Hello,

at first sight I would agree with you. For simply being sure not to use
the same number twice (at least within one file) it should work well.

But I see an other possible problem that won't be solved by neither of the
proposed numbering methods since it asks for the reuse of a number given
once:

If the rollingstock schema shall be used to follow the lifecycle of a
vehicle this would make some problems. The vehicles don't remain
unchanged. They may be sold or rent to an other owner or the owner applies
some renewing work on it and since the official numbering includes
describing parts the vehicle will receive a new number and even if it is
only a for a different use of exactly the some vehicle the owner may apply
a new number (for example "DB 261" was set to "DB 361" so that other staff
could be used on it).
Further more the same number may be reused when it's first "owner" has
been sold or destroyed.
So if the goal is to follow a vehicle over different changes applied to it
or to be sure over long term not to get confused with the numbers that
will be reused you better apply an independent unique number that contains
no description - and reuse it whenever giving data for that vehicle...
But how to make sure that the same unique ID will be reused whenever the
same vehicle is mentioned. At the moment given I don't see a good solution
for that.
Perhaps we could add an information like "previous GUID" to the schema so
that a chain could be build to follow the changes when the schema is used
for such purpose?

With regard to IP or MAC address: even if it's not perfect I can't find
better up to now.

Best regards
Thomas Kauer

Joerg von Lingen wrote:

> Hello,

> with regard to rolling stock there is no need to invent new GUIDs. Every railway operator use a
> numbering system for his rolling stock to indentify each single vehicle. Thus it would be simple to
> use the operator abbreviation, eg. SBB for Swiss Federal Railway, plus the operators vehicle number.

> The advantage would be, that such GUIDs are human readable - means everyone could understand what an
> identifier like "DB 425 201-1" stands for.

> With regard to IP or MAC address: up-to-date desktop computers are normally fitted with a network
> interface, but there are units without. And, as far as I know, even the MAC address can be
> manipulated by software.
> --
> Regards,
> Joerg von Lingen
> Institut fuer Bahntechnik GmbH, Dresden

> Nils Poldrack wrote the following on 05.04.2004 11:00:

>> Hello,
>>
>> Joachim Buechse from Ergon (CH) suggested on 2003-09-25 to use globally
>> unique
>> identifier for railML elements. He suggested to use the IP of the
>> creating computer concatenated with the milliseconds (e.g. since
>> 1970-01-01).
>>
>> In my "Instructions how to build a railML scheme" I adopted this proposal.
>> IPv4 and IPv6 create different spaces for their GUIDs.
>> Advantages of Joachim's proposal are:
>> - IP is easy to determine.
>> - The milliseconds since 1970-01-01 are easy to determine.
>> - There is not force to convert IPv4 to IPv6 or to pad the length.
>> - The saving format doesn't matter: HEX or Base64.
>> Disadvantages of Joachim's proposal are:
>> - No anonymity: via static IP it's possible to find the creator of the GUID.
>> - Not every computer has got an IP.
>>
>> Ulrich Linder made the suggestion to use the MAC instead of the IP:
>> there is anonymity and (almost) every PC has a network card and so a
>> MAC. (I don't know how to read the MAC - but this is a problem of the
>> software engineer.)
>>

>> Please write me your opinion. I prefer Joachim's GUID but using MAC
>> instead of IP. Thank you there much for your statement.
>>
>> Nils Poldrack
>> IVI Dresden
>>
>> PS. Sorry for my "rusty" English.