Subject: Re: REDUCE
Posted by Kenneth Mankoff on Fri, 30 Mar 2001 07:30:45 GMT
View Forum Message <> Reply to Message

On Fri, 30 Mar 2001, John-David Smith wrote:

> Kenneth Mankoff wrote:
>>
>>> The question, to all you C-programmers: is there a better way?
>> [snip]
>>> ...the code logic to compute the maximum will be the same, both
>>> symbolically for all types for many types, in the compiled code itself.
>>
>> Hi JD,
>>
>> hmmm... not 100% sure, but wouldn't c++ templates solve this problem?
>>
>> And for the cases where it is "symbolically" the same but not "compiled
>> the same", I'm not sure what this means, but I'm guessing you would handle
>> these cases with overloading your operators.
>>
>> Of course, C isn't C++, so this might not help.
>>
>> I can provide code examples and more info if you wish.
>

> Thanks for the suggestion. I had thought of that option, but I don't
> know much about templates, nor about linking C++ to IDL.

I think you can just put the regular C code in there to connect to the
client. Sorta like writing C+.

> I wonder whether the templates are just similar to my super macro for
> creating a different version for each type. Can you frame the maximum
> function I suggested in terms of a skeleton template which would operate
> on all the data types?

```
// function definition
template<class idlType> idlType maximum( idlType v0, idlType v1 )
{
   if ( v0 > v1 )
      return v0;
   return v1;
}

// function instantiation
int main(void)
```

```
{
   float f = maximum( 4.2, 4.3 );
   int   i = maximum( 43, 42 );
}
```

I thought for "symbolically" and "logically" the same, you were referring to comparison of strings or objects or structs or other types, where the standard ">" won't work, and you need "strcmp()" or something else.

-k.

Kenneth Mankoff LASP
mankoff@lasp.colorado.edu 1234 Innovation Drive
303.492.3264 Boulder, CO 80303
