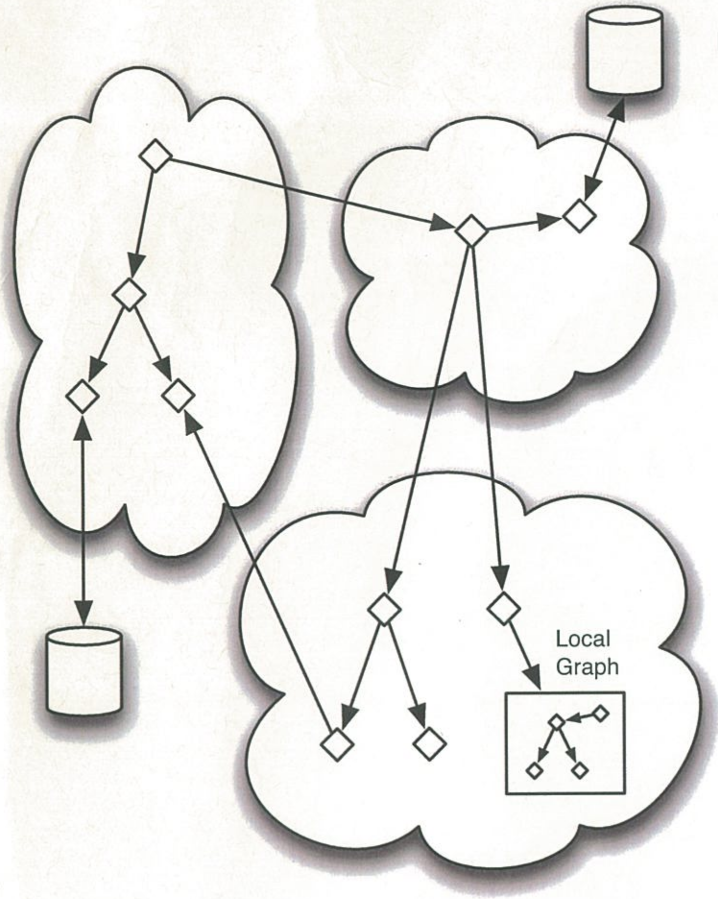
3Worlds Graph System, v26342



Logical Structure

- No Graph class
- sets of nodes are processed as TwIterable<Node>
- XML files are still loaded - but a TwIterable<Node> is returned
- a TwIterable<Node> can be generated using Node.traversal() - picks up all connected nodes
- Old Graph operations such as Visualisation etc. now operate on TwIterable<Node>
- Nodes are distributed over grid in accordance with a NodeHostMap
- Existing Node methods still work - e.g. getEdges()
- Existing Edge methods still work - e.g. getStartNode()
- Constraints (queries) still work
- Nodes make use of large data structures (databases, maps, even other graphs etc.) directly

Helper Classes

- *NodeList extends TwIterable<Node>* would include methods for IO (load from XML, visualisation) and applying processes to a set of nodes

Implementation

- Purple nodes are messaging Nodes
- Messaging nodes talk peer-to-peer once connected
- Each local node sees its edges and the nodes (actually message/stub node) they connect to
- A pre-defined set of messages is used to implement current Node and Edge methods
- Messages can be created by the user for domain specific applications

Advantages

- Graph is mutable - so, for example, UIs (described as a set of nodes) can be connected and disconnected from an existing graph at runtime
- No replication, synchronisation required
- Nodes talk peer-to-peer

Limitations

- Messaging Nodes are nodes - they are NOT Widgets, sims etc. So, a node can only talk to a remote node using Node methods. Of course, users can implement application specific messaging protocols.