Education in Software Defined Radio Design Engineering

Abstract— **Software Defined Radio (SDR), an interdisciplinary emerging technology, presents new challenges for communications engineers and engineering educators. In SDR, signal modulation and information coding are defined in the system's software, not hardware. The authors have incorporated SDR design into their respective curricula both to support the growing demand for SDR engineering and to teach widely applicable systems engineering concepts. SDR-oriented curricular changes include new courses, laboratories, and software design tools. Software radio design is taught as an interdisciplinary systems engineering undertaking, emphasizing the importance of systems engineering methodologies, design architecture, and hardware issues. The Software Communications Architecture (SCA), a military SDR design standard, is used as an illustrative example of smart systems engineering through establishment of a well-defined architecture. Software topics include software architectures, object oriented programming, the SCA and other relevant software standards, multi-rate signal processing, and software engineering. Hardware topics include the radio frequency front end, analog-to-digital and digital-to-analog converters, microprocessors, digital signal processors, and field programmable gate arrays. Hands-on SDR laboratories undergird project-based learning. Laboratories include development of SCA-based modular signal processing components and SDR AM receivers. This equips students for design and implementation of more complex components or applications, preparing them for the challenges of SDR design in industry or government.**

I. Introduction

Software defined radio design is a key topic in the graduate communications engineering curricula at the authors' universities. At each institution, this includes a lecture and laboratory course in software defined radio design, laboratory facilities for SDR design, and related research efforts. Very few universities include software defined radio in their curricula. In this paper, the authors discuss their SDR design curricula and the need for it, in the hopes of promoting and facilitating similar developments at other universities.

A. Rationale – why is SDR education important?

SDR design is an excellent topic to be included in graduate communications engineering curricula. SDRs are important for the wireless communications industry, the military, and the public safety sector. The demand for engineers with the appropriate skill set for software defined radio design is much greater than the current supply. SDR has long been implemented in base station design and is expected to be utilized in cell phones by 2010 and dominate cell phone designs in 2015[1]. Also SDR design is a good example of the benefits of best practices in interdisciplinary engineering. SDR design is typical of the complex and multidisciplinary design projects that challenge practicing engineers today. In engineering education, it is important to remember that many of the engineering technologies our students will work with in their careers have not yet been invented. Therefore it is important to emphasize in engineering education design techniques and concepts that are likely to be applicable to the engineering challenges of the future. SDR design involves communications engineering, radio frequency (RF) engineering, hardware and software engineering, analog and digital electronics, and systems engineering, and therefore relates well to the complex, interdisciplinary design challenges that face engineers generally. Furthermore, SDR design and implementation is financially practical in a university laboratory setting. In summary, the study of SDR design prepares students for careers involving SDR design and careers involving typical complex, interdisciplinary design.

B. Background on SDR

Engineers argue about the definition of a software defined radio, mostly regarding the degree of software-provided reconfigurability required. We will take, as a reasonable working definition, the one from Reed[2]. He defines a software defined radio as "a radio that is substantially defined in software and whose physical layer behavior can be significantly altered through changes to its software." A strong analogy can be made between SDRs and computers. A computer can be a word processor, a financial tool, or an analysis tool by running the appropriate software. Similarly, a SDR can be a cell phone, a wireless LAN transceiver, or a police radio by running the appropriate software. Examples of software defined radios include the US military's Joint Tactical Radio System[3], the US Navy's Digital Modular Radio[4], and Vanu's software defined Anywave® cellular basestations[5].

The primary desirable attribute of a SDR, in comparison to a traditional hardware radio, is that the physical layer behavior, to include modulation and/or demodulation, is defined in software, and therefore can be altered without hardware changes. This is accomplished by representing the signals as discrete time signals in the SDR's memory and performing the mathematical operations on the signals to modulate or demodulate numerically in general purpose digital electronics (e.g. microprocessors or similar integrated circuits). Therefore, a typical SDR includes a microprocessor or similar programmable integrated circuit, digital-to-analog and/or analog-to-digital converters, and a radio frequency front end (RFFE). The RFFE includes antennas, analog amplifiers, analog filters, and usually analog frequency up-converters and/or analog frequency down-converters.

SDRs are made possible by improvements in analog to digital and digital to analog converters and digital electronics that allow the sampling, reconstruction, and programmable processing of complex signals at speeds high enough to transmit and receive signals in real time.

SDRs are important due to their capabilities in multimode and integrated services, flexibility, and speed to market. The ability to support many forms of radio communications in one radio is important to the commercial wireless industry, the military, and the public safety sector. For example, multimode operation of a cellular basestation allows one basestation transceiver to support customers using various wireless handsets (2G, 3G, GSM, CDMA, etc.) without additional hardware, space, or electrical power. A military, police, or fire unit may not know in advance what other units they will need to communicate with, and therefore it is very useful for them to carry one radio that can communicate in all the modes of those with whom they are most likely to need to communicate. Multimode operations also imply that many SDRs can use the same hardware design. This could allow a radio producer to use a single hardware design to satisfy their wireless industry, military, and public safety customers' disparate needs. Flexibility is important due to the ability to reconfigure or upgrade the radios without changing hardware, including over the air updates. Upgrades in cellular standards can be very expensive and logistically awkward. SDR basestations allow the upgrade to occur over the air without replacing hardware, thus saving money and allowing for very rapid deployment of the upgrade. SDR technology can help field a product more quickly due to the time savings associated with software development as compared to hardware development. This gets products to market faster, accelerating the deployment of new capabilities. SDR is also the basis behind cognitive radios, radios that are aware of their environment and can learn from experience. Such radios are now being used for better spectrum management and rapid deployment of communications infrastructure.

II. Authors' SDR course

This section discusses the courses in SDR design offered at the authors' universities. Although SDR design has not become mainstream in American engineering curricula, there are several other universities which address SDR in their curricula. This paper will be limited to the courses offered at the authors' universities. There are significant differences between the SDR design courses at these two institutions to include length, emphasis, certain topics, and student background, but this paper will largely ignore the distinctions and describe them as if they were one course.

A. Goals and prerequisites

Most of the students in the course have some background in communications engineering. The goal of the course is to build on the students' knowledge of conventional radio systems to establish proficiency as SDR design engineers. This includes building an understanding of the technologies that underlie SDR including hardware and software technologies and architectures. Furthermore, the student must learn design challenges, design methods, and design tools for SDR. To develop the desired level of proficiency and confidence, the students must design, build, and test a SDR.

Due to our belief in the importance of design methodology and value of hands-on learning, we emphasize software engineering strategies, including software architecture, and use them in laboratory exercises. In particular, the Software Communications Architecture (SCA)[6], a popular open software architecture for SDRs that was established by the US military, is emphasized in the lectures and used extensively in the laboratory exercises and the class design projects. In particular, we utilize a software package called Open Source SCA Implementation::Embedded (OSSIE)[7] that provide an instantiation of the SCA core framework and related design tools for rapid development of SCA-compliant software. OSSIE has been developed at Virginia Tech and is available for free download via the Internet.

Prerequisites are a challenge for interdisciplinary engineering education, and SDR design is no exception. Requisite skills for SDR design include communications theory; digital signal processing (DSP); digital, analog, and RF hardware; software engineering; computer programming; and systems engineering. It would be impossible to teach all these aspects in one course and it would be impractical to require more than one or two of them as prerequisites. In engineering practice, interdisciplinary design is not attempted by finding an engineer with great expertise in all relevant areas, but rather by building a team of engineers who collectively have expertise in most of the relevant areas and are systematic about building the remaining proficiency they need. Along these lines, we have designed our course to require only the core of these requisite skills as prerequisite while we teach the essentials of the rest and encourage our students to collaborate to share their expertise in the remaining requisite areas. We have designed our course to be a graduate level course and that has made the pool of the remaining requisite expertise across the enrolled students to be quite adequate.

The textbook used, Software Radio: A Modern Approach to Radio Engineering by Reed[2], thoroughly covers all the topics in the course. There are several other textbooks one may want to consider[8, 9, 10, 11].

Although the course in our universities is a product of the graduate level electrical engineering curriculum, it could be useful to other curricula as well. Learning how to do SDR design includes learning many aspects of software engineering, object oriented programming, and systems engineering, so software engineering, computer science, and systems engineering and other related curricula might benefit from a course in SDR design. Furthermore, a broader background of the students in the class helps all the students learn, as we have discussed.

B. Topics covered

The course includes approximately 33 lecture hours which cover SDR software design, SDR hardware design, and digital signal processing for SDRs while emphasizing systems engineering and software engineering skills that are important in most complex interdisciplinary design efforts today.

SDR software design includes four lecture hours of software engineering and three lecture hours of the Software Communications Architecture. Key topics include software modularity, scalability, flexibility, and maintainability, common middleware, common Application Programming Interfaces (APIs), object oriented programming, and object request brokers. The software engineering terms modularity, scalability, flexibility, and maintainability are defined by the Institute of Electrical and Electronics Engineers (IEEE)[12], but they revolve around the systems engineering needed to make the complex system one that can be easily modified as requirements and technologies change over the life of the system.

SDR hardware design includes approximately four lecture hours of RFFE hardware design, seven lecture hours of analog-to-digital and digital-to-analog converters (ADCs and DACs), and five lecture hours of programmable digital electronics including microprocessors, digital signal processors (DSPs), and field programmable gate arrays (FPGAs). Most of these topics are taught using illustrative architectures for the various parts of the SDR radio, such as the relative advantages and disadvantages of various RFFE architectures for a SDR receiver. The role of systems engineering in relating component limitations to system capabilities is emphasized.

Digital signal processing for SDRs includes about three lecture hours on sampling, decimation, and interpolation of signals and six lecture hours on direct digital synthesis (DDS) of signals. Emphasis is placed on the flexibility offered by processing signals digitally and the importance of multirate strategies to mitigate computational complexity.

C. Laboratory Exercises

The laboratory exercises are key to the learning process in this class. The laboratory exercises are designed to lead the student through the process of various aspects of designing and operating a SDR while employing the system and software engineering strategies taught in the lectures and textbook. These laboratory exercises are freely available via the Internet[13].

The hardware and software required for the laboratories includes the OSSIE software, mentioned earlier, the very inexpensive Universal Software Radio Peripheral (USRP)[14], and a personal computer (PC). The OSSIE software includes an SCA-based core framework, tools for rapid development of SDR components and waveforms, and an evolving repository of pre-built components and waveforms. The core framework is software that provides common services that almost any SDR would need, in a way that conforms to the SCA. The USRP is a hardware board designed to operate as a computer peripheral that provides the RFFE and conversion between analog and digital signals. It is constructed as a motherboard with removable daughterboards. The daughterboards convert between the RF signals and intermediate frequency (IF) in-phase and quadrature signals. There are many daughterboards available, each supporting a popular RF band. The USRP is not needed during most phases of the design and our students did not use the laboratory simultaneously, so sharing equipment was easy. A PC of modest capabilities will suffice. The most demanding feature is that the PC should have at least one USB 2.0 port. We had no trouble supporting twenty enrolled students with a laboratory with four complete workstations (software, USRPs, and PCs) and four partial workstations (software and PCs). The total cost for equipment for this laboratory easily supporting twenty students was approximately $10,000.

The course includes five laboratory exercises, which are designed to lead the student through the process of using the OSSIE software to design a simple software defined radio. In the final laboratory, the students use the lessons learned in class and the prior four laboratories to design a SCA-based software defined radio capable of receiving commercial amplitude modulation (AM) broadcasts.

The first four labs teach skills necessary for the final lab in which the students build and operate a SCA-compliant SDR capable of receiving commercial amplitude modulation (AM) broadcasts. The first lab is an introduction to the OSSIE software package including the OSSIE waveform developer (OWD) tool. In this laboratory, the student uses provided components to assemble and run a simple waveform. Some of the terminology used in the laboratories is defined in the SCA specifications[6], including the terms ‘component’ and ‘waveform’. A component is a software object - typically one whose function corresponds to that of a typical component in a common hardware radio (e.g., a filter). A waveform is the set of transformations applied to information that is transmitted over the air and the corresponding set of transformations to convert received signals back to their information content. In the second laboratory, the students assemble provided components to produce and run a quadrature phase shift keying (QPSK) transmitter waveform. In this lab, the students learn how to modify existing software components using OSSIE component properties. This illustrates effective software reuse which is an important element in wise software engineering. In the third laboratory exercise, the students learn how to use the OSSIE component editor tool to build their own SCA-compliant components. The fourth lab teaches the students how to interface their SDR software with hardware input and output devices including the USRP and the computer's sound card. In the fifth and final laboratory exercise, the students use the lessons learned from the first four laboratories to design and build a SCA-based AM radio receiver using the OSSIE tools, the USRP with basic daughterboards, and the computer's sound card and speakers.

D. Projects

While the laboratory exercises described in section C are intended to teach students how to implement SDR designs and use the design tools provided, they are very tutorial and not intended to test the students' ability create their own original SDR designs or promote the convergent-divergent thinking and project team systems engineering approach required for design of complex systems. Unlike the laboratory exercises, there is essentially no written guidance, the students are encouraged to form teams and collaborate across teams, and sharing of hardware and software across teams is encouraged.

The purpose of the projects is to go beyond the mechanical design methods in the laboratory exercises and foster increased understanding of efficient team-based project design. This includes fostering student creativity and helping them practice design decision-making which relies on quantitative and non-quantitative analysis. Such decisions are critical in forming a team of the right mix of skills, scoping the project to fit the team's time and other resources, translation of the system requirements into an original design, and the many technical tradeoffs necessary in complex system design. Teaching these concepts in this way is often referred to as project based learning (PBL) and is described in Dym et al[15].

A list of potential projects is provided to the students, but they are encouraged to generate their own project ideas. In a recent course offering the provided list included the following topics: A frequency modulation (FM) SDR, a software defined oscilloscope and spectrum analyzer, a binary frequency shift keying (BFSK) SDR transceiver, a QPSK transceiver, an 802.11b band packet detector, etc.

Each team must submit a proposal before commencing work. This gives the instructor an opportunity to stop projects that are too easy or (more commonly) scope down projects that are too aggressive, and to make early recommendations about methods and skills needed.

E. Assessments

The assessments in the course include quizzes, laboratory reports, and projects. Quizzes are used for those areas that are easily taught as hard theory with clear right and wrong answers. Since much of SDR design is about options and multidimensional trade-offs with no clear right or wrong answer, written exams are often not the best assessment. Therefore, quizzes were used for approximately 20% of the grade.

The laboratory exercises were graded via objective evaluations of laboratory reports which totaled 40% of the grade. Lab reports required students to answer specific questions that were designed to be fairly easy for anyone who had completed the labs and very hard for those who had not. Lab reports also were used for feedback on the exercises themselves.

The remaining 40% of the grade was for the project. The authors believe that although it is highly desirable for the students to achieve a working product, it is also important to manage the scope of work and time commitment for the students so as to maintain the student's workload as appropriate for the credit assigned to the course. Therefore, we utilized two methods for objectively assessing the project. If the working product was demonstrated successfully to the instructor, only a very short written report was required, and a high grade was awarded. If the product was not working after the students had expended approximately twenty hours each working the project, they had the option of submitting a longer report detailing their attempts, lessons learned, and the key obstacles. The instructor followed up their report with oral questions regarding their attempt and then awarded a grade. If the instructor determined the project failed due to its complexity in spite of an intelligent approach and solid effort, a high grade was awarded. If the instructor determined the project failed due to a poor approach, lack of skill, or insufficient effort, a low to failing grade was awarded. In related classes, we have had the students perform design reviews on each other's projects. In those classes, their review is graded and can influence the instructor's grade for the project. This exposes them to the process of a design review, enables critical thinking, and helps students learn from each others' projects.

F. Potential improvements to course

This course has been offered with great success two times at each institution. Students have found it useful. Enrollment has remained high. Some of the graduating students have used their SDR design skills to secure positions in design and acquisition of SDR systems. SDR is a rapidly evolving field of engineering. We learn how it is evolving through our research work, the published literature, conferences, and collaboration. We continuously modify the course to reflect the current best practices of SDR design. In the future, we will utilize new rapid prototyping and debugging tools that have recently been incorporated into OSSIE, facilitating the development of even more sophisticated projects. Other potential future modifications or improvements to the course include augmenting the laboratory to include DSP and/or FPGA design, ultimately to include a SDR design project with a mixed microprocessor/DSP/FPGA based hardware solution to illustrate the effective combination of the relative strengths and weaknesses of these integrated circuits to produce a high-end SDR. Also to be considered is the addition of other design tools, such as GNU radio[16] or SCA Reference Implementation (SCARI) software[17].

III. Conclusions

SDR is an important part of the future of radio communications and therefore, of engineering.

SDR is currently underrepresented in engineering curricula in relation to its importance to the careers of communications engineers. Furthermore, SDR is an excellent candidate for inclusion in engineering curricula because it illustrates systems engineering concepts and methodologies that are important to most complex interdisciplinary design projects while only requiring modest laboratory investment. Due to its interdisciplinary design nature, teaching SDR is not easy. This has motivated the authors to document much of their curricula developments in SDR here in the hopes of facilitating the inclusion of SDR in engineering curricula that do not currently include it. Other potentially useful resources for SDR education, including other design tools, are listed on the web site maintained by the Education Working Group of the Software Defined Radio Forum[18].

References

1. U. Ramacher, "Software-Defined Radio Prospects for Multistandard Mobile Phones," IEEE Computer Magazine, October 2007, pp. 62-69.
2. J. H. Reed, *Software Radio: A Modern Approach to Radio Engineering*. Upper Saddle River, New Jersey: Prentice Hall, 2002.
3. S. I. Erwin, "Pentagon to Invest in PC-Style Radios," National Defense Magazine, August 2002.
4. P. Tran et al, "Digital Modular Radio: An Interoperable Capability Enabler in Net-Centric Environment", Proceedings of the Military Communications Conference (MILCOM), Orlando, Florida, October 2007.
5. J. Chapin, Software Radio for Cost-effective Growth Opportunities for Rural Carriers, The Basestation e-Newsletter, June 2007. Available: http://www.openbasestation.org/Newsletters/June2007/Vanu_SoftwareRadio.htm
6. Software Communications Architecture Specification, Joint Program Executive Office (JPEO) Joint Tactical Radio System (JTRS) Standard, 2006. Available: http://sca.jpeojtrs.mil/
7. Open Source SCA Implementation::Embedded (OSSIE) Software package, Virginia Tech. Available: http://ossie.wireless.vt.edu/
8. J. Mitola, III, *Software Radio Architecture: Object-Oriented Approaches to Wireless Systems Engineering*. San Francisco: Wiley, 2000.
9. B. Farhang-Boroujeny, Signal Processing Techniques for Software Radios. 2008.
10. H. Harada and R. Prasad, *Simulation and Software Radio for Mobile Communications*. Artech House, 2002.
11. J. Bard and V. Kovarik, Software Defined Radio: The Software Communications Architecture. San Francisco: Wiley, 2007.
12. *IEEE Standard Glossary of Software Engineering Terminology*, IEEE standard 610.12, 1990.
13. F. Kragh, C. Dietrich, et al, *Example Labs for use with OSSIE 0.6.1*. Available: http://ossie.wireless.vt.edu/trac/wiki/UserGuides
14. Universal Software Radio Peripheral Data Sheets. Available: http://www.ettus.com/
15. C. Dym et al, "Engineering Design Thinking, Teaching, and Learning", Journal of Engineering Education, pp. 103-120, Jan. 2005.
16. GNU radio project software. Available: http://gnuradio.org/trac
17. SCARI software, Communications Research Centre Canada. Available: http://www.crc.ca/en/html/crc/home/research/satcom/rars/sdr/products/scari_open/scari_open
18. SDR Forum Education Working Group Home Page. Available: http://www.sdrforum.org/wiki/tiki-index.php?page=Education+Working+Group+Home+Page