It would be useful to allow TreeCtrl to manage user data too; that one would be easy if it could store and retrieve classes derived
from TreeCtrl::Node.

I don't know the internals of TreeCtrl, but adding a couple of :

TreeCtrl::Node *GetNodePtr(int id);
void SetNodePtr(int id, TreeCtrl::Node *node);

could do the trick from the interface side.
This would allow code like that one :


```
class MyNode : public TreeCtrl::Node
{
    MyData myData;
    MyNode(myData &data) { myData = data; }
};

TreeCtrl tree;
tree.Add(new MyNode(someData)); // takes ownership of node, similar to Array
......
MyNode *node = dynamic_cast<MyNode *>tree.GetNodePtr(id);
MyData data = node->myData;
```


This would greatly extend the usefulness of TreeCtrl, imho, using it to manage/reorder and so on custom data.

Max