5.7 THE WSR-88D COMMON OPERATIONS AND DEVELOPMENT ENVIRONMENT - STATUS AND FUTURE PLANS

Thomas J. Ganger*
Mitretek Systems, Falls Church, Virginia

Michael J. Istok
NWS OS&T, Silver Spring, Maryland

Steve Shema
FAA ROC Deputy, Washington, DC

Bill Bumgarner
BAE Systems, Washington, DC

1. INTRODUCTION

The Common Operations and Development Environment (CODE) is sponsored by the National Weather Service (NWS) and the Federal Aviation Administration (FAA) through the NEXRAD Product Improvement (NPI) Program. The objective is to shorten the development, integration and validation cycle for new algorithms and applications on the WSR-88D (Ganger, 2000).

The Open Radar Product Generator (ORPG) includes a system architecture that provides an opportunity to increase the pace of technology infusion, including the deployment of new algorithms, into the operational weather radar. The use of the ORPG's Common Operations and Development Environment (CODE) for algorithm research and development, would allow those development activities to produce software that can more easily be integrated into the operational ORPG, thus broadening participation in algorithm development, test, and evaluation activities for the WSR-88D.

The recent versions of WSR-88D CODE support basic algorithm analysis, development, test and integration activities for algorithms targeted for Open Build 2. CODE is already being used by several organizations for WSR-88D algorithm analysis and development; Istok (2002). Not all portions of the guidance provided for algorithm development have been completed, and some enhancements to the ORPG's algorithm interface and other services are needed to increase productivity and shorten the learning curve. This paper summarizes the current status of CODE, what remains to be done, and provides a vision for the future of CODE in support of WSR-88D algorithm development.

2. ORPG CODE SUPPORT FOR ALGORITHM DEVELOPMENT

* *Corresponding author address*: Tom Ganger,
Mitretek Systems, 3150 Fairview Park Drive South,
Falls Church, Virginia 22042-4519
e-mail: tganger@mitretek.org

2.1 *What is ORPG CODE*

ORPG CODE is a WSR-88D algorithm development environment. CODE does not support algorithm development for other platforms nor does it support ORPG software maintenance in general. However, CODE does aid in ORPG maintenance by defining an API that, if followed, keeps the interface between the algorithms and the rest of the ORPG consistent and narrow. This will permit the evolution of ORPG services while maintaining algorithm integrity by localizing the affect of changes to a few libraries comprising that interface.

2.2 *Support for Algorithm Development*

The CODE package includes several components. The first component is the source code distribution for the ORPG. This includes the complete source code and instructions for the configuration of a Sun workstation for installing, compiling, and running the ORPG. The existing ORPG services that are of interest to the algorithm developer are provided by three libraries that are called the "Algorithm API" in this paper. The ORPG provides two bindings for these services. A FORTRAN binding which provides the interface for the ported legacy algorithms and an ANSI-C binding that can be used to write new algorithms in C. Planned enhancements to this API are discussed later in this paper.

The second component is an ORPG Application Development Guide. This guide includes an overview of the ORPG architecture, a guide to the ORPG development environment, and documentation of internal data structures (e.g., base data and final graphic products). The development environment guide provides instructions for using the ORPG makefiles, compiling software, configuring the ORPG in order to add data stores and tasks for new algorithms, and properly integrating development source code into the ORPG directory structure. Recently, a guide for algorithm adaptation data was added to CODE. This guide included an overview of existing algorithm adaptation data, instructions for defining new data and configuring the

ORPG to recognize the new adaptation data.

The third component includes ORPG specific analysis tools and guidance for writing WSR-88D algorithms. The algorithm writing guide includes: documentation of the ORPG Algorithm API, guidance for the proper structure of data driven and event driven algorithms, and sample algorithms demonstrating use of the API. Currently, CODE includes two product analysis tools. One provides a textual output of both the header portions of a graphic product and the data packets, with options to format the data with raw or scaled values. The other tool provides a graphical display of the various data layers within a product. For an update on the analysis tools see Stern (2002).

3. FILLING THE GAPS IN ORPG CODE

Though the current development environment can support the development of algorithms that correctly use the WSR-88D ORPG applications services, there is more to be accomplished.

3.1 *Guidance to Algorithm Developers*

The most significant omission in guidance are instructions for the preparation of an algorithm for handoff to the Radar Operations Center (ROC) for integration into the operational system. This document would include coding guidelines and requirements for the proper structure of algorithms and use of API services. Guidance for algorithm testing and documentation of testing need to be defined. Finally, a comprehensive algorithm documentation package must be described so that all information needed to update formal WSR-88D documentation can be provided.

3.2 *Algorithm API*

Though the existing ANSI-C API services are sufficient to meet the needs of the vast majority of WSR-88D algorithms, there are limitations. Some of the planned enhancements are discussed here.

Base data ingest is provided only as individual radial messages. An enhanced API should also provide services to ingest base data via complete volumes and elevations in addition to just individual radials. This enhancement can easily be developed from existing services.

Another major area for API enhancement is improved support for correct assembly of the final product to comply with the Interface Control Document (ICD). Existing services automatically insert most of the header type information but offer no support for placing the data into the appropriate data packet structure.

Currently, there are no standards for data structures for either intermediate product formats or data structures internal to algorithms. An enhanced API should define standards for intermediate products and provide a recommended internal data structure. A standard internal data structure would facilitate the creation of a reusable data manipulation / science library.

This data manipulation / science library, or common calculations library, should not be a statically defined library (in the sense of being complete) like the rest of the algorithm API. Rather, this library should evolve and be expanded as reusable components are developed.

3.3 *Other ORPG Services*

In addition to the Algorithm API services, other ORPG software services must be modified. Currently, the ORPG configuration files (adaptation data) provide the capability to define new data stores (products) and ORPG tasks so that the ORPG can generate the products based upon a routine generation list. In order to actually distribute new products and generate new products based upon product requests, ORPG source code must be changed. These services should be modified so all aspects of configuration for new algorithms are handled via configuration files / adaptation data.

4. CODE AND THE ALGORITHM INSERTION PROCESS

In the past several decades meteorological & hydrological algorithms have been developed at numerous research laboratories for varied Doppler weather radars. When algorithms were ready for operational baseline use, the algorithm science was transferred to the ROC where standardized descriptions of the algorithm were produced, the algorithm logic was recoded, baseline documents were updated, and the algorithm science was further tested in an integrated baseline system. This process could typically take 18-24 months depending on complexity and available ROC resources.

The ROC and two NEXRAD agencies, the NWS and the FAA, are exploring a new process for the WSR-88D algorithm development and integration activities. An early focus is to define the nature of the algorithm "package" (documentation and source code) that must be provided to the ROC. This includes the guidance to algorithm developers discussed in paragraph 3.1. This paper presents a basic description of the early phases of development that would be included in such a process. Details have not been worked out and some organizational responsibilities are still under consideration.

4.1 *Applied Research Phase*

Although not mandatory, it is strongly encouraged that all algorithm development be performed using an

ORPG clone running the latest ORPG baseline software (i.e., CODE).

A researcher / developer would be responsible for creating and thoroughly testing a new algorithm. Developmental testing should be performed with a wide range of data sets which would rigorously test this new algorithm.

If practical, real-time performance testing should be conducted as well to better define the operational capability of this algorithm. This capability would best be demonstrated using an ORPG clone interfaced to the Base Data Distribution System (BDDS) of the ORPG. Alternatively, this type of testing would be accomplished in the implementation phase.

4.2 *Independent Validation & Verification Phase*

Before proceeding to the IV&V phase, the sponsoring agency must receive PMC approval to proceed using tri-agency ROC resources for the IV&V activity. The sponsoring agency would be responsible for presenting a briefing on the purpose and maturity of the algorithm. The PMC, along with advice from others, would make a decision on proceeding with IV&V at the ROC. However, if the sponsoring agency wished to independently fund a non-ROC group to perform the IV&V, it may do so. But there still should be a courtesy briefing to the PMC on the agency's intent and algorithm capability.

The purpose of this IV&V function is to review the scientific results of the researcher's testing and to test the algorithm using different data sets than were used during development testing. If the algorithm was developed on an ORPG clone using baseline software and APIs, this could be a simple task. However, if other development platforms and APIs were used, there may be considerable effort to change the code to use baseline APIs so that the algorithm can be tested on an ORPG clone. The sponsoring agency would be responsible for this transformation.

The IV&V team would produce a report of their scientific test results. Recommendations would be made to the sponsoring agency in the report on a proceed decision.

4.3 *Technical Advisory Committee (TAC) Review Phase*

After the IV&V report is issued, the results should be presented to the TAC following their standard methodology for presenting new science. The TAC would make a proceed recommendation based on the scientific results of the prior algorithm testing and presentation. Once a positive recommendation was received from the TAC, the sponsoring agency would then arrange for algorithm implementation and creation of baseline documentation for transfer to the ROC.

4.4 *Implementation Phase*

The implementation group would receive the testing results, code, and documentation from the researcher and IV&V activity to begin the task of producing operational and maintainable code. This task would include developing all changes required by the ROC to update formal baseline documents, recoding or modifying the developer's source code to ensure it follows ROC coding and design standards, documentation standards, and testing standards. Once implemented on an ORPG clone, the integration would be confirmed by running a validation data set to verify the output is the same as the IV&V team's output. The implementation group would not assess the scientific value of the algorithm. During implementation sub-phases several reviews would be held with the developer, IV&V team, and the ROC. These would include a requirements review, detailed design review, testing review, and a transfer readiness review.

After all parties are satisfied to the scientific correctness of the science and implementation efforts, all necessary documentation and results would be transferred to the ROC for their subsequent reviews, integration, testing, and final deployment to the field.

5. SUMMARY

The WSR-88D Common Operations and Development Environment (CODE) includes the tools, documentation, and guidance to provide an environment for the development and integration of WSR-88D algorithms using services of the Open Systems Radar Product Generator. Once the ORPG Algorithm API services and guidance for algorithm developers are complete, CODE will be ready to provide support for the NPI algorithm process.

6. REFERENCES

Ganger, T. J., R. E. Saffle, A. D. Stern, B. T. Ulery, and A. White: Development and Integration of New Hydrometeorological Algorithms into the WSR-88D using the Common Operations and Development Environment (CODE), *16th IIPS*, January 2000, Long Beach, CA, Paper 11.4

Istok, M. J., A. D. Stern, T. Ganger, D. Smalley, D.J. Seo, and C. McAdie: Experiences using WSR-88D CODE as a Developmental Tool for Radar Algorithm Development , *18th IIPS*, January 2002, Orlando, FL, Paper 5.9

Stern, A., M. J. Istok, T. Ganger, and S. Dobbs: Developmental Utilities for the WSR-88D Common Operations and Development Environment, *18th IIPS,* January 2002, Orlando, FL, Paper 5.8