

Software Engineering

CSE470

Spring 2002

Answer to Homework 4

First, the 1..* on the CompositeState (near the diamond) is wrong. Foreach StateVertex there is only one CompositeState, but for each CompositeState there can be any number of StateVertices (from which the CompositeState is composed). Also, the StateVertex and Transition ends of the aggregation from Dynamic Model should probably be 1..*.

Next, in database terms, aggregation is just like any other association and we implement it the same way. Below, "points to" means "contains the existence-id of some other object".

Schema for Dynamic Model Class diagram

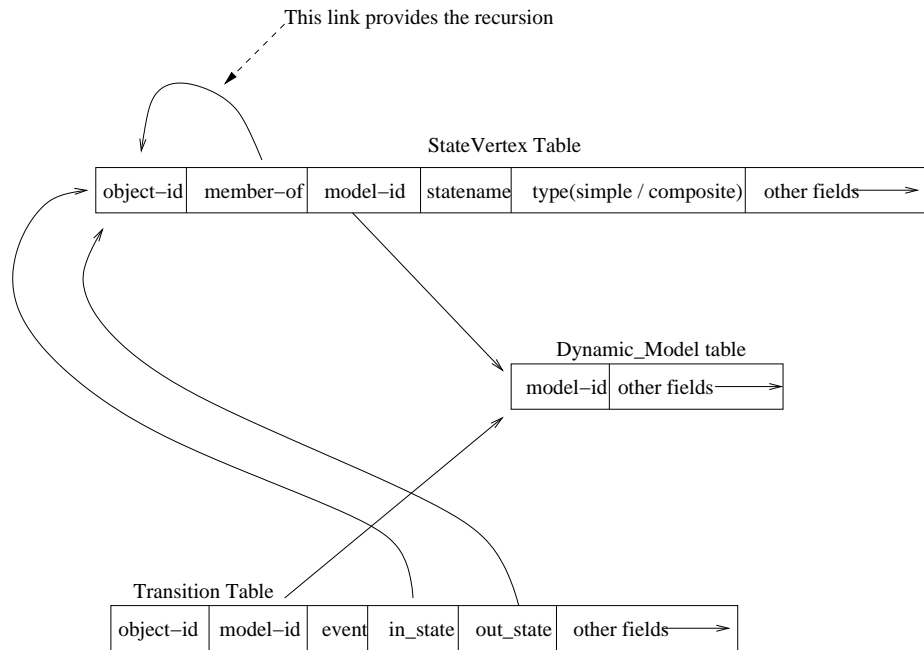


Figure 1: Schema for the Dynamic Model

Figure 1 show the schema for the class model given in the homework. Pointers have been added to show how the fields link.

In this case, there are no new attributes in the children of StateVertex, so we can make one table for StateVertex and reserve a "type" field containing either 'composite' or 'simple'. To connect the states to their composite states, we need a 'member-of' field that points to the parent, or composite, that this one is a member of. For example, if CS1 is composite and contains simple states S1 and S2, and composite state CS3, both S1 and S2's 'member-of' fields would point to CS1 (by containing CS1's object-id), as would CS3's 'member-of' field. If CS3 contained further states, those would point to CS3 by containing CS3's object-id.

To fetch the children of a composite state, say CS1, we join CS1's object-id with the StateVertex table's 'member-of' field (this is a StateVertex to StateVertex table join... a self-join). This fetches all rows such that the 'member-of' matches CS1's object-id... CS1's children. A clever join can probably retrieve the entire recursive relationship in one operation.

To get the parent of a state, S1, find the StateVertex row whose object-id matches S1's 'member-of' field.

Assuming we have more than one dynamic model, the Dynamic_Model table contains an object-id, plus whatever else we need. Each StateVertex and Transition row contains a pointer (call it model-id) to Dynamic_Model, for the model the components are contained within.

The transition linkage is provided by a foreign key in Transition pointing to StateVertex. There are two fields for this, one for 'in' and one for 'out'. We must allow for a null key here since one or the other (but hopefully not both) foreign keys can be null.