Subject: Re: Urgent object question
Posted by Ted Cary on Thu, 24 Jan 2002 16:22:05 GMT
View Forum Message <> Reply to Message

Hi Davids,

I did not work out all the inheritance and keyword passing problems in detail, but I think (hope) my situation is simplified because each object in my container is of a different class. There's a one-to-one correlation between ISAs and HASAs in this case. I also (sort of) put the subclassed container object inside of the very container it subclasses, but I did this by passing an object reference to it as an argument to the Init method of every single object in the container. This means that the contained objects have to be written so that they know they are going to be contained in the subclassed container, but this was the only way I could figure out how to do the two-way communication I was speaking of before. If I'm doing something incredibly dense, please tell me--I created my first object ever only two weeks ago, and I need to finish this program soon... which is why I said "urgent" in the original message.

Thanks

Ted Cary
tedcary@yahoo.com

"David Burridge" <dave@clogic.f9.co.uk> wrote in message
news:oxV38.1473$k91.79975@wards...
> Hi Ted,

> The way I see it, using the IDL_Container (or a subclass if you need some
> specific
> behaviour) has a ton of advantages. The only problem with it is 1) you
need
> to be
> absolutely clear on the ISA vs HASA relationship and 2) you can have
get/set
> calls
> flying everywhere - causing infinite loops and making traceback
impossible!
>
> By only passing unresolved get/set keyword requests to the parent object,
> we've cut
> down on the tracking problem and eliminated infinite loops. This is
> relatively easy

> using the keyword inheritance methods. Secondly, we've buried all the
> mechanics
> in a single superclass (e.g. inheriting IDL_Container and passing
unresolved
> keywords to parent container objects) so that the hierarchy can be created
> almost
> transparently by simply inheriting our top-level object. Last of all, by
> accepting the
> parent class as a param to the init method, our object adds ITSELF to the
> container,
> adding to the transparency effect.
>

>
>
