P2.15 THE DISTRIBUTED RADAR DATA ACQUISITION AND CONTROL SYSTEM (DIRAC) - A NEW GENERATION OF MULTI-CONTROLLER NETWORKS RUNNING AN OPEN RADAR CONTROL SOFTWARE

Lothar Wörhoff *, Stefan Beyer, Klaus Mähner
Gematronik GmbH, Germany

1. INTRODUCTION

In 1998, in the course of the COST 75 seminar in Locarno, Gematronik has introduced the new concept for an open design of a heterogeneous Radar data acquisition, control and monitoring system. Now, after being implemented in Gematronik's new METEOR 500/1500 Radar series, the first results can be reported.

2. BACKGROUND

Modern radar meteorology raised demanding requests for an enhanced control and data flow as integral part of a radar system. The main objectives are: a flexible integration of additional internal sensors in order to measure additional radar- and associated operational parameters (extended evaluation and research, simplified maintenance), scalable and customized systems based on a flexible modular design, support of component upgrades and interfacing with third-party hard- and software as well as network integration of radars and product distribution via Internet/Intranet.

The recent developments in data/signal processing and network/web technologies provide the opportunities for a migration to highly flexible, open configurations. Those systems rely on hardware-independent, embedded operating systems, programming languages and software tools running on distributed COTS (Commercial off-the-shelf) hardware components. Single-platform radar controllers that make use of proprietary, in-house developments are regarded more and more as too restrictive.

3. CONCEPT

3.1 Distributed controlling devices

How to bring the right level of computing power into the individual modules to ensure a strict modular system design with scalable performance? How to serve realtime requirements and integrate heterogeneous controllers and sensors in order to run decentralized, simultaneous processing tasks? - The answer is a new design of a radar control network: *Dirac* - Distributed Radar data Acquisition and Control

* Lothar Wörhoff, Gematronik, Neuss, Germany;
e-mail: L.woerhoff@gematronik.com

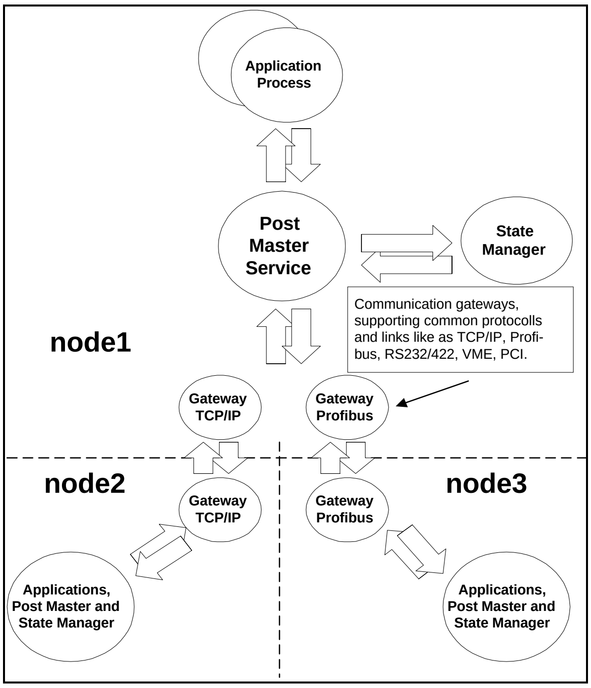
3.2 Radar control network architecture

Each module of the radar (transmitter, receiver, etc.) is equipped with its own controller. These controllers are implemented by using COTS devices only.

In order to establish a transparent access for any application via the Dirac network to each of those controllers, the software architecture reflects them as nodes, consisting of a 'State Manager', a 'Post Master' and several 'Gateways'.

The State Manager acts like a database that is equipped with an additional news server. Any request for information which is raised by an application program (e.g.: 'current antenna position?') is handled by the relevant State Manager (e.g. the manager for the Antenna Control Unit). The communication between several nodes is handled via a backbone that consists of one State Manager per node and several types of gateways. The gateways are acting as 'bridge' between two nodes.

The Post Master performs the routing of the messages. Based on a global routing table the Post Master is capable to route any request through the entire Dirac network.



3.3 How to communicate? - The macro language RCL

R(adar) C(ontrol) L(anguage) - An open specification of status/command syntax allows to communicate from 'everywhere to everyone'. Each request for information or control status changes as well as the corresponding system response messages are formulated in RCL. The application programs have transparent access to every module in the radar network. They do not need to know the source that administrates the relevant information or the way how to establish the physical communication to that node. The necessary routing of the messages is handled by the subordinate Dirac network.

In order to fulfill the typical communication requirements of common radar application programs, RCL posses an extensive set of commands.

Example:

a) a request for the actual antenna azimuth position '*GET azposreal*' is answered by '*SEND azposreal 145.3',*

b) to set a new antenna azimuth position : '*SET azposreal 120.0*',

c) for continuous updates of the antenna azimuth position: '*GETNEWS azposreal'.*

In multi radar networks the RCL commands are extended by a radar address prefix:

Example:

a) '*radar1 GET azposreal'* sends *a* request for the antenna azimuth position of radar 1;

b) '*radar2 GET azposreal*' sends a request for the antenna azimuth position of radar 2.

Another major benefit of the RCL language structure is the RCL script capability. RCL commands can be embedded into command macros by using conditional statements like e.g. loops or conditional jumps. Therewith complex procedures like e.g. calibration routines can easily be automated. A RCL Script interpreter is an integral part of the METEOR radar software and it is used for any standard calibration and test procedure within the Gematronik weather radar systems.

The RCL format description can be made available to interested users (please refer to: www.gematronik.com). Based on this easy-to-use syntax any proprietary or third-party application which uses this communication front-end can access any information in the Dirac network.

3.4 User Interfaces - Realtime visualization of control and data flow with Ravis®

A consequence of the heterogeneous network architecture is the usage of platform-independent, non-proprietary implementation techniques for the applications on the presentation layer. Experiences have now been made with Java for realtime visualization and control of individual radars or radar networks. This is already the basis for the Control and Maintenance software Ravis®. Ravis® is fully programmed in Java by using modern network and web technologies. It offers full remote sensing and monitoring capabilities to any radar sensor that talks RCL. Ravis allows to run a multi-radar, multi-user network as a kind of Intranet solution. An appropriate level of access security is ensured as well.

Ravis®, a state-of-the-art tool for radar realtime visualization and control, is one example for an application program that communicates within a Dirac network. It is able to login to any Dirac network node via the standard TCP/IP protocol. Ravis® can access any information that is stored in the network itself. By using the RCL states that describe the configuration of the radar system, Ravis® may run its self-configuration routine during start-up. The software identifies the functionality of the radar system automatically and adapts its menu options accordingly. Therewith Ravis® requires minimum knowledge and effort when being installed in an existing network for the first time.

4. RESULTS AND BENEFITS

Dirac provides comprehensive remote access capabilities

- remote visualization, configuration and maintenance can be performed from any node within the LAN/WAN network.
- it enables online surveillance of system parameters during operational data acquisition

Dirac relies on an extremely flexible, scalable design:

- functionality and performance can be customized to fulfill the specific needs for an individual radar or a whole radar network.
- The auto-detection of the radar system configuration facilitates the system setup and maintenance procedures.
- remote online interaction or even live software upgrades from remote side are possible at any time.

Dirac is a real basis for an open system architecture:

- Interfacing of third party radar system components that uses RCL as well as modular upgrades require minimum integration effort.
- Interfacing of user-specific control and analysis software can be provided via an 'RCL communication shell'. Therewith additional radar control procedures or extended information exchange can be integrated. Standard application software like e.g. LabView might be used for these purposes as well.