the ability to receive time series data and it is composed of TSConditions objects.

A control center is modeled with one or more Virtual Control Centers (VCC). A VCC is mapped onto an MMS Virtual Manufacturing Device (VMD). It therefore has the same definition and behavior as an MMS VMD. By mapping a TASE.2 VCC to an MMS VMD, the TASE.2 VCC performs the same function for a control center that a MMS VMD does for a real device.

V. Implementation Of APLI Protocol

APLI was designed over Windows (Berkeley) sockets whose services were realized through the following methods of class APLI_Provider: ap_snd() and ap_rcv(). Class APLI_Provider contains some additional methods: ap_open(), ap_close(), ap_init_env() and ap_set_env(), which are used for opening, closing, initialization and setting APLI environment, respectively. The APLI completely hides Windows (Berkeley)

Sockets in manner that associations and data exchange are done through the Application References.

There are a lot of service primitives that are used in the ap_snd() invocation, as first parameter, so desired service can be performed. An example of the requesting service A-ASSOCIATE is shown in figure 2.

```
ap_snd(A_ASSOC_REQ, pduBuffer, calledAppRef, callingAppRef)
```

Fig. 2. Example of Association Request APLI service

Parameter pduBuffer contains protocol data unit from upper protocol layer that need to be transferred via APLI protocol. All other services are realized in similar fashion.

VI. Implementation Of MMS Protocol

The most important is class VMD which represents model of Virtual Manufacturing Device. Figure 3 shows VMDs state diagram with transitions for MMS environment.

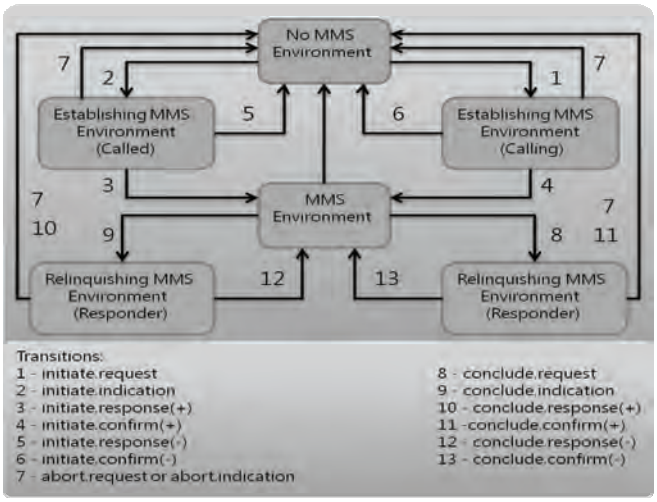


Fig.3. VMDs state diagram with transitions for MMS environment

MMS Environment state consists of three sub states. For the case of MMS client (initiator), the state diagram with transitions is shown in Figure 4.

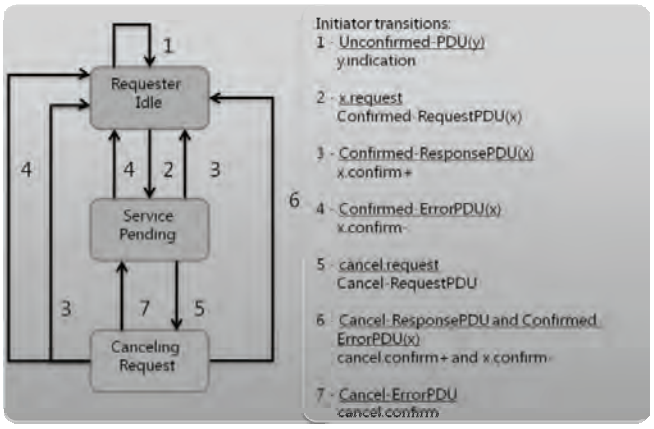


Fig. 4. MMS clients transition diagram.

MMS requests (responses) need to be packed into MMS protocol data unit, whereupon those MMS PDUs are carried out by APLI services. For the purpose of packing and parsing MMS protocol data units the MmsPduPacker and the MmsPduParser have been developed. Mapping of all fourteen MMS PDU types are shown in Figure 5.

MMS PDU	APLI Service Primitive
ConfirmedRequestPDU	M-DATA request, indication
ConfirmedResponsePDU	M-DATA request, indication
ConfirmedErrorPDU	M-DATA request, indication
RejectPDU	M-DATA request, indication
CancelRequestPDU	M-DATA request, indication
CancelResponsePDU	M-DATA request, indication
CancelErrorPDU	M-DATA request, indication
InitiateRequestPDU	M-ASSOCIATE request, indication
InitiateResponsePDU	M-ASSOCIATE response, confirm (with Result parameter accepted)
InitiateErrorPDU	M-ASSOCIATE response, confirm (with Result parameter rejected)
ConcludeRequestPDU	M-RELEASE request, indication
ConcludeReponsePDU	M-RELEASE response, confirm (with Result parameter accepted)
ConcludeErrorPDU	M-RELEASE response, confirm (with Result parameter rejected)

Fig. 5. Mapping of MMS protocol data units (PDUs) onto APLI services.

Data access on server side are done trough V_Put() and V_Get() methods. These methods are completely configurable so they can be used in integration with various applications and/or overlying protocols.

Alongside of other implementations that uses standard ACSE, Presentation, Session, COTP, and TPKT protocols over TCP protocol, this implementation have some improvement in regard to the size of packets that are transferred. The thing is that every packet is smaller for about 120 Bytes. In case of MMS request PDUs significant improvement in about 200% has been reached because the most of these PDUs are smaller than 50 bytes. Size of the MMS response PDUs can be very variable so as improvement, but it is still present.

This can be very important in multi client-server environments. On the other hand this implementation is not compatible with other vendor's implementations.

VII. Implementation of TASE.2 Protocol

This implementation of TASE.2 protocol uses services of previously developed MMS protocol and includes the following classes and the corresponding services: VirtualControlCenter, Association, BilateralTable, DataValue, DataSet, InformationMessage, DSTransferSet, TSTransferSet, IMTrasferSet, Device, Program, EventEnrollment, EventCondition, and some additional classes. The class VirtualControlCenter comprises all other classes and maps onto VirtualManufacturingDevice.

VirtualControlCenter have the same Environment states as VirtualManufacturingDevice. Association class represents application association which is used to uniquely identify the association between TASE.2 client and TASE.2 server. This implementation makes use of access control mechanism, so bilateral tables are implemented accordingly. Bilateral tables are closely related to association. Figure 6 depicts class VirtualControlCenter.

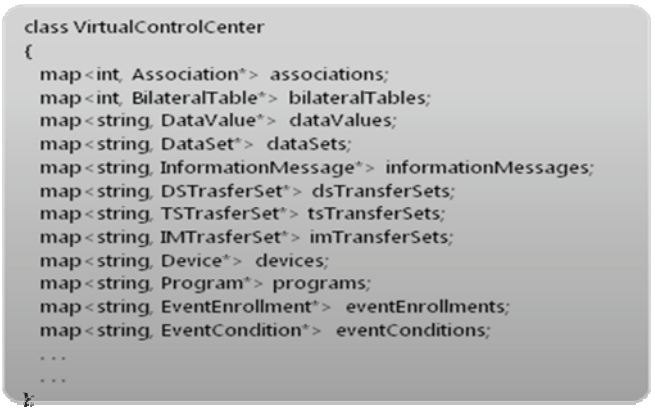


Fig. 6. Class VirtualControlCenter

Every Bilateral Table is connected with one and only one client control center. On the other hand one bilateral table can be used for more than one application association with particular client control center which can be configured via Application References. CCCD (client control center designation) identifies the TASE.2 client control center for which the Bilateral Table is defined with the TASE.2 server.

Hereafter are descibed the purpose of the other Bilateral Table members. The domainName member is the name of the TASE.2 Domain. Other members come in pair and are used for defining clients permitions. For example, dataValues comes with dataValuesACS. The dataValues field indicates those Data Value objects that are defined for the remote control center identified by the CCCD field. The dataValuesACS (class AsscessControlSpecification) represents the access permissions for dataValues for the remote control center identified by the CCCD field. While establishing association TASE.2 server checks existence of appropriate association identifier. If an identifier is not found, the association is refused, otherwise server proceeds with other authentication. Next follows checking of bilateral table existence for this client. If bilateral table is not found, the association is refused, otherwise server proceeds with other checks. Other checks include examination of bilateral table version and TASE.2 protocol version. If both checks pass, the association is successfully established.

Each service required by TASE.2 client is first checked trough the bilateral table, which is engaged with TASE.2 client. In this way TASE.2 client may perform only the services which previously have been granted to it.

For the purpose of pre configuration class ConfiguraitonLoader has been developed. After establishing association with TASE.2 server, the Configuration Loader simply interprets requests previously loaded from configuration file.

VIII. Testing

Testing and verification for TASE.2, MMS and APLI protocols has been performed trough C++ unit testing framework (CppUnit). TASE.2 protocol has been also tested in integration with Telvent (Spanish Company) OASYS SCADA

All dynamic data in Telvent OASYS SCADA are kept in the real time CMX database. The dynamic data are those data whose values are frequently changed. Access to CMX database is done trough ADO was tested under laboratory conditions between multiple TASE.2 applications which both play TASE.2 client and server roles. In both cases testing was considered successful.

IX. Conclusion

This paper presents overview of TASE.2, MMS, and APLI protocols and one possible implementation. Testing and verification has been presented too. The main purpose of developing these protocols is research in this area. This implementation can be used in real systems which has been approved trough integration with Telvent OASYS SCADA. Further plans are to go on with developing TASE.2 and MMS protocols, and develop conformance building blocks 3, 4, 6, and 8. Also, further plans include various protocols optimizations.

References

[1] *ISO 9506 part 1 – Industrial Automation System - MMS service definitions,* Geneva Switzerland 2003.

[2] ISO *9506 part 2 – Industrial Automation System - MMS protocol specification,* Geneva Switzerland 2003.

[3] *IEC 60870-6-503 – TASE.2 Services and Protocol,* Geneva Switzerland 2002.

[4] *IEC 60870-6-702 – TASE.2 Functional profile for providing the TASE.2 application service in end systems,* Geneva Switzerland 1998.

[5] *IEC 60870-6-802 – TASE.2 Object models,* Geneva Switzerland 2005.

[6] *IEC 60870-6-505 – TASE.2 User Guide,* Geneva Switzerland 2006.

[7] *ISO/IEC 8649 – OSI– Service Definition for Association Control Service Element,* ITU 1996.

[8] ISO/IEC 8650 – OSI – Connection Oriented Protocol for *Association* Control Service Element: Protocol Specification, ITU 1996.

[9] ISO/IEC 8327-1 – OSI – Connection Oriented Session Protocol: Protocol Specification, ITU 1996.

[10] ISO/IEC *8422*-1 – Information technology – Abstract Syntax Notation One (ASN.1): Specification of basic notation, ITU 1998.

[11] ISO/IEC *8422*-2 – Information technology – Abstract Syntax Notation One (ASN.1): Information Object Specification, ITU 1998.