

## Re: implementing roles in OOP.....

**Source:** <http://coding.derkeiler.com/Archive/General/comp.object/2004-03/1233.html>

---

**From:** Ilja Preuß ([preuss\\_at\\_disy.net](mailto:preuss_at_disy.net))

**Date:** 03/17/04

Date: Wed, 17 Mar 2004 13:11:07 +0100

You might want to take a look at the Role Object pattern (do a google search...).

Regards, Ilja

Mark Nicholls wrote:

> *I am having a problem with my coding style.*  
>  
> *I have been doing OOD/A/P for several years and over those years has*  
> *evolved to a very interface/delegation/pattern based.*  
>  
> *My problem is that basic text book implementation of 'roles' that an*  
> *object has to satisfy is by implementing the interfaces of those roles.*  
>  
> *This encourages bloated classes implementing more and more interfaces*  
> *to satisfy more and more roles and generally makes the implementation*  
> *of that class highly coupled between internal states and role.*  
>  
> *There is no context for the role i.e. so if I have an object which is*  
> *both a a member of two hierachies in two different contexts there is a*  
> *interface clash*  
> *e.g. a person is a member of a network of family members and a member*  
> *of network of work colleagues so....*  
>  
> *class CPerson : IFamilyNode,IEmployeeNode*  
> {  
> *bla bla bla*  
> }  
>  
> *but really I want to use a generic node implementation class in two*  
> *different roles here, the fact that I have two different contexts*  
> *shouldn't really be a problem.*  
>  
> *but I can't go*  
>  
> *class CPerson : INode,INode // interfaces clash*  
> {  
>  
>

comp.object: Re: implementing roles in OOP.....

- > }
- >
- > *I can ask the object for a reference to an interface to that role (and*
- > *it's completely up to the class how it implements that interface i.e.*
- > *by itself, by returning a contained object, by creating a new role*
- > *implementer.....).*
- >
- > *This gives a very different style of programming and a whole host of*
- > *questions.*
- >
- > *i.e. something like*
- > *class CPerson*
- > {
- > *INode AsFamilyNode()*
- > {
- > *hmmm now how shall I implement this role?*
- > }
- >
- > *INode AsEmployeeNode()*
- > {
- > *hmmm now how shall I implement this role?*
- > }
- > }
- >
- > *Is there any standard books/literature that discuss this style of*
- > *coding?*
- >
- > *There are all sorts of questions about identity, what is a role and*
- > *what is a class, should roles be able to have sub roles, should I be*
- > *able to add and remove roles, how should I process composite*
- > *structures of these roles, how can I create reusable role*
- > *implementation and compose them together etc etc etc.....*
- >
- > *I've sort of looked at web sites for "Role based modelling" and*
- > *similar subjects but they are thin on implementation – sticking to the*
- > *relative sterile simplicity of the modelling rather than the dirt and*
- > *mess of the real world implementation.*