

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

General Evaluation:

I do not approve of making the following documents an American National Standard;

- Device Description Language (CP/2003-1011) : **No with reasons**
- Device Management Protocol (CP/2003-1010) : **No with reasons**
- Session Data Transport Protocol (CP/2003-1009) : **No with reasons**

I do approve making the following documents an American National Standard;

- Packet Format (CP/2003-1008) : **Yes with comments**
- ACN Architecture Overview (CP/2003-1007r1) : **Yes with comments**
- Discovery (CP/2003-1012) : **Yes with comments**

I hereby grant ASC E1 and ESTA the non-exclusive, royalty-free rights to use my comments, and I understand that I acquire no rights in any publication in which my comments are used.

Name: **Dipl. Ing. Erwin Rol**

Organization: **Erwin Rol Software Engineering**

Street Address: **Tölzer Straße 12**

City: **Holzkirchen** Postal Code: **83607**

Country: **Germany**

Telephone: **+49-(0)8024-479377**

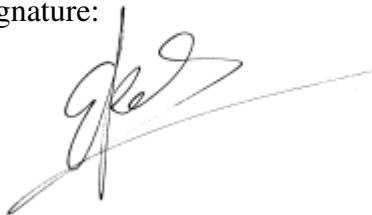
Fax: **+49-(0)8024-479379**

Mobile: **+49-(0)171-6929198**

Email: **erwin@erwinrol.com**

www: **http://www.erwinrol.com**

Signature:

A handwritten signature in black ink, appearing to be 'ER' with a long horizontal stroke extending to the right.

Date: 5. Dec. 2003

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

The following companies and/or persons signed in agreement with the comments given in this document:

Company: **ENTTEC**
Address: **Po Box 282**
City: **Kew, Vic** Postal Code: **3101**
Country: **Australia**
Contact: **Nicolas Moreau**
email: **nicolas@enttec.com**
www: **<http://www.enttec.com>**

Name: **Hado Hein (freelance LD, software author)**
Address: **Kopenhagener Straße 42**
City: **Berlin** Postal Code: **10437**
Country: **Germany**
email: **hh@batchmaker.de**
www: **<http://www.batchmaker.de>**

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

Comments or reasons:

General

Keywords

A first note on keywords like MUST, SHOULD, etc. Those keywords might be understood differently by different people, to prevent that there is for example RFC 2119 that describes the meaning of those keywords for use in RFC's. It would be wise to add a reference to RFC2119 to all ACN specs.

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

ACN Architecture Overview (CP/2003-1007R1)

Unclear and/or Unnecessary Text

In Section 4 there is talked about TCP/IP ports not working well with multicasting, without giving an explanation why not. Or an explanation should be added or the note should be removed. Since at the moment it is just wrong, TCP/IP ports work just fine with multicasting.

In Section 7.3 there is talked about serial ACN. The problem is that there is no serial ACN standard, but a section 7.3 makes you believe there is one.

Packet Format (CP/2003-1008)

Unclear and/or Unnecessary Text

In Section 8.0 on Page 12 there is mentioned that all ACN protocols **MUST** use network byte order, but in the same sentence there are exceptions allowed. So I think the **MUST** should be a **SHOULD**.

Discovery (CP/2003-1012)

SLP Restrictions

The SLP protocol is a standard, and there is no reason to change or cripple it. By forcing a MTU size restriction on the protocol it is not standard anymore. And because of this it will be hard if not impossible to use available SLP implementations.

Session Data Transport Protocol (CP/2003-1009)

Typo's and formatting errors

In Section 9.2.2 the packet description has a member called "Timein" this should be "Time in".

In Section C.6 the SRC=01 and SRC=11 in the packet example are wrong.

Unclear and/or Unnecessary Text

In Section 2 there is talked about a "multicast IP port", but there is no such thing as a "multicast IP port". There are multicast groups that can be registered and there are

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

UDP and TCP ports that can be registered.

In Section 5 the definition of the session identifier is not correct, since the session identifier does not unambiguously identify one particular session. It only identifies one particular session of a leader, different leaders can use the same ID.

In Section 6 there is mentioned that members can not send messages to each other, this is not 100% correct since NAK messages sent are received by other members too.

In Section 8 the impression is given that there is upstream reliability directly in SDT, this is misleading.

Refuse codes in section 10.5 are negative numbers, this might be useful in programming API's but is confusing in network protocols. It would be easier and more logical to just have unsigned error codes.

Message Format Descriptions

In Section 9 the SDT messages are described, but there seems a rather inconsistent use of terms. For example 9.2.2.6 and 9.2.3.1 mean the same thing but are described in different terms. Also the order of the rules is different from message to message, one time, for example 9.2.2.6, 9.2.3.1 and 9.2.5.4 all refer to the same thing but are all in different positions in the "rules-list". It would be easier when the rules followed a strict order and used the exact same terms. Also using "rectangles" of the same size for fields with different number of bytes is confusing, for example fields "JOIN/MID to be used by component joining" and "JOIN/Last used Total Seq. Nr." look like having the same number of bytes from the diagram. It would be better if the last field would have twice the height.

Security

In Section 3.1 there is talked about the fact that ACN does not offer any security, encryption or authentication. The section also simply points to applying security at "higher layers of the protocol stack", this means any manufactures could implement something in its own way and by that making his devices incompatible with other ACN devices. This is also true when security is implemented at a lower level by using something like Secure Socket Layer to transport data, since there is no ACN over SSL this means every manufacturer will be able to implement it in its own way. Also simply ignoring security is not an option with todays boom of wireless and Internet connected networks. I think there should at least be a better explanation on the risks of using the unsecured ACN protocol, but I would rather see a real effort to design certain (industry standard, no home made encryption, authentication algorithms please) in to ACN.

Small Systems

Forcing one ACN spec for all systems ranging from 8bit microcontrollers to highend

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

64bit systems just makes the system less useful, instead of more useful. SDT is designed to run over IP/UDP (or other packet-based highspeed networks like IPX), and with that already has a certain system requirements. If a system can not meet these requirements it can not use SDT and should use some other protocol to transport DMP. I think SDT should be used for transporting DMP (and maybe other protocols) over high speed networks by devices with enough system resources. For smaller systems there should be some other ACN protocol to transport DMP, for example the mentioned (but not specified) serial ACN. So instead of trying to force SDT to fit into a small system I think there just should be a separate spec for transporting DMP over less high speed networks to less powerful systems. With a spec like that, it will also not be a problem to design a “router” from ACN to “Serial-ACN”.

UDP Addresses

Trough out the specification there is talked about two kinds of addresses, upstream and downstream. The definitions for these addresses are not that clear and might cause confusion. First of all there are always two addresses involved in sending network traffic, a source address and a destination address. Also a UDP address always consists of a IP number and a Port number.

Unrecoverable Reliable Message

In section 8.6 there is an explanation on what to do when reliable messages are lost. Depending on the configuration this lost will be reported or not. What I think is problematic is that processing of the message continues, since important data might be missing and so processing new data might cause serious problems. A better solution would be to terminate the session and let the leader decide what to do.

Unreliable upstream

The fact that the upstream traffic is not reliable means that higher level protocols like DMP have to deal with making it reliable. This means not only a duplication of efforts (since every higher level protocol must implement it) it also means possible timing problems. For example when a DMP reply comes in too late it will have to retry it, since it doesn't know if the downstream request has failed or if the upstream reply has failed, basically for any request reply protocol like DMP the whole reliability of SDT is useless in this case, since it has to deal with timeouts and retransmits anyway. To solve this problem I suggest making the SDT protocol strictly one way (apart from some of the control messages like NAK) , and use a separate upstream session for upstream traffic, this session is needed to event subscription anyway.

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

Device Management Protocol (CP/2003-1010)

General

The layout of the specs seems different from the SDT spec, especially when it comes to the packet “diagrams” (they are ASCII-art here and line-art in the SDT spec) .

Unclear and/or Unnecessary Text

Like in the SDT spec the use of negative error numbers is confusing, The Codes in Table 3 on page 23 should simply be unsigned.

In Figure 6 on Page 19 shows the format for the DMP Standard Type. The first two bits are Rng, it would be more logical to move those two bits to position 4 and 5. That way the Message Type will simply start at 0 and does not need a shift, and the numbers in Table 4 will be actually correct.

Getting the DDL

The DMP spec introduces TFTP as the means to get the DDL from the device itself. This seems also like a unnecessary restriction to hope it can be implemented on the most low resource system as possible, since TFTP uses UDP and not TCP. Forcing the location to be on the device seems like a limitation to me, it would be better to advertise DDL via SLP, so that it could be stored on another place too (maybe a faster server than a small dimmer). But most of all I would advice to use HTTP to export the DDL's, this means using TCP, but thats not a problem, since even the smallest TCP/IP stack supports TCP. The advantage is that on a highly loaded network the downloading via TCP is more reliable than via UDP, and the use of HTTP is more common than the use of TFTP. Also the use of HTTP make it possible to just point a webbrowser at the device and look at its DDL (most webbrowsers can even pretty-print XML). This would even make it possible to have an ESTA or vendor-webserver on the Internet that holds all known DDL's that than directly could be used.

Property Types

The number of property types seems rather limited. Especially the fact that the DCID needs to be transferred as a string instead of the shorted binary form seems like a waste. Also that it is not possible to send bits in an efficient way is a shame, since there will certainly be properties that only need one bit, especially when setting a range or getting a range, or by using the Quick set message.

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

The size of the property values is described by the DDL of the device, this makes DMP useless without DDL. A way around it would be to add messages that can get the size and type of the properties. This would be the same format as the DMP_GETPROPREQ message, but it will return a fixed size type id for every property, this way the DMP protocol could do its own verifying of the requested properties. And this feature does not change or break anything.

Error Code Description

Table 3 has a number of error codes but does not explain them that well. Some of them are clear but a few of them are not. And some seem missing, like for setting values that are out of range, or strings that are too long. Also the DMP_DATASIZE seems useless, since a wrong data size is almost impossible to detect, since if the value of a property takes too much space everything in the message will shift and so the next address will also be wrong.

Device Description Language (CP/2003-1011)

General

First of all the DDL is not limited to describing ACN devices, it could also be used to describe DMX or RDM devices. That's why I believe DDL should not be part of ACN but that it should be a standard on its own.

Typo's and formatting errors

The document contains the following typo's and formatting errors

1. Page 16, Figure 6 – Closing > at wrong place
2. Page 17, Figure 7 – Closing > at wrong place
3. Page 28, Figure 12 – `</string/>` should be `</string>`
4. Page 28, Figure 13 – `shared (false|true) (false)` should be `shared (false| true) "false"`
5. Page 44, Figure 29 - `shared (false|true) (false)` should be `shared (false| true) "false"`

Unclear and/or Unnecessary Text

The Table 1 on Page 35 is not complete, since it does not describe write-only properties.

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

XML Restrictions

The XML restrictions proposed in Section 7.2.1.2 on page 22 are not acceptable, especially since some of the features removed by those restrictions are then reinvented in a poor way, like for example the replacement of “entities” and “xml:language” by languageset. The restrictions are applied under the name of keeping possible XML parser implementations small in case DDL is used in “lightweight systems”, this decision is flawed. In most cases the “lightweight systems” will be end devices, for example a dimmer, the dimmer has no need to parse or handle its own DDL description. On the other hand, systems that will be parsing DDL descriptions are most likely controllers that have more than enough system resources to do so. And also, today's super computers are tomorrow's “lightweight systems”.

String Type and MTU

The String property type as described in Table 2 on Page 39 refers to the MTU but makes no statement about the size of the MTU. This means that there can be major incompatibilities where a string property is used, since some device might have a big string that is too large for the MTU of some backbone protocol. There is also no solution in protocols like DMP on what to do in that case. This once again shows that forcing a MTU limit is not a smart thing to do, and cause more problems than it solves. Just allowing the maximum UDP size would be a better thing.

Lack of a Complete DTD

The DDL can not be described with a DTD. This seems mostly due to the protocol tag, since that has to be different for every protocol. Since DTD basically describes the syntax of a XML document, it seems like it is not possible to describe the syntax of DDL, this sounds like bad and incomplete design, and is another reason to make DDL a separate standard and taking a closer look at it.

Lack of Standard Profiles

The DDL has not enough standard profiles, this will cause the massive incompatibilities between different manufacturers. For example, every manufacturer can use his own DCID for a simple 8 bit dimmer, meaning that the user will get flooded with a large number of different (but the same in function) devices. The way to solve this is to let a central authority (the ESTA?) make a large number of common DCID that are to be used when ever possible. This should not be part of the DDL spec since that DDL spec describes the language and not the devices, the same is actually true for the standard behaviors, they should not be in the DDL spec but in a own document.

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

Languageset

The languageset that should make it possible to use different national languages in places where human readable text is needed does not work for all textual parts in the DDL. For example the text “human readable text” in `<behavior>human readable text</behavior>` is not translatable. XML offers a standard way to offer multi language documents in the form of the `xml:language` tag.

The languageset references in the form of `_key` are used for two different things, firstly to translate human readable text and secondly to make the DDL more readable. How internationalization human readable text in DDL files could be done is shown by use of several code examples.

Take the following DDL code;

```
<property label="human readable text">
</property>
```

When we replace this with;

```
<property>
    <label>Human readable text</label>
</property>
```

we can use the `xml:language` attribute to translate the text as follows;

```
<property>
    <label xml:lang="en">
        Human readable text
    </label>
    <label xml:lang="nl">
        Mens leesbare tekst
    </label>
</property>
```

This solution will also work for the behavior description, by introducing a label in the following way, replace

```
<behavior>human readable text</behavior>
```

with

```
<behavior>
    <label xml:lang="en">
        Human readable text
    </label>
    <label xml:lang="nl">
        Mens leesbare tekst
    </label>
</behavior>
```

Public Review Comments

BSR E1.17, Entertainment Technology Multipurpose Network Control Protocol Suite

In case a lot of translations are the same the XML entity references can be used in a following way

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE DDL SYSTEM "ddl.dtd"
[
    <!ENTITY label_hrt
        "<label xml:lang='en'>
            Human readable text
        </label>
        <label xml:lang='nl'>
            Mens leesbare tekst
        </label>"
    >
]>
<DDL>
<device>
<property>
    &label_hrt;
</property>
<behavior>
    &label_hrt;
</behavior>
</device>
</DDL>
```