

This article describes an R2CORBA success story in the heterogeneous setting of the control system for CARMA, an array of radio telescopes in southern California.

CARMA is typically used for 3 mm and 1.3 mm interferometry. To increase its sensitivity for participation as a VLBI station for the Event Horizon Telescope, we built a "phased array processor" (aka beamformer). This new hardware system, developed independently from the rest of CARMA, is controlled via software mostly written in Ruby.

Interfacing the phased array processor with the CARMA control system, which does not use Ruby, could have posed a serious challenge, but fortunately the CARMA control system uses CORBA and Ruby now has R2CORBA! Most of the control system is written in C++ and uses TAO, but the user interface is written in Python and uses omniORB. To interface with the CARMA control system, we needed a Ruby ORB that could perform the following:

- Participate with the ImR, which is used to startup all the CARMA processes across a multitude of hosts.
- Bind and resolve names to objects in the NamingService.
- Call into remote objects implemented in C++.
- Publish objects that can be called from both C++/TAO and Python/omniORB.
- Compile a fairly large collection of IDL files (80 files, 10K+ lines of code, including some deprecated(!) constructs).

Early releases of R2CORBA handled most of these requirements. Thanks to the fantastic response from the great people at [RemedyIT](#) (the creators of R2CORBA), subsequent releases of R2CORBA met all of our needs for this nontrivial heterogeneous setting. Our Ruby scripts became full-fledged participants in the federation of distributed objects that comprise the CARMA control system!

It's great to have such a robust CORBA binding for Ruby and it's great to have such a robust scripting language for CORBA. David MacMahon Radio Astronomy Lab University of California, Berkeley - See more at: <http://orbzone.org/node/151#sthash.sRKJXMjm.dpuf>

David MacMahon
Radio Astronomy Lab
University of California, Berkeley