

BMC Software Inc.

Technical Disclosure Publication Document

Authors

Jeff Piazza

Posted: April, 2011

Overview

This document describes a solution/system/algorithm that uses metadata associated with class definitions to facilitate the export and import of a subgraph of a graph of objects.

Background

The data structures maintained by large applications such as the BladeLogic Automation Suite can be viewed as a large network of persistent objects. Each object (node) is a member of some object class, and the object classes form a hierarchy. From time to time it's necessary for objects from one environment to be transferred to another environment. (E.g., after testing is completed, a job definition and its supporting objects may need to be replicated from the testing environment to the production environment.) Since it's infeasible simply to extract and transfer the complete data set from one environment to the other, it's necessary to select subsets of the complete network to be exported from one application environment and subsequently imported into another.

The selected network subset may include graph edges that terminate on nodes not included in the exported subset. In general, the exporting process must decide which nodes and edges to include in the exported data, and which connected objects to avoid transferring. For objects (nodes) not exported, the exporting process must include sufficient information that the importing process is able to identify an equivalent pre-existing node in the receiving environment, and terminate the specified imported edge on this node. This invention solves this problem.

Solution

In the export process, metadata is retrieved for each node from the class(es) of which the node is an instance. The metadata identifies, for a given node, which edges originating from that node should be exported, and whether the object on which the edge terminates should or should not be included in the export set.

©Copyright 2011 BladeLogic, Inc.

For edges which terminate on nodes that are not to be included, the metadata also describe the construction of a function to run in the importing environment which will retrieve (or create) a correspondent node to which the importing process will resolve the edge. This description might typically take the form of a template for a function or query in an appropriate scripting language.

As an example, in the diagram below, node α , a member of class Alpha, is to be exported. The export metadata associated with the Alpha class tells the export process to export edge A, along with its endpoint, and edge B, without its endpoint. Edge C is not mentioned in the Alpha class metadata, and so is not exported.

Because the node on which edge A terminates will be exported, the export process recurs on that node, which may result in additional nodes being exported.

Although edge B is exported, the node on which it terminates is not. Instead, a function (in this case a query in a hypothetical query language) is constructed, instructing the importing process to retrieve a node matching the source terminal node's name and version. ("Name" and "version" are assumed to be attributes of the node/object.)

Drawings

