X3J16/96-0075 WG21/N0893

Arguments against extending ->*

Erwin Unruh
Siemens Nixdorf, Germany

I object to change the semantics of ->* as proposed by the working group. The reasons of that objection are:

1. The need was not demonstarted sufficiently enough. On private conversation it was expressed to me that the library was asked to add ->* to iterators and auto_ptr, but they would not object to omitting that operator.

2. The paper used as base is not specific enough regarding the needed WP changes. During drafting committee we filled up most of the holes. There is the possibility that we have missed some point. We also had to make some design decisions on the fly. Whether they are correct was not discussed in a working group, but only between a few people during drafting.

3. The special handling does not fit into the reasons of special handling of other special operators. For the other operators we have very specific reason to have them being handled differently:

operator =	because it is generated implicitly
operator->	because the right hand side is no expression
operator()	because we already allow classes as parameters to functions.
operator[]	because it appears like a weighted operator
operator,	because it already has meaning for any pair of types.
operator &	because taking the address is a default operation.

Just adding operator ->* because it makes life easier looks out of place.

The ARM, where we have this list from, explicitly specifies that the usual rules apply for ->* (pages 332 and 338). So the present rules do have some justification.

4. There was no discussion about the real underlying problem and other possibilities to solve it. The real problem may be the lack of support for a *bounded member function* within the current language and library. A proposal of this week does add a first step in this direction (Bjarnes Mem_fun classes). No work was done to check the possibilities in this direction.

Also there were no discussions about other forms of how to extend ->* in a reasonable way (maybe using operator-> ?). There is no discussion of whether a version with the right hand side of ->* having special treatment is a good idea.
We do have some experience for writing smart pointers, but to my knowledge nobody tried to write a *smart pointer to member* class. Such a class would have to handle ->* correctly and may benefit from a different resolution for ->*.

Overall, I think this a not well thought through. It attempts to provide a fix for a library problem without really analysing the ramifications, alternatives and merits.

I propose to withdraw the motion on friday.

Dateiname:	ERWINARR.DOC
Verzeichnis:	C:\ERWIN
Vorlage:	C:\MSOFFICE\WINWORD\VORLAGEN\NORMAL.DOT
Titel:	Arguments against extending ->*
Thema:	
Autor:	Borland Loaner
Stichwörter:	
Kommentar:	
Erstelldatum:	14.03.96 12:55
Version:	4
Letztes Speicherdatum:	27.03.96 08:12
Zuletzt gespeichert von:	Erwin Unruh
Letztes Druckdatum:	27.03.96 08:18

Nach letztem vollständigen Druck

Anzahl Seiten:	1
Anzahl Wörter:	383 (ca.)
Anzahl Zeichen:	2.186 (ca.)