Subject: Re: passing parameters from base to base
Posted by Stein Vidar Hagfors H[2] on Tue, 03 Dec 2002 17:14:52 GMT
View Forum Message <> Reply to Message

"Pavel A. Romashkin" <pavel_romashkin@hotmail.com> writes:

```
> JD Smith wrote:
>>
>> In an ideal system, ObjMsg objects would find each other and setup
>> their own intercommunication themselves.  You get into lots of
>> chicken-and-egg dependency issues in this case though.
>
> This is exactly what I was talking about :-)
```

I may not be getting exactly what you're talking about, but this is exactly where I think a singleton might be useful: You can always "find" it (it'll be created & initialized if it doesn't exist!) as long as you know it's name, e.g. (pardon any mis-programming here, I haven't been keeping up with objects in daily life!)

```
   ;; Locate the message center
   dummy = obj_new("message_center",object=MsgCtr)

   MsgCtr->Register,self,"mainprog",uniqkey  ;; Key allows multiple instances
                                             ;; of mainprog

   ;; The LocateObjMsg could generate a proxy object if ObjMsg
   ;; hasn't registered yet!  The key makes sure we don't pick a
   ;; wrong instance of ObjMsg (that has already been connected
   ;; with another self-class object)

   objmsg = MsgCtr->LocateObjMsg("ObjMsg",uniqkey)

   if objmsg->isproxy() then begin
      objmsg->queue_state,/on           ;; Keep all messages I send
      objmsg->inform_me_when_registered,self,"through_this_method "
   end

   objmsg->send_message,self,"Send me data anytime","through_this_method"
```

Etc, etc.. Now, when "ObjMsg" registers with the message center, it (MsgCtr) will generate a call to the above object's "through_this_method" method, with information that the proxy "objmsg" is now replaced with a true ObjMsg instance, with so-and-so unique key. It will then dump all the queued messages from the proxy onto the registering ObjMsg (maybe triggered by ObjMsg calling a MsgCtr->deliver_pending_messages method, gives you more control over whether or not the object is fully initialized).

The number of variations on this scheme is endless, and details will differ according to what kind of messaging system is already implemented. However, it sketches out a method to deal with chicken-and-egg problems.

One key problem might be how to use the unique keys (e.g. system time at object creation) to enable creation of multiple independent interlinked systems (e.g., two display windows with separate sets of linked controls). Only a half-thought-through solution, but I don't see that you have problems that cannot be solved by this general approach

--

--- --------------

Stein Vidar Hagfors Haugan
ESA SOHO SOC/European Space Agency Science Operations Coordinator for SOHO

NASA Goddard Space Flight Center,   Tel.: 1-301-286-9028
Mail Code 682.3, Bld. 26, Room G-1,   Cell: 1-240-354-6066
Greenbelt, Maryland 20771, USA.   Fax: 1-301-286-0264

--- --------------