Subject: Re: Some problems with/questions about the infrastructure schema...
Posted by Wolfgang Keller on Thu, 17 Jun 2004 10:05:52 GMT
View Forum Message <> Reply to Message

Am Wed, 16 Jun 2004 17:49:26 +0200 schrieb Volker Knollmann:

> Wolfgang Keller wrote:
>> 1. Generally I propose that identifiers such as "type", "length", "value"
>> etc. should not be used at all, as they risk to collide with typical
>> reserved keywords.
>
> Do you speak of reserved keywords within your database-dialect (SQL,
> etc) or your programming language?

Reserved keywords in as many contexts as the RailML definitions will be typically used. Both databases and programming languages, and maybe also others.

>> It would be best imho if all identifiers were unique
>> within the entire schema in order to avoid confusion.
>
> Hmmm, I would prefer the opposite syntax. For example, I would
> appreciate a common attribute "elemID" for all elements. Right now, we
> have "ID", "elemID", "ocpID", "connectionID", etc. "elemID" is used in
> many children of <trackData>, why not everywhere?

Easy answer. >:-> For the sake of inobfuscation. Or, why make things more difficult than absolutely necessary. RailML is (considered to be) an open specification. Within bulkloads of code, it's easy to get lost, especially but not only for others than the individual who wrote the code. Instantly figuring out what all that code is about can be pretty difficult when everything is identified by an elemID. However, if you read trackID, you instantly know that the element in question is a track element. The same applies for all other identifiers. Having unique identifiers everywhere avoids lots of mistakes.

>> 4. Wouldn't it be useful/would it be impossible to include such
>> considerations as technology-independence in the design of the schema, so
>> that the logical structuring can also be used for plain-ASCII data exchange
>> (such as datagrams sent over narrow-bandwidth wireless connections etc.),
>> for relational databases and maybe also other implementations...?
>
> What exactly do you mean? A formal description like ER-Diagrams or similar?

No. I just propose to take aspects of technology-independence into consideration in the design process. This does not require actually creating and maintaining a unique relational database (or whatever else) schema, just designing the XML schemas in such a way that it is easily

possible to do so if someone wants to.

Just like XML, which was unknown ten years ago, other information representation technologies may come up in the future. If you take care of technology-independence now, you can take advantage of these later. Otherwise, most if not all the work invested may be lost.

> Imho XML is the best choice for our needs:
> * structured and hierachical

It's more "semi-structured", with very loose constraints on the structure, which makes it difficult to map XML schemas to more restrictively structured representations (such as for example, relational database schemas, but probably also others).

> I agree, that due to a certain redundancy within the file (opening /
> closing tags, ...) the file size is not optimal.
> But for the transmission over bandwidth- or volume-critical connections,
> you can apply $FAVOURITE_COMPRESSION_UTILITY to your data and you should
> get satisfying results...

Nope, sorry. The experts who are working on such topics know why they are doing things they way they do them. And they don't use zipped XML for their datagrams.

Under different conditions, different information representation technologies have different advantages. If you want to allow for automatic processing of data without extensive handwork, then you will have to take this into account.

Best regards,

Wolfgang Keller
