Implementation and Evaluation of Laboratory/Tutorial Exercises for Software Defined Radio Education

Carl B. Dietrich[1], Frank E. Kragh[2], S.M. Hasan[3], Carlos Aguayo Gonzalez[4], Adeyemi A. Adenariwo[5], Haris I. Volos[6], Cecile C. Dietrich[7],Donna L. Miller[8], Jason Snyder[9], Stephen H. Edwards[10], Jeffrey H. Reed[11]

Abstract – Software defined radio (SDR) integrates several areas of electrical engineering, computer engineering, and computer science. Graduate courses at the Naval Postgraduate School and Virginia Tech survey SDR concepts and enabling technologies and provide hands-on experience via a project-oriented approach. The institutions have developed exercises that use the OSSIE open source SDR software, based on the Software Communications Architecture (SCA), an open U.S. Department of Defense standard and prevalent industry approach to SDR engineering. The exercises are suitable for use in university and short courses, and for individual study. Introductory exercises reinforce SDR and SCA concepts while familiarizing students with SDR infrastructure and rapid prototyping tools and preparing students for design and implementation projects. Later exercises include development of SDR broadcast receivers and digital transceivers, remote monitoring of SDR applications, and distributed SDR applications. The exercises are assessed based on retrospective pre- and post-tests, student evaluations, and analysis of download data.

Keywords: Software Defined Radio, Software Communications Architecture, Laboratory, Communications, Interdisciplinary

---

[1] Wireless@Virginia Tech, Blacksburg, VA, cdietric@vt.edu

[2] Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, fekragh@nps.edu

[3] Wireless@Virginia Tech, Blacksburg, VA, hasan@vt.edu

[4] Wireless@Virginia Tech, Blacksburg, VA, caguayog@vt.edu

[5] West Virginia University, Morgantown, WV,

[6] Wireless@Virginia Tech, Blacksburg, VA, hvolos@vt.edu

[7] Center for Assessment, Evaluation, and Educational Programming, Virginia Tech, Blacksburg, VA, cdietric2@vt.edu

[8] Department of Electrical and Computer Engineering, Naval Postgraduate School, Monterey, CA, dlmiller@nps.edu

[9] Department of Computer Science, Virginia Tech, Blacksburg, VA,

[10] Department of Computer Science, Virginia Tech, Blacksburg, VA

[11] Wireless@Virginia Tech, Blacksburg, VA, reedjh@vt.edu

INTRODUCTION

A Software defined radio (SDR) is a radio where most of the functionality including modulation, demodulation, and forward error correction coding is implemented in reprogrammable hardware components, such as general purpose processors and digital signal processors. SDRs are very attractive when it is desired that the radio be multimode, upgradable by software update, or otherwise flexible. SDRs are evolving technology with substantial investment from the American military and the mobile phone industry. This has generated a substantial demand for communications engineers with expertise in software defined radio design.

Several American engineering universities have introduced SDR into their engineering curricula [Arslan, 1; Bilen, 2; Goodman, 6; Hoffbeck, 7; Kang, 8; Katz, 9; Schelle, 14]. This is partially to respond to industry demand and partially because SDR design is an excellent example of modern multidisciplinary design challenges that is well suited to academic study. Virginia Tech (VT) and the Naval Postgraduate School (NPS) each include a course in SDR design in their graduate electrical engineering curriculum. These courses include lecture and laboratory portions. The laboratory portion is particularly novel and includes a sequence of laboratory exercises and projects that build upon one another to develop expertise in SDR design. These laboratories exercises have been used by both VT and the NPS for the courses as well as for several short courses at conferences. These laboratory exercises are the subject of this paper.

The complexity and interdisciplinary nature of software defined radios (SDRs) demand a diverse set of skills to master them. This is more evident when implementing advanced software architectures, such as the software communications architecture (SCA). Unfortunately, several areas that come together in SDR are traditionally isolated and subject to engineering specialties. Radio engineers deal with communication theory and signal processing, RF engineers with front-ends and antenna design, computer engineers with embedded system development and efficient implementations, and computer scientists with software reuse techniques, reconfigurable software architectures, and software design patterns. Hence, there is a need to expose students to all different areas of expertise required in SDR with a hands-on approach including experimentation and exercises.

Because the SCA places a high priority on software reuse and abstraction, it follows complex software engineering principles and design patterns. These techniques and principles are not part of the typical communication engineering curriculum, so these students can find them counterintuitive and even counterproductive. With practical experimentation and reference implementations, these principles and concepts become more apparent and their benefits evident.

The SCA also requires a significant amount of support files and knowledge of component-based development. Without the proper tools, SCA development can be slow and the learning curves rather steep. Fortunately, there are tools that can greatly facilitate SCA SDR development, but it is necessary to familiarize SDR engineers with them to boost their productivity.

Extra motivation and inspiration come with the successful application of complex theoretical principles. When students are able to capture signals over the air and see their knowledge materialized in a tangible implementation, they get an increased level of satisfaction and accomplishment. Struggling with problems in unfamiliar areas forces students to look at the issues from a different perspective and leads to a better understanding of principles and theories and better trained SDR engineers.

LABORATORY EXERCISES

The laboratory exercises introduce students to many aspects of SDR design, including the SCA and SCA-based design software environment and tools. The software used for the laboratories is the Open Source SCA Implementation - Embedded (OSSIE) package, which includes an SDR core framework based on the SCA, a set of tools for rapid development of SDR components and waveforms applications, and an evolving library of software components and waveform applications.

The OSSIE labs and tutorials provide a progressive introduction to the different aspects of SCA-compliant SDR development. The labs go from basic familiarization with the architecture and tools, to capturing signals off the air and remote debugging and monitoring. These labs highlight the benefits of the component model of the SCA in term of software reuse. The labs also address potential trouble areas and provide advice on potential debugging techniques.

Labs 1-4

Labs 1-4 introduce development and execution of SCA-based waveforms and components using the OSSIE tools, component library, and infrastructure software. These labs involve simulations run on a PC, but the basic approach is the same as that used in developing applications for use with an RF front end. In Lab 1, students learn how to build a simple baseband simulation of a digital communications link that uses quadrature phase shift keying (QPSK). In Lab 2, students build a waveform that also includes an amplifier component, and also learn to modify properties (configurable parameters) of the components and use the ALF waveform application visualization and debugging environment. In Labs 1 and 2, the students build the waveform using software components from the OSSIE library. In Labs 3 and 4, the students design their own simple software components. In Lab 3, students design a simple amplifier component and in Lab 4 they design a signal source component. Completion of Labs 1-4 provides the students with the skill set required to design their own waveforms for a wide range of software defined transmitters and receivers.

Lab 5: SDR Application: Building an AM Receiver

The main goal of Lab 5 is to show students a final SDR implementation by building and running a waveform that comprises the use of software components and hardware devices. In this lab, students develop an AM receiver that runs on a PC and uses an Ettus Research Universal Software Radio Peripheral (USRP) [Ettus, 4], a relatively low cost RF front end that connects to a computer using a USB 2.0 interface. The lab makes use of sample rate conversion concepts introduced in previous lectures and devices, which are software abstractions of physical hardware elements. The waveform uses a USRP controller component, devices that interact with the USRP and the sound card, and three signal processing components that perform decimation, automatic gain control, and demodulation. If an outdoor antenna or strong local AM broadcast station is not available, the AM demodulator can be replaced with an FM demodulator to receive VHF signals using bandpass sampling and the Basic Rx daughter board or UHF signals with a 400MHz daughter board. In some cases, a computer's sound card may not function correctly with a given Linux distribution or with the OSSIE sound card device. In such cases, the signal spectrum can still be observed.

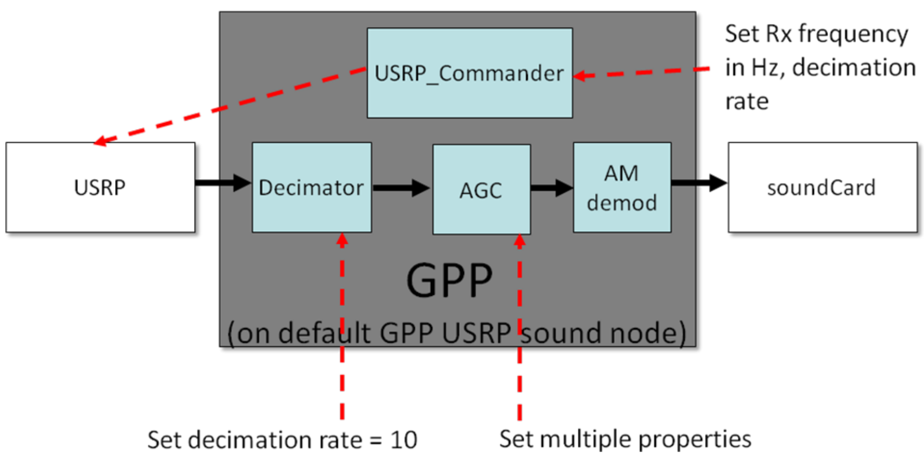

Figure 1. AM broadcast receiver built in Lab 5. Optionally, the AM demodulator can be replaced with an FM demodulator and used with the appropriate USRP daughter board and frequency setting to receive VHF or UHF narrowband FM signals.

Lab 6: Remote Waveform Application Visualization and Debugging

Lab 6 gives students an opportunity to visualize, probe, and debug an OSSIE SDR waveform application running on one computer (representing an SDR) using the ALF waveform visualization and debugging tool running on a second computer. The Common Object Request Broker Agent (CORBA) naming service, Domain Manager, Device Manager, and waveform application run on the first computer. The second computer connects to the first via Ethernet or WLAN and is configured to interact with the CORBA naming service on the first computer. When prototyping a waveform on a desktop or laptop PC, the ALF tool can be run on the same computer as the waveform. However, the remote visualization capability allows monitoring waveform applications using a separate test computer such as would be needed in cases in which the radio has no or a very limited graphical user interface. A portable computer can therefore be used to monitor radios that are embedded in large vehicles such as ships or aircraft, provided a network connection is available.

Lab 7: Distributed Waveform Applications

In Lab 7, students build and run a waveform that is distributed over two OSSIE nodes running on the same computer and then on two different computers. Each node specifies a set of devices such as general purpose processors to be managed by a device manager. Lab 7 builds on labs 1, 2 and 6. In this exercise, students build nodes using the OSSIE Waveform Developer (OWD) and then develop and run a waveform that runs on these nodes using the OSSIE Eclipse Feature (OEF). The ALF waveform visualization and debugging environment is loaded from either computer to visualize the waveform components and their connections, the constellation diagram, and the signal spectrum. Students observe the processes on each computer to see where each component is deployed and running. Lab 7 can serve as a starting point for implementation and investigation of more advanced distributed waveforms. For instance, a waveform could be run on many nodes. This is important because advanced SDRs usually employ multiple processors, often including general purpose processors (GPPs), digital signal processors (DSPs), and field programmable gate arrays (FPGAs).

If multiple students run waveforms on multiple nodes, it affects the network efficiency. This affects the processing load of each node. Multiple users can use the system. We could also develop a waveform that uses multiple processors in the computer or in an external device.

Lab 8: Distributed AM Receiver

In Lab 8, students build and run a broadcast receiver waveform that is distributed over two OSSIE nodes running on different computers. In this lab, we receive an AM signal using a USRP connected to the server, control the USRP using the USRP commander on the server and then decimate and demodulate on the client to listen to the signal. Lab 8 builds on labs 1, 2, and 4-7. Lab 8 is the distributed waveform extension of Lab 5 without AGC. Students learn the advantages of deploying a decimator on the GPP (the Server) that is directly connected to the RF front end, as opposed to the second GPP (client). Lab 8 prepares students for additional experimentation. They could measure variations in throughput or latency depending on where the decimators are deployed or when multiple users or waveforms are using the system simultaneously. Performance of the system could be improved based on where the components of the waveforms are deployed. After completing Labs 7 and 8, students are prepared to develop a waveform that uses more than two processors. The number of processors required in a system is determined by factors such as the data rate, computational complexity of algorithms used, and number of simultaneous waveforms.

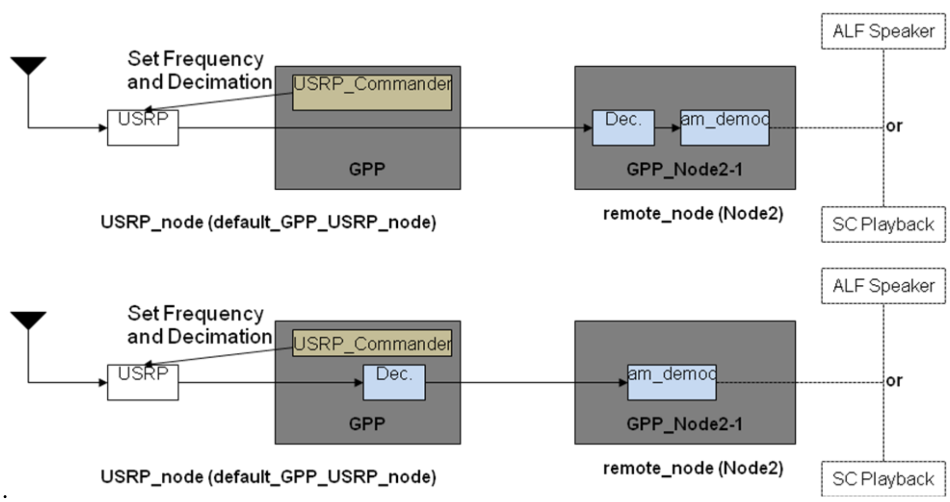

Figure 2. Distributed AM receiver developed in Lab 8

Digital Voice Lab

In this lab, the students have the opportunity to experiment with a waveform that implements all the basic functions of a digital communication system: modulation/demodulation, channel coding/decoding, interpolation/decimation, and symbol synchronization. The application of this waveform is the transmission and reception of digital voice data, however, the core of the waveform can be adapted to serve other applications. The goals of the lab are to familiarize the students with the inner workings of digital communication systems, and give them a basic structure that they can potentially modify to build their own digital waveforms.

The waveform has an application interface component called "OSSIETalk" which features a "push to talk" button; when the button is pressed, it captures audio through the microphone, applies continuously variable slope delta (CVSD) encoding and forwards the samples to the next component. The component also accepts CVSD encoded samples and plays them through the sound card.

The students are asked to build the waveform in two parts: first, the waveform is developed up to the symbol level (modulator) and tested through the Channel component, which is used to simulate an AWGN channel. Furthermore, the students test the waveform using different SNR levels and they adjust the convolutional forward error correction (FEC) code in order to improve the sound quality at low SNR levels.

On the second part, the waveform is developed in its full form: on the transmitter side the symbols are assembled into a frame, and pulse shaped (interpolated). On the receiver side, the received signal is decimated and passed through an Automatic Gain Control (AGC) component, and symbols are synchronized and extracted from the frame at the same time. The students have the opportunity to test the waveform either as a simulation using the Channel component or over the air using the USRP.

Multi-Antenna Methods Lab

In this lab the students are familiarized with three core multi-antenna methods: Maximal Ratio Combining (MRC), Space Time Block Codes (STBC), and Spatial Multiplexing. In the waveform, a component generates a known stream of symbols, which distributes the symbols across the transmit antenna(s) based on the multi-antenna method, then the transmit signals go through a fading channel (simulated using the Kronecker model) and are assigned to the received antenna (s). The received signals are then processed as appropriate based on the transmit method. Finally, a receive component demodulates the received symbols and computes the number of errors.

The students are asked to change the noise level and the number of antennas in order to observe their effects on the number of errors in the received data. In addition, the students have the opportunity to try a more advanced version of the waveform that includes a convolutional FEC code and observe the benefits of FEC coding over a fading channel. Finally, the students are encouraged to use ALF, OSSIE's waveform application visualization and debugging environment, to plot the signal at the different stages of the waveform.

ROLE OF RAPID DEVELOPMENT TOOLS

Originally, students completed these labs using the OSSIE Waveform Developer (OWD), a rapid development tool for creating prototype components and waveforms. OWD generated all of the necessary XML descriptors for the components and waveforms to ensure that they are compatible with the OSSIE core framework. In addition, it also generated skeleton source code implementations for components. This only addresses a portion of the development process however. The students still have to fill in the implementation, build the prototypes using the automake tools, and install them to the proper location in the file system before they can be used.

The OSSIE Eclipse Feature (OEF) is a set of plug-ins created to encompass the entire development process. OEF duplicates the core functionality of OWD (xml descriptor and skeleton implementation generation), while adding many features. First, it allows students to fill in their implementation with all the benefits of a fully-featured integrated development environment (IDE) including syntax highlighting, code completion, and integrated debugging. It provides XML editing and checking support when modifying the descriptor files. Other freely available Eclipse plug-ins can provide support for many other features, including unit testing and integration with popular version control software including CVS and Subversion. Finally, OEF completely automates the process of building and installing components and waveforms. Every time a student saves his or her work, the component or waveform is automatically rebuilt and installed so it is immediately available for testing.

EVALUATION OF LABORATORY EXERCISES

Study design: Virginia Tech Group

Retrospective pretest designs have proven useful in evaluating program outcomes [Pratt, 13] and in assessing the outcomes of short term educational interventions [Moore, 11]. In the Virginia Tech graduate student sample, the retrospective pretest design was used to assess how each lab affected student understanding. Students were asked to rate their understanding of course topics on a Likert-type scale of 1-5, with '1' signifying the lowest rank and '5' the highest. These ratings pertained to understanding both before and after each lab (see Appendix). [Lam, 10] classified this question format as a post test and retrospective pretest method. Of the various forms of retrospective self reports of educational change, this form was less likely than other forms to result in an overestimate of educational change after a specific educational intervention [Lam, 10], in this case, the software defined radio labs.

Sample

Eleven of fourteen Virginia Tech graduate students enrolled in ECE 5674 participated in this study. These students are in the Electrical and Computer Engineering program. Of the students enrolled in the course, 3 were in computer engineering and 11 were in electrical engineering. Approximately 21% of the students enrolled are female. About half of the students are United States citizens.

Analysis

The Wilcoxon matched-pairs signed ranks test was used to analyze students' self reported understandings before and after the labs. The Wilconxon test was chosen because it does not make assumptions about the population from which the sample is drawn. Furthermore, it is an appropriate test for ranked or ordinal data as opposed to quantitative data. [Bradley, 3; Gibbons, 5]

Results

The median rank for each lab as well as the range of scores for each lab is given in Table 1. Median self-reported understanding was higher after the labs than before in all cases, and in two-thirds of the cases, the increase in understanding was found to be statistically significant. Statistical significance was defined at a criterion alpha level of .05. In Table 2, *p*-values with an asterisk were found to be statistically significant. The *p*-value refers to the probability of having differences as or more extreme as the differences observed if the null hypothesis-no change in students' understandings-is true.

Table 1 Students' self-reported understanding of selected topics before and after labs (N=number of responses)

Lab/Topic	Before			After		
	N	median	range	N	Median	range
Lab 1 SCA Components	11	2	4	11	4	3
Lab 2 SCA Components	11	3	3	11	4	2
Lab 2 SCA Properties	11	2	3	11	4	2
Lab 3: SCA Components	11	3	2	10	4	3
Lab 3: SCA Properties	10	3.5	3	10	4	2
Lab 4: SCA Components	10	3.5	3	10	4	2
Lab 4: SCA Properties	10	4	4	10	4	2
Lab 5: SCA Devices	10	1.5	2	10	4	2
Lab 5: SCA Device manager	10	2.5	3	10	4	4
Lab 5: SCA Decimation	10	2.5	3	10	4	2
Digital waveform blocks	10	3	4	10	4	2
Digital waveform lab: Decimation	10	4	3	10	4	2
Digital waveform lab: Interpolation	10	3	4	10	4	2
Lab 6: Corba in SCA	10	2	3	10	3	3
Lab 7: SCA Device Manager	10	2.5	3	10	3.5	4
Lab 7: SCA Executable devices	10	2.5	3	10	3.5	3
Lab 8: SCA Device Manager	10	3	3	9	4	2
Lab 8: SCA Executable devices	10	3.5	2	9	4	3
Lab 8: SCA Devices	10	3	2	9	4	2
MIMO Lab: MIMO Techniques	10	2	3	10	4	3
MIMO Lab: MIMO requirements	10	1.5	4	10	4	2

Table 2 *p*-values for before-after median self-reported understanding pairs (* = statistically significant result)

Pairs	p value (* = values significant at α=.05)
Lab 1 SCA Components before and after	.016*
Lab 2 SCA Components before and after	.014*
Lab 2 SCA Properties before and after	.011*
Lab 3: SCA Components before and after	.041*
Lab 3: SCA Properties before and after	.026*
Lab 4: SCA Components before and after	.180
Lab 4: SCA Properties before and after	.066
Lab 5: SCA Devices before and after	.004*
Lab 5: SCA Device manager before and after	.080
Lab 5: SCA Decimation before and after	.016*
Digital waveform blocks before and after	.039*
Digital waveform lab: Decimation before and after	.059
Digital waveform lab: Interpolation before and after	.026*
Lab 6: Corba in SCA before and after	.014*
Lab 7: SCA Device Manager before and after	.066
Lab 7: SCA Executable devices before and after	.039*
Lab 8: SCA Device Manager before and after	.066
Lab 8: SCA Executable devices before and after	.408
Lab 8: SCA Devices before and after	.038*
MIMO Lab: MIMO Techniques before and after	.011*
MIMO Lab: MIMO requirements before and after	.017*

Student Ratings of NPS Labs

Students completing the SDR course at the Naval Postgraduate School are asked to rate the laboratory exercises as part of the course evaluation. The overall rating for the labs for the Summer 2009 course, the most recent offering, was 4.33 on a scale of 1 to 5, with 5 being the best possible rating.

Downloads of Laboratory/Tutorial Exercises

Laboratory exercises 1-3 and 5-8 are posted on the OSSIE web site [OSSIE, 12] and have been downloaded at a rate of approximately 1000 per year each as shown in Table 3. The Lab 4 listed in Table 3 is an older version that provides instructions for setting up the USRP device interface. The current Lab 4, in which students create a signal source component, was not yet posted as of December 2009. In addition to use at academic institutions, the labs are known to have been used by employees of companies that include Agilent, Motorola, Rockwell Collins, and Thales.

Table 3. OSSIE Laboratory Exercise Download Statistics, May-November, 2009

Lab	May	June	July	Aug	Sept	Oct	Nov	Total	Projected Annual
1	131	167	134	161	151	167	109	1020	*1749*
2	91	99	99	109	109	112	67	686	*1176*
3	96	104	111	114	119	126	59	729	*1250*
4	62	67	57	74	71	71	13	415	*711*
5	128	166	167	162	163	176	81	1043	1788
6	61	61	79	70	67	73	40	451	*773*
7	64	53	63	44	61	59	47	391	*670*
8	56	48	62	48	56	64	39	373	*639*

FUTURE WORK

Adjustments to the labs are planned based on feedback from the students. In addition, as the OSSIE Waveform Workshop rapid development tool suite matures, some of the lab exercises can be completed more quickly and may be combined in future course offerings. Also, as the tools are developed further, more aspects of the SCA are hidden from the students, so additional steps may be added to the labs in which students delve into details of the SCA. Examples include browsing XML files and identifying key information related to the framework, signal processing components, and applications. Also, it may be helpful to ask students to page through console output to identify specific events processed by the Domain Manager, possibly running the node booter at different debug levels to reveal details of its operation.

Some labs that do not require RF hardware may be assigned for out-of-class completion to make room for additional exercises and lectures. Candidate topics for enhanced lectures and new exercises include software engineering and collaborative software development. For example, the OSSIE Eclipse Feature makes use of Subclipse, an Eclipse plug-in that provides an interface to the subversion revision control system. In future offerings of the course, we plan to set up a subversion repository that allows students to submit waveform applications, signal processing components, and node configurations developed in the labs for testing by an instructor or teaching assistant.

Lab reports to date have been brief and focused on feedback to improve the labs, including suggested questions to be asked of students who complete the labs. More formal lab reports may be introduced in future course offerings. Templates for the lab reports may be developed using a word processor and provided for students to fill in as they complete the labs, to allow creation of more structured and formatted reports while allowing students to concentrate on technical aspects of the labs. This approach is used in Virginia Tech's senior-level analog and digital communications laboratory course.

CONCLUSIONS

Due to the multidisciplinary nature of SDR design, it is challenging to teach and it is challenging to learn. As is often the case, a hands-on experience is very useful for learning concepts and strengthening understanding. This paper has described a well-developed sequence of increasingly sophisticated design experiments that provides this experience for students at Virginia Tech, the Naval Postgraduate School, and short courses at several technical conferences. Discussion of student opinions of the laboratory experience is included. The laboratory exercises and all necessary software is available for free download at [OSSIE, 12]. The authors encourage faculty members and others to consider using these exercises to help satisfy their SDR design education needs.

REFERENCES

[1] Arslan, H., "Teaching SDR through a laboratory based course with modern measurement and test instruments," Proceedings of the SDR Forum Technical Conference, November 2007.
[2] Bilen, S., "Implementing a Hands-on Course in Software-defined Radio," 2006 ASEE Annual Conference Proceedings, June 2006.
[3] Bradley, J., *Distribution free statistic,* Prentice Hall, Englewood Cliffs, NJ, 1968.
[4] Ettus Research web site, http://www.ettus.com.
[5] Gibbons, J., *Nonparametric statistics: An introduction.* Sage, Newbury Park, CA, 1993.
[6] Goodmann, P., "A Software-Defined Radio Project for First-Year ECET Students", 2007 ASEE Annual Conference Proceedings, June 2007.
[7] Hoffbeck, J.; Melton, A., "RF Signal Database for a Communication Systems Course", 2006 ASEE Annual Conference Proceedings, June 2006.
[8] Kang, J.; Olson, B.; Felzer, A.; Chandra, R.; Oldak, S., "Simulink Based Real-Time Laboratory Course Development," Proceedings of the IEEE International Conference on Microelectronic Systems Education, pp.15-16, 3-4 June 2007.
[9] Katz, S., "Using Software Defined Radio (SDR) to Demonstrate Concepts in Communications and Signal Processing Courses", 2009 ASEE Annual Conference Proceedings, June 2009.
[10] Lam, T. and P. Bengo, "A Comparison of Three Retrospective Self-Reporting Methods of Measuring Change in Instructional Practice," *American Journal of Evaluation,* v. 24, no. 1, 2003, pp. 65–80.
[11] Moore, D., and C. Tannis, "Measuring Change in a Short-Term Educational Program using a Retrospective Pretest Design. *American Journal of Evaluation*, v. 30, no. 2, 2009, pp. 189-202.
[12] OSSIE project web site, http://ossie.wireless.vt.edu.
[13] Pratt, C., W. McGuigan, and A. Katzev, "Measuring Program Outcomes: Using Retrospective Pretest Methodology. *American Journal of Evaluation,* v. 21, no. 3, 2000, pp. 341–349.
[14] Schelle G.; Fay D.; Grunwald D.; Connors, D.; Bennett, J., An Evolving Curriculum to Match the Evolution of Reconfigurable Computing Platforms. In Proceedings of WRCE2006 (IEEE Computer Society Workshop on Reconfigurable Computing Education), March, 2006.

Carl B. Dietrich

Carl Dietrich is a research assistant professor at Virginia Tech, where he completed Ph.D. and M.S. degrees after graduating from Texas A&M University. He worked with the Defense Information Systems Agency, Arlington, Virginia and Bell Northern Research, Richardson, Texas and conducted research on adaptive and diversity antenna systems and radio wave propagation. His current work in software defined radio (SDR) includes leading projects related to the OSSIE open source effort. He chairs the Wireless Innovation Forum Educational Work Group, is a member of ASEE, IEEE, and Eta Kappa Nu, and is a Professional Engineer in Virginia.

Frank E. Kragh

Frank Kragh is an Assistant Professor of Electrical and Computer Engineering at the Naval Postgraduate School in Monterey, California. Dr. Kragh received his B.S. from Caltech in 1986, his M.S. from the University of Central Florida in 1990, and his Ph.D. from the Naval Postgraduate School in 1997. His chief research and teaching interests are digital communications, software defined radio, multiple-input multiple-out systems, and military communications systems.

S.M. Shajedul Hasan

Dr. S.M. Shajedul Hasan is a Research Scientist for Wireless @ Virginia Tech. He received his B.S. degree from Bangladesh University of Engineering and Technology, Dhaka, Bangladesh, the M.S. degree from the University of Tennessee, Knoxville, TN, and Ph.D. degree from Virginia Tech, Blacksburg, all in electrical engineering. His research interest is in the RF front end, software radio, and CR area. He is also an expert in the field of instrumentation and measurement, and applied digital signal processing. Dr. Hasan was the recipient of the 2007 William Bazzy Fellowship from the Microwave Journal. He also participated in the 2007 smart radio challenge competition organized by the SDR forum and received the best design award. He was the recipient of the first prize of the IEEE Myron Zucker Student Design Award in 2001.

Carlos R. Aguayo Gonzalez

Carlos Aguayo Gonzalez is a Ph.D. candidate in electrical engineering at Virginia Tech. He received a B.S. from Instituto Tecnológico y de Estudios Superiores de Monterrey (ITESM) Campus Guadalajara, Mexico, in 2000 and an M.S. from Virginia Tech in 2006. He worked from 2000 to 2002 as a software engineer for Mixbaal, Guadalajara, Mexico, developing spread spectrum radios for last-mile applications. His current research interest is efficient implementations of SDR, including their validation, certification, and testing. He is a member of IEEE.

Adeyemi A. Adenariwo

Adeyemi Adenariwo received his B.Sc. from Louisiana State University, Baton Rouge, Louisiana in Dec 2005 and M.Eng. degree from Virginia Tech, Blacksburg, VA in May 2009. He worked at Tyco Electronics, Lynchburg VA in fall of 2007. He is currently studying to attain his Ph.D. at West Virginia University. He developed and updated OSSIE Labs at Virginia Tech. His interests include wireless communications, electronics, power and control systems.

Haris I. Volos

Haris I. Volos (hvolos@vt.edu) is a doctoral candidate in Virginia Tech's Bradley Department of Electrical and Computer Engineering. He received an M.S. in electrical engineering from Virginia Tech as a scholar in the Cyprus America Scholarship Program (CASP), and a B.S. (summa cum laude) in electrical engineering from Old Dominion University. His research interests are artificial intelligence methods for multi-antenna systems, SDRs, and propagation measurements and analysis for ultra-wideband signals. He is a member of the Wireless@VT research group.

Cecile C. Dietrich

Cecile Dietrich has applied quantitative and qualitative research methods and data analysis techniques to evaluation and basic research since 1998, and has worked with Virginia Tech's Center for Assessment, Evaluation, and Educational Programming (CAEEP) since 2006. Her background and interests include human development, health/life sciences, mathematics, and science, technology, engineering, and mathematics (STEM) education. She has conducted research and performed evaluation work for the World Bank, state and local educational and extension agencies, independent and university-based engineering educators, and faith-based organizations.

Donna L. Miller

Donna Miller is a Research Associate and manager for the software defined radio laboratory in the Electrical and Computer Engineering Department at the Naval Postgraduate School in Monterey, California. Ms. Miller received her B.S. in Electrical Engineering from Worcester Polytechnic Institute in 1985 and her M.S. in Software Engineering from the Naval Postgraduate School in 2000. She joined the Naval Postgraduate School faculty in 2004.

Jason Snyder

Jason Snyder is a third year PhD student in the Department of Computer Science at Virginia Tech. His research interests include Software Engineering and Computer Science Education with a focus on finding ways to fight attrition in introductory Computer Science courses.

Stephen H. Edwards

Stephen Edwards, Associate Professor of Computer Science at Virginia Tech, has interests in component-based software, automated software testing, and educational uses of computers. As the PI on an NSF phase II CCLI project, he developed Web-CAT, the most widely used open-source automated grading tool for computer programming assignments, with nearly 10,000 users at over 30 institutions worldwide. He is also a member of his department's undergraduate program committee, and chair of the subcommittee on curriculum and courses. Dr. Edwards has a background in component-based systems and has collaborated on software-defined radio research since 2007.

Jeffrey H. Reed

Jeffrey H. Reed is the Willis G. Worcester Professor in the Bradley Department of Electrical and Computer Engineering and director of Wireless @ Virginia Tech. His area of expertise is in software radios, cognitive radios, wireless networks, and communications signal processing. He is an IEEE Fellow and the author of *Software Radio: A Modern Approach to Radio Design* (Prentice Hall, 2002) and *An Introduction to Ultra Wideband Communication Systems* (Prentice Hall, 2005).