

EML - Bug #428

eml-constraint overlaps with packaging concepts

02/14/2002 05:15 PM - Matt Jones

Status:	Resolved	Start date:	02/14/2002
Priority:	Normal	Due date:	
Assignee:	Matt Jones	% Done:	0%
Category:	eml - general bugs	Estimated time:	0.00 hour
Target version:	Beta7	Spent time:	0.00 hour
Bugzilla-Id:	428		
Description			
<p>The current incarnation of eml-constraint allows the enumeration and definition of integrity constraints that apply to entities. These are currently drawn from the relational model, including UNIQUE, PRIMARY KEY, FOREIGN KEY, and CHECK constraints. It may also be extended to include other types of relationships between entities that are not part of the relational model.</p> <p>The "triple" element allows us to create arbitrary relationships between identifiable objects in EML, and is used for associating data with metadata, and groups of metadata and data objects together as a "package". This usage is very similar to the relational model, in that it allows us to define 3-valued tuples in a graph structure. Constraints between entities could conceivably be modeled using this infrastructure, probably with some modifications to the concept of a "relationship".</p> <p>So, the question arises. Should we try to develop a unified approach to the specification of constraints and the specification of packages? It might be more elegant, but possibly at the cost of simplicity and ease-of-use. My gut feeling is that this is not something we should pursue, but would like to hear other people's reasons for or against it.</p>			
Related issues:			
Has duplicate EML - Bug #427: eml-constraint use of identifiers		Resolved	02/14/2002

History

#1 - 02/15/2002 09:24 AM - Peter McCartney

I hope im using this right.

We grappled with this dilemma and at one point took the indecisive solution to have both a relation.xsd in which was based on ER Studios data model for describing relations between entities and a constraint.xsd based on the existing constraint.xsd. Foreign keys are both relationships and constraints, so this wasn't a very desirable solution. To be consistent with EML, weve basically dropped the relation module and were planning on using constraint since thats where things were going. The only other choice is to reduce constraint to only checks that reference the table's own fields and put all relational information in relation.

we also nested constraint under the table so that we dont need to rely on some kind of pointer to locate all the constraints that affect that table. one still needs a pointer however, to look up the referenced table. this makes it very fast to find all the tables that this table is dependent on, but a little more work to find all the ones that depend on it.

I dont have a strong preference over extending constraint vs re-adding relation.xsd other than to wind up with only one place where i scan to find all relationships. We should ask ourselves which question are you more likely to ask when using a dataset - how were inserts, updates and deletes to this table constrained? or how does this table join with other tables in the dataset? I think the former. another question we should ask is does the way we do this affect the potential use of EML in the future as a data modeling (as opposed to metadata) language?

at matt's request, im trying to create a merged version of this and other

modules for CVS that shows the differences that i outlined in the lengthy email i sent to the lter discus list, but if you want to quickly see how we envisioned these two modules as of december, you can look at:

<http://ces.asu.edu/bdi/subjects/metadata/december2001/dataset/>

#2 - 04/16/2002 04:46 PM - Matt Jones

The issues here are the same that were just addressed in bug#427. See the discussion there for details of the resolution.

- This bug has been marked as a duplicate of 427 ***

#3 - 03/27/2013 02:14 PM - Redmine Admin

Original Bugzilla ID was 428