Subject: Re: Urgent object question
Posted by David Fanning on Wed, 23 Jan 2002 05:01:21 GMT
View Forum Message <> Reply to Message

Ted Cary (tedcary@yahoo.com) writes:

> Does anyone have experience with a problem like this? Or any
> suggestions? Fanning?

Uh, funny you should mention this. I am up to my
eyeballs in a problem almost exactly like this one. :-(

Let me get back to you when I have a little more
time and I see the path just a little clearer than
I do at the moment. But I've been fooling around with
container objects and object hierarchies that communicate
with one another in a way that is analogous to the way
widgets communicate. But it is all still a bit unsettled....

Cheers,

David
--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Urgent object question
Posted by Ted Cary on Wed, 23 Jan 2002 05:11:15 GMT
View Forum Message <> Reply to Message

Mmmm... IDL_Container? Could I just subclass an IDL_Container and my "planet" object widget, then fill the container with all my "satellite" object widgets? If each satellite had an object reference to the planet or (better) to this very subclass containing the IDL_Container, then all of them could change the planet object's fields or get its fields via Set/GetProperty methods. Then all that's left is to override the planet object's methods so that they also call methods on every object in the IDL_Container, and there's my two-way communication.

This also solves the problem of destroying the object widgets when their TLBs are destroyed by the user. In my first scheme, the XManager Cleanup procedure for each satellite would have to destroy the object part of the object widget , a problem if the object it is trying to destroy has been subclassed! The XManager-called cleanup routine would have to test the

object before destroying it, and then if the object was subclassed it would have to reset all the corresponding self fields that were inherited from just its object. Using the IDL_ Container, the satellites' self fields are each in separate objects within the whole "planetary system" object, so they can be destroyed along with their object and with the TLB of their Widget interface.

It makes sense now, but I'm also very tired--probably this is what IDL_Containers are used for all the time. Any obvious problems with this system before I start coding it tomorrow, besides the fact that I keep calling my objects planets and satellites? (If there are more formal programming terms, tell me.)

Thanks.

"David Fanning" <david@dfanning.com> wrote in message news:MPG.16b7f18d835b7a669897d9@news.frii.com...
> Ted Cary (tedcary@yahoo.com) writes:
>
>> Does anyone have experience with a problem like this? Or any
>> suggestions? Fanning?
>
> Uh, funny you should mention this. I am up to my
> eyeballs in a problem almost exactly like this one. :-(
>
> Let me get back to you when I have a little more
> time and I see the path just a little clearer than
> I do at the moment. But I've been fooling around with
> container objects and object hierarchies that communicate
> with one another in a way that is analogous to the way
> widgets communicate. But it is all still a bit unsettled....
>
> Cheers,
>
> David
> --
> David W. Fanning, Ph.D.
> Fanning Software Consulting
> Phone: 970-221-0438, E-mail: david@dfanning.com
> Coyote's Guide to IDL Programming: http://www.dfanning.com/
> Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Urgent object question
Posted by David Fanning on Wed, 23 Jan 2002 15:06:35 GMT
View Forum Message <> Reply to Message

Ted Cary (tedcary@yahoo.com) writes:

> Mmmm... IDL_Container? Could I just subclass an IDL_Container and my
> "planet" object widget, then fill the container with all my "satellite"
> object widgets? If each satellite had an object reference to the planet or
> (better) to this very subclass containing the IDL_Container, then all of
> them could change the planet object's fields or get its fields via
> Set/GetProperty methods. Then all that's left is to override the planet
> object's methods so that they also call methods on every object in the
> IDL_Container, and there's my two-way communication.

That is the basic idea, yes. We are in the process
now of building a complicated enough system to test
whether it works or not. For us, the bottom line is
whether this system gives us advantages in maintaining
and extending large applications.

> This also solves the problem of destroying the object widgets when their
> TLBs are destroyed by the user.

Yes, containers have many advantages and this is certainly
one of them.

> It makes sense now, but I'm also very tired.

After a solid week of this we find that our ideas
always look wonderful about midnight. But not all
of them have survived the bright glare of morning. :-(

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Urgent object question
Posted by mvukovic on Wed, 23 Jan 2002 15:20:32 GMT
View Forum Message <> Reply to Message

"Ted Cary" <tedcary@yahoo.com> wrote in message news:<a2lk23

... lots of stuff deleted

Folks,

My ``bible'' for complex object behavior is ``Design Patterns'' (you can find it on Amazon). It merits very carefull study.

Code that uses those ideas of associations and inheritances can result in very complex behavior, and thus requires extensive documentation. It is also sometimes a pain in IDL, because you need to recompile several objects in order to get your object to accept the latest editorial change.

Mirko

Subject: Re: Urgent object question
Posted by Ted Cary on Wed, 23 Jan 2002 15:42:34 GMT
View Forum Message <> Reply to Message

"David Fanning" <david@dfanning.com> wrote in message >

> After a solid week of this we find that our ideas
> *always* look wonderful about midnight. But not all
> of them have survived the bright glare of morning. :-(
> ... We are in the process
> now of building a complicated enough system to test
> whether it works or not.

Morning is almost over, and it still seems to be a good idea. If you want a not-so-complicated system to test, I can send you guys mine. I'm sure the boss would be happier if *real* programmers wrote my application. Hopefully you and your friends will publish the results on your web page, so that I will know what I've done wrong. Who's the "we" you speak of anyway--I hope you haven't started referring to yourself and your computer in the first person plural...

Subject: Re: Urgent object question
Posted by David Fanning on Wed, 23 Jan 2002 17:11:13 GMT
View Forum Message <> Reply to Message

Ted Cary (tedcary@yahoo.com) writes:

> Morning is almost over, and it still seems to be a good idea. If you want a
> not-so-complicated system to test, I can send you guys mine. I'm sure the
> boss would be happier if *real* programmers wrote my application.

Well, at least until the bill arrived. :-)

> Who's the "we" you speak of anyway--I hope
> you haven't started referring to yourself and your computer in the first
> person plural...

I told you I need to get a life!

Actually, I've had to engage a real programmer myself,
so I can get on with the tennis. :-)

Cheers,

David

--
David W. Fanning, Ph.D.
Fanning Software Consulting
Phone: 970-221-0438, E-mail: david@dfanning.com
Coyote's Guide to IDL Programming: http://www.dfanning.com/
Toll-Free IDL Book Orders: 1-888-461-0155

Subject: Re: Urgent object question
Posted by David Burridge on Thu, 24 Jan 2002 15:16:09 GMT
View Forum Message <> Reply to Message

Hi Ted,

Ted Cary wrote in message ...
>
> "David Fanning" <david@dfanning.com> wrote in message >
>
>> After a solid week of this we find that our ideas
>> *always* look wonderful about midnight. But not all
>> of them have survived the bright glare of morning. :-(
>> ... We are in the process
>> now of building a complicated enough system to test
>> whether it works or not.
>
> Morning is almost over, and it still seems to be a good idea. If you want
a
> not-so-complicated system to test, I can send you guys mine. I'm sure the
> boss would be happier if *real* programmers wrote my application.
Hopefully
> you and your friends will publish the results on your web page, so that I
> will know what I've done wrong. Who's the "we" you speak of anyway--I hope

> you haven't started referring to yourself and your computer in the first
> person plural...
>

I just got back from that week of programming Dave mentioned, so forgive the delay in sticking my oar in!

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.

As Dave said, this idea is still bedding down. I wrote a little test app which proved the
concept, now we're trying to break it!-) Hope this garble is understandable and helpful!

Dave
