There is actually nothing wrong with not providing a 'deep copy' functionality. In most (all?) languages with full object oriented programming support (which I don't include IDL in yet, for the lack of several key features) it is always the responsibility of the coder to provide a copy constructor.

I do this for all the IDL custom objects I create by considering it mandatory to implement a copy function that returns an instance of the copied object. That's no different from what is required in genuine OO languages.