LexEVS 6.1 Design Document - Detailed Design - Performance - Hierarchy Traversal

Contents of This Page

Document Information

Author: Bauer, Scott
Email: bauer.scott@mayo.edu
Team: LexEVS
Contract: ST12-1106
Client: NCI CBIIT
National Institutes of Heath
US Department of Health and Human Services

Revision History

Version	Date	Description of Changes	Author
1.0	2013/03/05	Initial Version	Bauer, Scott

Overview

LexEVS has long relied on a relational database to provide the data store for semantic assertions made about the entity level constructs in terminologies and ontologies. Recently it has become clear that graph database technology has matured enough to allow the the relationships between entities defined by these assertions to be stored in a way that better reflects the nodes and edges of these relationships. Benchmarking tests and practicality reviews have led the LexEVS team to the conclusion that a graph database back end for LexEVS associations will vastly improve traversal performance time and potentially simplify implementation of the association API.

Database Hierarchy Performance Evaluation

New technologies such as the MVRB-tree algorithm implmented in the OrientDB graph database have proved far more efficient and scalable than the traditional relational data base management system.

Graph Traversals

Time - mm:ss.ms (process time)

MySQL			
Depth	Small	Medium	Large
8	00:09.69	00:15.13	00:18.93
9	00:53.29	01:17.72	01:07.35
10	07:17.80	09:51.33	05:55.29

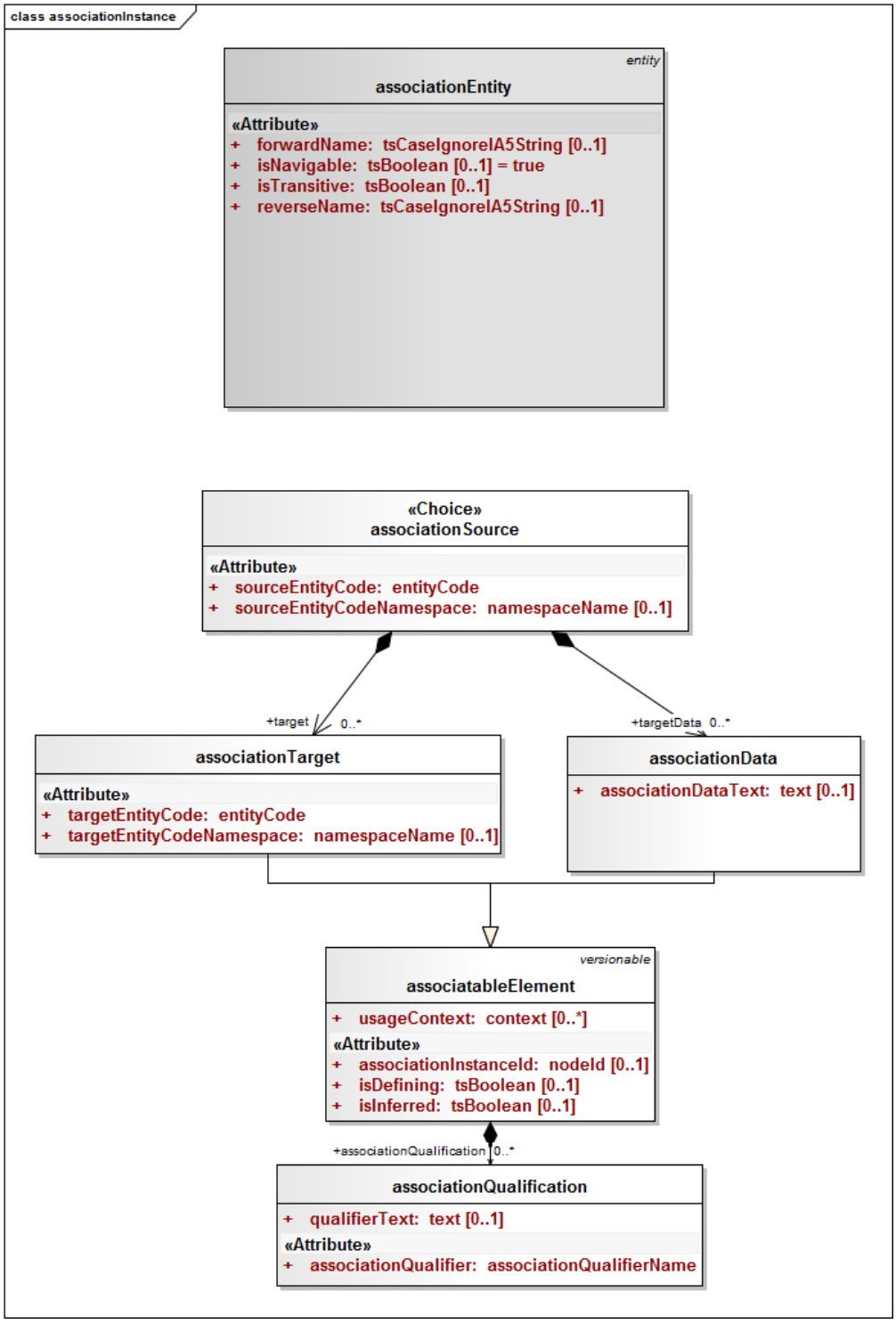
OrientDB			
Depth	Medium	Large	Huge
8	00:00.23	00:00.91	00:00.245
9	00:00.23	00:00.93	00:00.450
10	00:00.29	00:00.133	00:01.150

Small: 100 nodes/500 edges
Medium: 1,000 nodes/5,000 edges
Large: 10,000 nodes/50,000 edges
Huge: 1,000,000 nodes/5,000,000 edges

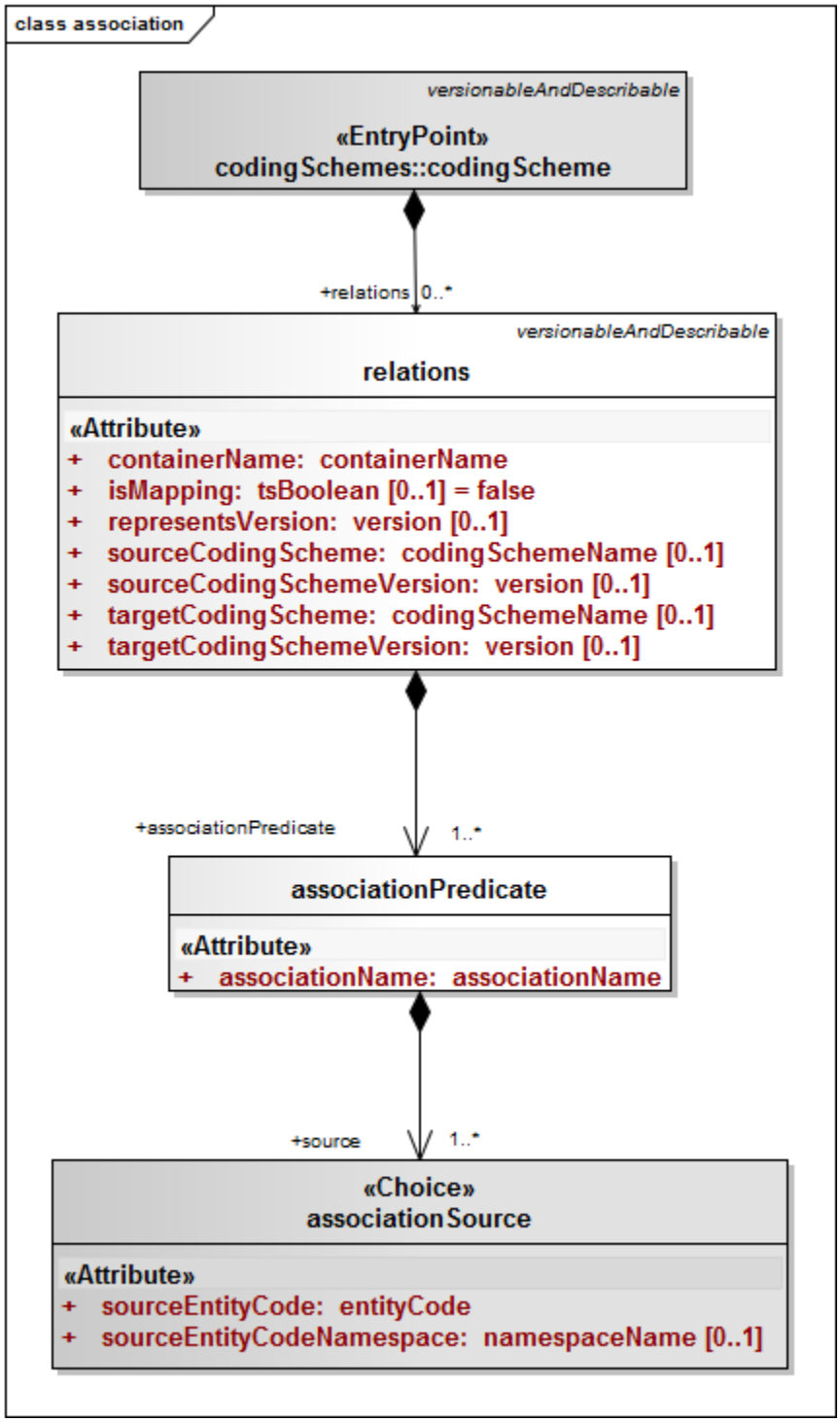
*** Process ran out of memory

LexEVS Association Logical Model

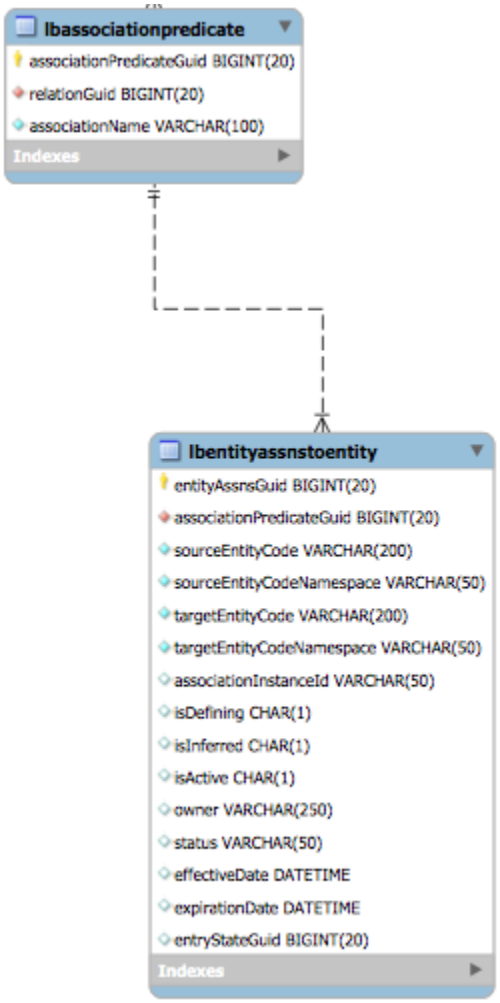
The LexGrid Model defines relationships in terms of a source and target node with an edge defined separately in the AssociationPredicate model element. These are the construction basics for larger coded node graphs which are currently represented in a relational schema. The performance restrictions of the relational schema have been well documented above. The source and target structure of LexGrid will be mapped to the structure of the higher performing graph database OrientDB.



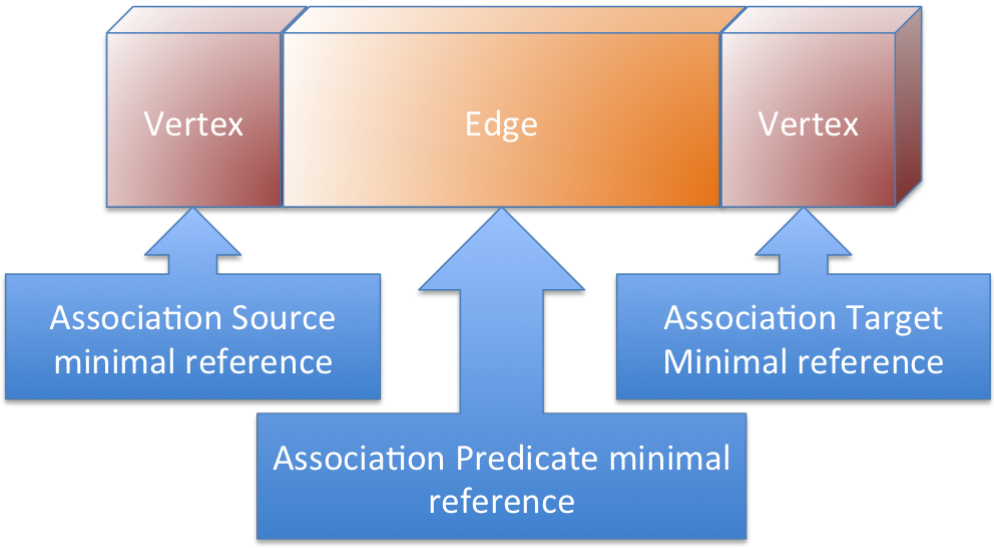
While the graph based database seems capable to handle the functions shown in the diagram above, some calls to LexEVS will continue to access some of the model elements that define metadata about the association.



LexGrid in the LexEVS schema (From the MySQL workbench)

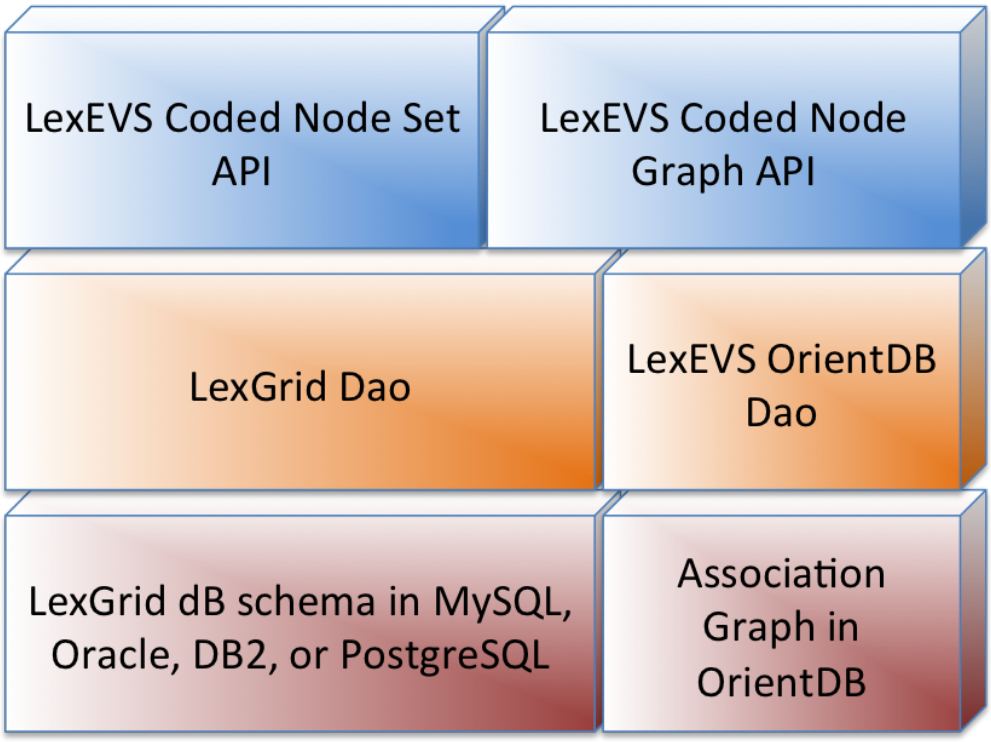


Mapping LexGrid data model elements to OrientDB



LexEVS Hierarchy Performance Architecture

While the new implementation of the node graph will largely run against the OrientDB service, some portions of the legacy LexEVS API will be needed to access various metadata and property elements.



Code Considerations

A CodedNodeFactory will determine whether this is an implementation that uses the graph database in conjunction with the relational database or a purely relational database. And a newly implemented DAO and OrientDBCodedNodeGraph provide the underpinnings of what will be a higher performance version of LexEVS' traversal of relationship hierarchies in stored terminologies.