The Development and Deployment of Embedded Software Curricula in Taiwan

Shiao-Li Tsao
Dept. of Computer Science
National Chiao Tung University,
Hsinchu, Taiwan
sltsao@cs.nctu.edu.tw

Tai-Yi Huang
Dept. of Computer Science
National Tsing Hua University,
Hsinchu, Taiwan
tyhuang@cs.nthu.edu.tw

Chung-Ta King
Dept. of Computer Science
National Tsing Hua University,
Hsinchu, Taiwan
king@cs.nthu.edu.tw

ABSTRACT

The Embedded Software (ESW) consortium under the Very-Large-Scale Integration (VLSI) Circuits and Systems Education Program which is supervised by the Ministry of Education (MOE) of Taiwan was initiated in 2004 to develop and promote embedded software education. One of the major missions for this consortium is to develop and deploy embedded software curricula which are not well established in traditional computer science (CS) and electrical engineering (EE) education programs in Taiwan. Therefore, the ESW consortium has spent three years in developing embedded software curricula with total twelve new courses. In this paper, our strategies, implementation and experiences for developing and deploying embedded software curricula are presented.

Keywords

Embedded Software (ESW), Educational Curricula, System-on-Chip (SoC) Design

1. INTRODUCTION

Taiwan is one of the leading countries in the world in manufacturing information technology (IT) and integrated circuit (IC) products. To improve global competitiveness of local IT/IC industries, Taiwan government is seeking strategies to help local companies to migrate their businesses from manufacture-oriented products to design-oriented products. Embedded software (ESW) that provides high add-on values for IT/IC hardware is regarded as one of the key strategies [3]. Therefore, the Ministry of Education (MOE) of Taiwan has initiated the Embedded Software consortium under the Very-Large-Scale Integration (VLSI) Circuits and Systems Education Program since 2004 to promote ESW education and build up the fundamental research and development energies for embedded software technologies [1][2]. The missions of the ESW consortium are to develop embedded software curricula for universities, to promote embedded software education and activities, and to bridge the connections between industries and universities to exchanges ideas and new technologies. Hence, several task groups under the ESW consortium are formed to handle the specific working items and missions for the consortium [4].

The development and deployment of ESW curricula are the major missions for the consortium and are handled by the curricula development task group. The task group first investigates curricula for embedded software, embedded system, computer science, and computer engineering developed by ACM, IEEE-computer science (IEEE-CS) and leading universities in the world [5][6][7][8][9][10][11][12][13][14]. Then, the reference ESW curricula were proposed by the task group and approved by the ESW advisory board, formed by distinguished professors and industrial executives. The curricula consider the needs from local industries, cooperate with the existing courses and curricula for computer science (CS) and electrical engineering (EE) programs, and offer both theoretical and practical trainings to students. The ESW consortium also provides a tailoring guideline for universities to adopt the curricula in their education programs. Based on the curricula, course modules which are missing or not well established in the current programs are developed. Cooperating with the development of the curricula and course modules, a deployment program which helps universities to develop their own ESW curricula in the undergraduate and graduated programs, and sponsors universities to establish laboratories for the ESW education is also initiated. With the investment for the past three years, the curricula and infrastructures for ESW education have been successfully established in many universities in Taiwan. Now, the curricula development task group moves to the next stage to establish a database for hands-on labs which are extremely important to ESW education, and keep developing course modules to address new ESW technologies such as hardware/software co-design, embedded multi-core programming, simulator, emulator, and debugger design for SoCs/embedded processors.

The rest of the paper is organized as follows. Section 2 presents the strategies to develop and deploy ESW curricula. Section 3 describes the curricula and their development processes. Section 4 discusses the results and experiences, and finally, Section 5 summarizes our future work.

2. OUR STRATEGIES

The development and deployment strategies were proposed in the very beginning stage, and they have been enhanced according to the feedbacks from professors and the inputs from MOE representatives and members in the ESW advisory board. Figure 1 and Figure 2 illustrate the flow charts for the development and deployment of ESW curricula, respectively. During the development phase, the reference ESW curricula are first developed based on the curricula suggested by ACM, IEEE-CS, and other universities. The members in the ESW advisory board review the ESW curricula and provide comments and suggestions to improve the curricula. According to the reference ESW curricula, missing courses and the courses without mature lectures and hands-on lab materials are determined. Then, the projects which develop the lecture materials together with hands-on labs for these courses are initiated. The ESW office is responsible for seeking candidates for the project leaders that must be senior or experienced professors in the fields related to the course subjects. The project leader further needs to form a development team which has at least four domain professors to develop the course. The MOE then sponsors the team for three years with a total budget about USD 80,000. During the first year of the project, the professors in the team develop course and hands-on lab materials. The deliverables are lecture slides and notes for hands-on labs. Professors who develop the course are also encouraged to write a text book cooperating with the slides and lab materials. They will receive additional funding for writing a text book. During the second and third year, the team receives feedbacks and inputs from other professors using the course materials, and maintains the slides, notes and the text book. Besides the development of course materials, the professors need to use the course materials in their own courses. The trial run procedure is to receive feedbacks from students and to improve the labs and course. Also, they have to introduce the course to other universities in the curricula promotion workshop. The curricula promotion workshop is an important event to introduce and promote the ESW curricula. It is held twice a year and hosted by the ESW consortium. This is also an important occasion for the development teams to gather inputs from other professors and to exchange the ideas and teaching experiences with them. Besides curricula promotion workshops, all project leaders of the courses have to attend the regular consortium meetings which are held quarterly. In the meetings, project leaders have to update their status and may request additional supports from the consortium or the MOE.

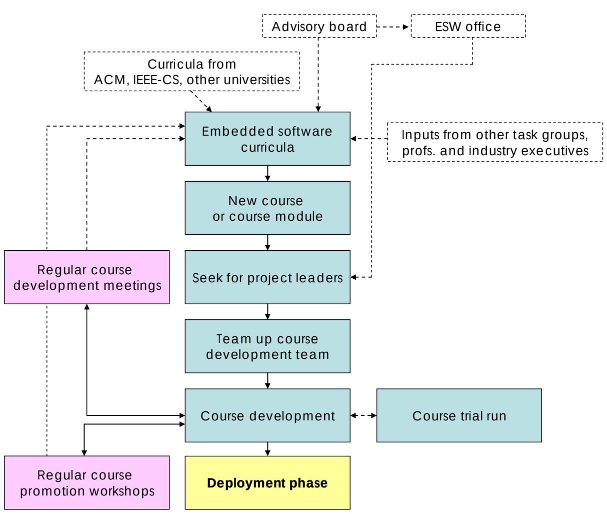

Figure 1. Development strategies and flowchart

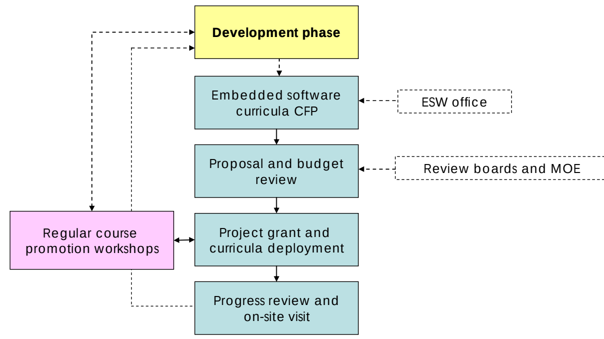

Figure 2. Deployment strategies and flowchart

After new course materials become stable, the new course moves to the second stage, i.e. deployment stage, which is to encourage other professors to use the course in their universities. In order to encourage the universities to use the curricula and materials developed by the ESW consortium, a deployment program which is also sponsored by the MOE of Taiwan is offered. The deployment program helps universities to develop their own ESW curricula for undergraduate and graduated programs and build up the infrastructure such as a dedicated laboratory for the ESW education. The program normally sponsors universities for two years, and the total budget for two years is about USD 100,000 to USD 150,000 per university. The budget is mainly used for equipment purchase and to setup the ESW education laboratory. Besides the funding from the government, the universities who apply the program have to commit a dedicated space for the laboratory and at least 20% matching fund of the total budget.

According to experiences learned by other sponsorship programs for curricula deployment, course-based sponsorship programs have several drawbacks. For example, universities receive budgets and have the course in their programs in the first one or two years, but the course may be removed from their programs if there are no continuative funding and project support. Also, the course might overlap with the other existing courses and cannot be integrated into the existing curricula of the educational programs. To overcome these problems and also consider that the ESW curricula normally require equipment support, the deployment program for the ESW curricula is modified as a curricula-based and laboratory-oriented sponsorship. The reference ESW curricula and a list of new course modules developed by the ESW consortium are provided in the call for proposal (CFP). Universities who would like to submit a proposal has to provide their own ESW curricula which could be a complete new one, a tailored version from the reference curricula, or enhancement curricula integrated with their existing programs. They must commit to have a dedicated laboratory for the ESW education and also have new courses or courses with enhancements in their curricula. In order to deliver the new courses or enhance their existing courses, universities should propose their needs for the equipment which must be installed in the dedicated ESW education laboratory. The review committee evaluates the proposals and decides the amount of budget based on their curricula and their plans for establishing the ESW laboratories. This approach makes the sponsorship program more effective and efficient than the previous course-based sponsorships.

Besides establishing the ESW curricula and laboratory, professors have to participate the semi-annual curricula promotion workshops. They have to present their experiences, and demonstrate their results such as hands-on labs from students. The sponsorship program is for two years but has an annual review to evaluate the progress and determine the budget for the second year. The statistical information such as the number of equipment purchased, the number of new courses which are delivered or enhanced, the quality of the course and lab materials, and the number of students enrolled these courses are collected and evaluated. The review consists of on-site review which checks the laboratory, equipment, course modules and interviews with professors and students, and paper document reviews.

3. EMBEDDED SOFTWARE CURRICULA

Universities in Taiwan are categorized into universities and technical universities which provide quite different training to students and have different education objectives. Universities train students with research, design and development abilities. Technical universities offer trainings to students with technical skills and implementation abilities. To address the different needs, the curricula deployment task group has developed two reference ESW curricula for universities and technical universities. Also, the courses and materials are different in the two curricula. The sponsorship programs for the course deployment for universities and technical universities are also separated. Figure 3 depicts the curricula for universities. The courses are classified into embedded hardware and System-on-Chip (SoC) courses which focus on hardware design, embedded system software courses which introduce system software for embedded systems, and embedded application courses which provide training for developing embedded applications and services. Also, the courses can be categorized into three levels, i.e. fundamental courses for undergraduates in the third and fourth year, intermediate courses for senior undergraduate students or graduated students, and advanced courses for graduated students such as master or PhD students. According to the proposed curricula, eight courses are either newly developed or enhanced. Except the embedded middleware course which is still being developed, other seven courses had been developed. Brief descriptions of the eight courses are:

- **Embedded software programming**: Programming languages and design considerations for applications running on embedded systems and SoCs are quite different from these on general purpose PCs. The course introduces several well-know programming languages for embedded systems such as embedded C/C++, Java for mobile edition (J2ME), and C#. Moreover, the course introduces optimization technologies such as down-sizing, power consumption and etc. for embedded software development. The course outline is as follows. They are (1) introduction of embedded systems and embedded operating systems, (2) basics of embedded programming languages, (3) characteristics of embedded programming languages, (4) embedded C++/C, (5) Java for embedded systems, and (6) C# for embedded systems.

- **Embedded tool chains**: This course consists of five modules and a case study on an open-source tool chain. The modules are (1) tool chain introduction, (2) basic compilation techniques, (3) assembler and linker, (4) optimizations for high-performance and code density, and (5) energy-saving methodologies. The case study gives students hands-on experiences on porting an open-source tool chain to both a RISC and a DSP platform.

- **Implementation of embedded operating system (EOS)**: EOS which is different from OS has to deal with many hardware-specific functions and/or customized optimizations. A new course focusing on bootloader design, EOS design, porting and optimizations is thus developed. The topics that the course address are (1) fundamental concepts of embedded operating systems such as process management, scheduler, device drivers, timer, interrupts, exceptions, memory management, and power management, and (2) embedded OS implementations including porting, starting from scratch, and hardware/software co-design issues.

- **Embedded middleware**: Traditionally, electronics vendors are very adept at providing embedded hardware platforms and low system-level software at competitive cost; however, they often lack necessary software skills to develop a complete, complex embedded applications and systems that have much higher profit margins. Developing these high-margin embedded applications require good understanding of the various embedded middleware platforms and knowledge in emerging international middleware standards that provide interoperability among different embedded systems and hardware vendors. The goal of this course is to educate students about fundamental concepts in embedded middleware as well as hands-on programming skills on these emerging middleware platforms and standards. The lab of this course contains two parts. The first part includes individual programming assignments which each student will practice programming APIs of different embedded middleware systems (such as .Net Compact, J2ME, Jini, RFID middleware, etc.). The second part includes a course project where students will form teams to complete complex embedded applications by drawing components from different embedded middleware systems. The topics of the course include (1) programming embedded middleware, (2) emerging standards in embedded home and multimedia middleware, (3) embedded database and RFID middleware, and (4) location middleware.

- **I/O and device driver**: This course is developed to answer strong demand by system companies for well-trained engineers in porting and writing device drivers. To satisfy this need, the section of I/O management in the fundamental course of Operating System Concepts with emphasis on device drivers and extensive implementations is elaborated. Students are required to implement drivers for devices such as communication interfaces, and audio and video cards. In addition, for hardware-accelerated capabilities, students are trained to implement a device in a FPGA extension and to access it through a device driver. The course outline is as follows. They are (1) basics of embedded systems, program development and tool chains, and embedded operating systems, (2) introduction to embedded processors and platforms, (3) I/O subsystem – hardware and software perspectives, (4) debugging embedded software, (5) interfaces and peripherals, (6) kernel modules and device driver designs, (7) Windows CE: architecture and device drivers, and (8) embedded Linux: architecture and device drivers.

- **Embedded real-time system**: This course extends the one on Embedded OS Design and Implementations to deal with real-time characteristics required by an embedded real-time system and real-time operating system. Modules such as real-time scheduling, resources management, issues on priority inversion, and real-time driver architecture are essential. Students will acquire knowledge through both in-class lectures and implementation projects on open-source real-time operating systems. The topics of the course include (1) introduction to real-time systems and embedded real-time operating systems, (2) real-time scheduling algorithms, (3) implementations of embedded real-time operating systems, and (4) case studies.

- **Embedded compiler design**: This course starts with general introduction and background information on embedded compilers. It next shifts its focus to optimizing skills for developing efficient and energy-saving object code for power aware embedded systems. The course consists of four dependent modules: (1) ILP compiler introduction, (2) DSP compiler introduction, (3) compilers for embedded parallel processors, and (4) data dependence analysis for embedded heterogeneous processors.

- **ESW for networked SoCs**: The course is the project-oriented course that aims to give students a system level practice on embedded system and software. Instead of focusing on a stand-alone system, this course addresses more on the networked SoC system. Heavy hands-on experiments are required for students to build an intelligent transportation coordination system on robot cars. The course outline is: (1) Basics of Networked SoC systems (2) Real-time programming (3) Wired-line, wireless, sensor technologies and programming (4) Low power technologies and programming (5) System integration, verification and validation.

Different from universities, technical universities train students to have more programming and engineering skills. The ESW consortium thus invites professors in the technical universities to develop suitable courses and curricula for technical universities. Figure 4 illustrates the reference ESW curricula for technical universities. The courses for technical universities are classified into embedded hardware/SoC courses and embedded software courses. Similar to the curricula for the universities, courses are categorized into fundamental, intermediate and advanced levels. Some of courses are specially designed and developed for technical universities. Some of courses are tailored from the course modules developed for universities but hands-on labs are enhanced. The course development and curricula promotion process are identical to the universities, but faculties from technical university are invited to develop the courses. Currently, there are four new courses which have already developed. The objectives of the development courses are briefed below.

- **Introduction to embedded system**: Embedded systems are not only applied to IC or IT industries, but also widely employed on control, mechanical, aerospace, biological, and automobile industries. A general introductory course on embedded system could be very useful to these students who are not major in CS or EE but need embedded system skills or knowledge on their domains. Therefore, the course is designed and developed. This course is an entry level course and provides general introduction and hands-on practices on all aspects of an embedded system. The course outline is as follows. They are (1) basics of embedded systems, (2) design flow of embedded systems, (3) embedded processors, programming and I/Os, (4) non-OS embedded software programming and embedded operating systems, and (5) embedded system development platforms and system integrations.

- **Embedded system labs**: 8-bits micro-controllers such as 8051 are normally used as the education platform for micro-computer system training. Advances in IC technologies, the usages of 16-bits, 32-bits embedded processors right now become the market trends. To help technical university students to step into the high-end embedded system development, the course uses ARM as the reference architecture to introduce the embedded system design. The course provides complete and step-by-step hands-on trainings for students to practice the development of ARM-based embedded systems. The topics of the course include (1) labs for cross compiler tool installation, (2) labs for C programming compilation and download, (3) labs for ARM assembly programming development, (4) labs for more complex ARM assembly programming, (5) labs for basic Linux system calls, and (6) labs for simple TFTP program and simple HTTP program.

- **Interface design**: Interfaces such as RS-232, parallel ports for computer I/O systems are frequently used in the technical universities. However, these interfaces are upgraded to faster and more complicated interfaces such as I2C, I2S, USB, P1394, PCMCIA, and etc. To help students to learn such new interfaces and have hands-on experiences on these interfaces are the major goals of the course. The topics of the course include (1) development platforms and tools for I/O design, (2) memory-mapped I/O and port I/O, (3) GPIO, (4) I2C, (5) I2S, (6) DMA, (7) flash memory, and (8) analog and digital (AD) and DA converting.

- **Implementation of USB devices/drivers**: As USB quickly becomes a primary choice for the interfacing protocol, it is critical for a portable embedded system to provide USB ports alone with USB device drivers and firmware. To meet such a demand, this course that is specifically designed for technical universities is proposed. This course provides a comprehensive hands-on training on USB technology including device driver, firmware, and physical-layer IP. Each student needs to develop software on a host to recognize and control a device through USB interface. Students also learn to use a set of tools such as a USB analyzer, oscilloscope, and function generator to debug their software. The topics of the course include (1) introduction to USB, (2) related tools, (3) USB protocols, (4) basics of firmware programming, (5) HID programming, (6) USB device driver, (7) USB analyzer, (8) USB host design, (9) HID classes for USB host, and (10) USB host controller driver.

4. HANDS-ON LAB MAP

ESW courses usually require heavy hands-on practices but the development of hands-on labs needs significant efforts to make them useful and complete. To maintain the hands-on labs including teaching assistant (TA) notes, reference source codes or reports, and knowledge and experiences learned from the labs are extremely important. Therefore, the ESW consortium built a hands-on lab database and tried to establish a hands-on lab map for references since last year. The ESW consortium requests each course development project to turn in at least four hands-on labs which include a step-by-step TA notes, and a hardware and software platform for the lab. Also, for these universities who received the financial support by the MOE, they also have to turn in at least 4 sets of hands-on labs for each course at its final review. These notes and platforms were sent to other professors for peer reviews. The review teams need to reproduce the labs based on the lab notes and platforms. The review comments are sent to the development team to improve the note qualifies. Once the lab notes are approved and these hands-on labs can be checked in the database. Currently, more than 80 hands-on labs have been established in the lab database which is shared by all professors who adopt the course modules or lab modules. We label each lab with an index of *CourseNum-UnivNum-LabNum*. For example, a lab module with a label of 002-01-03 indicates that it is the third lab module developed by the first university for the second course.

For ease of reference, we classify these 80 modules by three categories: platforms, operating systems, and tool chains. There are currently 14 embedded platforms and 5 different operating systems. Due to the space limit, we only give out partial information of each classification table. The complete tables are accessible at the official web site of the ESW consortium [2].

4.1 Classification by Platforms

Table 1 gives part of our classification by platforms. The first column lists the platforms used in each set of labs. The second column lists the operating systems used by these labs. An empty slot of this column indicates that no operating system is installed. The third column gives the topics of labs and the fourth column gives the labels. This table helps professors to easily identify the platform suitable to their teaching needs. In addition, it serves as a map to develop unavailable modules for a selected platform, avoiding redundant efforts.

4.2 Classification by OS

There are five different operating systems used among all lab modules. Figure 5 shows the names of these operating systems and the number of labs installed with each operating system. Linux-based operating systems are strongly favored by these hands-on labs. Table 2 gives part of our classification by operating systems. The first column shows the names of the operating systems and the second column shows the platforms. The third and the fourth columns show the topics of labs and their labels. By this table, an interested professor can easily find out the topics of labs and platforms for his/her favored operating systems. Similarly, a new course on embedded operating systems or related topics can enhance this table by developing unavailable lab modules.

4.3 Classification by Tool Chains

One major category of courses in the ESW curricula is to develop tool chains for embedded platforms. These tool chains are mainly compilers and debuggers. There are totally 16 labs related to the development of tool chains. We classify these labs by the developed tools.

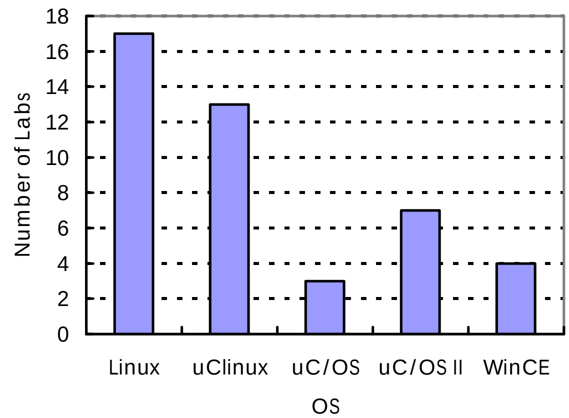

Figure 5. The number of labs with each OS

5. RESULTS AND EXPERIENCES

The ESW consortium has initiated twelve course development projects since 2004. Eight courses are for universities and the other four courses are designed for technical universities. The twelve projects received about total USD 600,000 for the course development, and total USD 150,000 per year for maintenance. 43 professors from more than 20 universities involve the course development. Among 43 professors, 18 professors are from EE and related departments, and the others are from the CS department. More than five curricula promotion workshops which introduce, promote the course and curricula, and demonstrate the results were held. Also, more than 400 attendees including professors, students, and engineers from industries participated the events.

As for the sponsorship program to deploy the ESW curricula, the MOE of Taiwan received 24 proposals and approved eleven proposals last year. A total USD 600,000 is funded for eleven universities for the first year. The amounts of budget are the same for the second year. The statistical data for the first year shows that the ESW education infrastructures have been established in the eleven universities. More than 30 new courses or courses with enhancements are lectured in the eleven universities in the first year. A total of 1000 students enrolled these courses under the ESW curricula deployment program. In this year, the ESW consortium also announced the call for proposal for technical universities, received more than 80 proposals, and granted more than 40 proposals. The ESW curricula will be deployed over technical universities soon.

Several issues and challenges were raised in the past three years. The first one is about the common education platform. The curricula developed by the ESW consortium require heavy hands-on practices but the hands-on platforms are usually different in the course modules. It is very expensive for universities to purchase several lab platforms and also very difficult for professors, teaching assistants (TAs) and students to learn different lab platforms. To resolve this problem, to develop different sets of hands-on labs over various platforms or to require all hands-on labs to be developed over one or two common hardware platforms are two possible approaches. The first approach requires the investment of time and teaching assistants' resources in developing reference labs. The second approach requires defining one or two hardware platforms which meet the needs by all courses. Since the courses normally require open sources and rich technical supports from vendors or open source community, the selection of platform is quite difficult at this stage. It is believed after running the curricula for more years, the platforms will be harmonized and the reference hands-on labs over different platforms become rich. The second challenge is about the development of hands-on labs. As mentioned in the previous section, the ESW consortium has started to establish a database for hands-on labs to resolve this issue since last year.

6. CONCLUSIONS AND FUTURE WORK

The paper presented the strategies to develop and deploy ESW curricula in Taiwan. First, the reference ESW curricula that provide a comprehensive and modern training to students were proposed for different needs of universities and technical universities. New courses and hands-on labs were developed to support the curricula. Then, the deployment program which encourages universities to establish ESW curricula and helps universities to establish the ESW education laboratory is also offered. Our future working items are to seek a common teaching platform, construct a complete database for the hands-on labs and hands-on experiences, and develop more advanced courses to support ESW research and education.

ACKNOWLEDGEMENT

The authors would like to thank the Ministry of Education of the Republic of China for financially supporting this program and the ESW consortium.

7. REFERENCES

[1] The SOC Consortium, VLSI Circuits and Systems Education Program, Ministry of Education, Taiwan, http://moesoc.ee.ntu.edu.tw.

[2] The ESW Consortium, VLSI Circuits and Systems Education Program, Ministry of Education, Taiwan, http://esw.cs.nthu.edu.tw/.

[3] Industrial Economics and Knowledge Center, "The supply-demand analysis of high-tech engineers in key industries and deficit-recovery solutions," Technical report, Industrial Technology Research Institute, 2003.

[4] Tai-Yi Huang, Chung-Ta King, Yin-Tsung Hwang, and Youn-Long Steve Lin, "The embedded software consortium of Taiwan," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3):612-632, August 2005.

[5] ACM Curricula Recommendations, http://www.acm.org/education/curricula.html.

[6] David Jeff Jackson and Paul Caspi, "Embedded systems education: future directions, initiatives, and cooperation," First Workshop on Embedded System Education (WESE), 2005.

[7] Alberto Luigi, Sangiovanni-Vincentelli, and Alessandro Pinto, "Embedded system education: a new paradigm for engineering schools?" First Workshop on Embedded System Education (WESE), 2005.

[8] Suehee Pak, Eunha Rho, Juno Chang, and Moon Hae Kim, "Demand-driven curriculum for embedded system software in Korea," First Workshop on Embedded System Education (WESE), 2005.

[9] Peter Marwedel, "Towards laying common grounds for embedded system design education," First Workshop on Embedded System Education (WESE), 2005.

[10] Alberto L. Sangiovanni-Vincentelli, and Alessandro Pinto, "An overview of embedded system design education at berkeley," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3): 472 - 499, August 2005.

[11] Philip Koopman et al., "Undergraduate embedded system education at Carnegie Mellon," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3): 500 - 528, August 2005.

[12] Janos Sztipanovits et al., "Introducing embedded software and systems education and advanced learning technology in an engineering curriculum," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3): 549 - 568, August 2005.

[13] Rudolph E. Seviora, "A curriculum for embedded system engineering," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3): 569 - 586, August 2005.

[14] P. Caspi et al., "Guidelines for a graduate curriculum on embedded software and systems," ACM Transactions on Embedded Computing Systems Special Issue on Embedded Systems Education, 4(3): 587 - 611, August 2005.

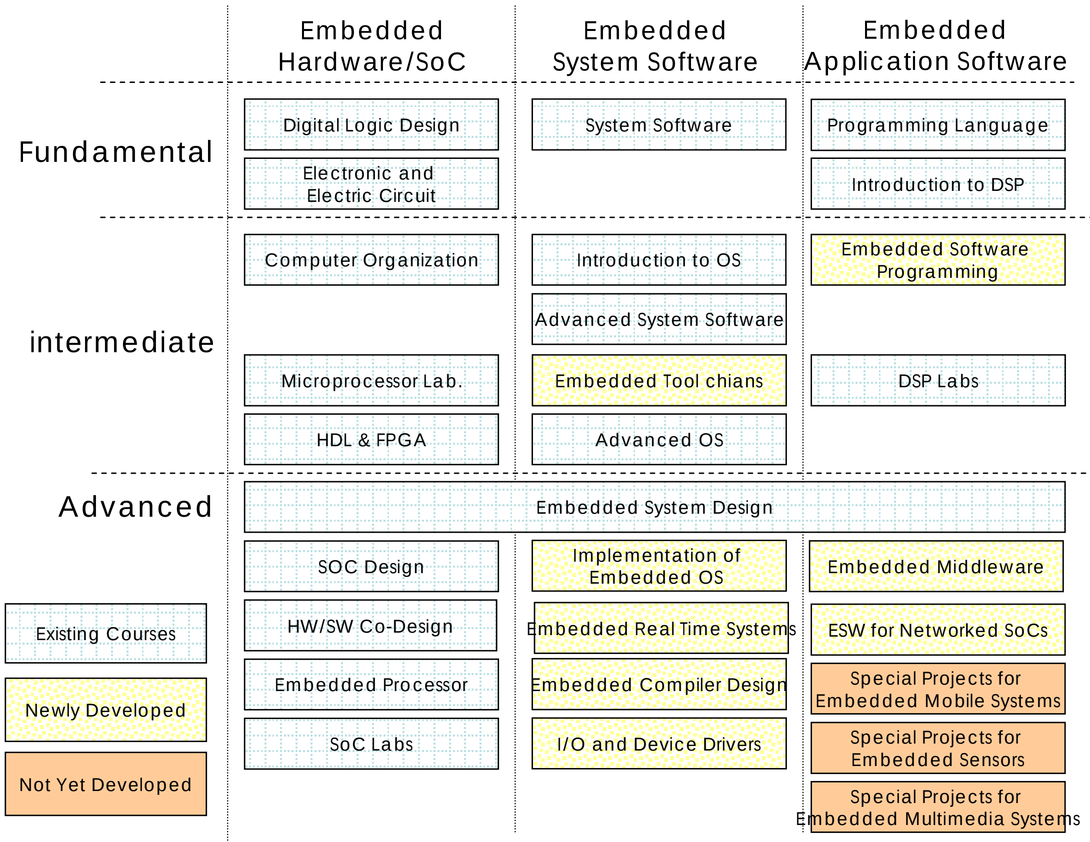

Figure 3. Embedded software curricula for universities

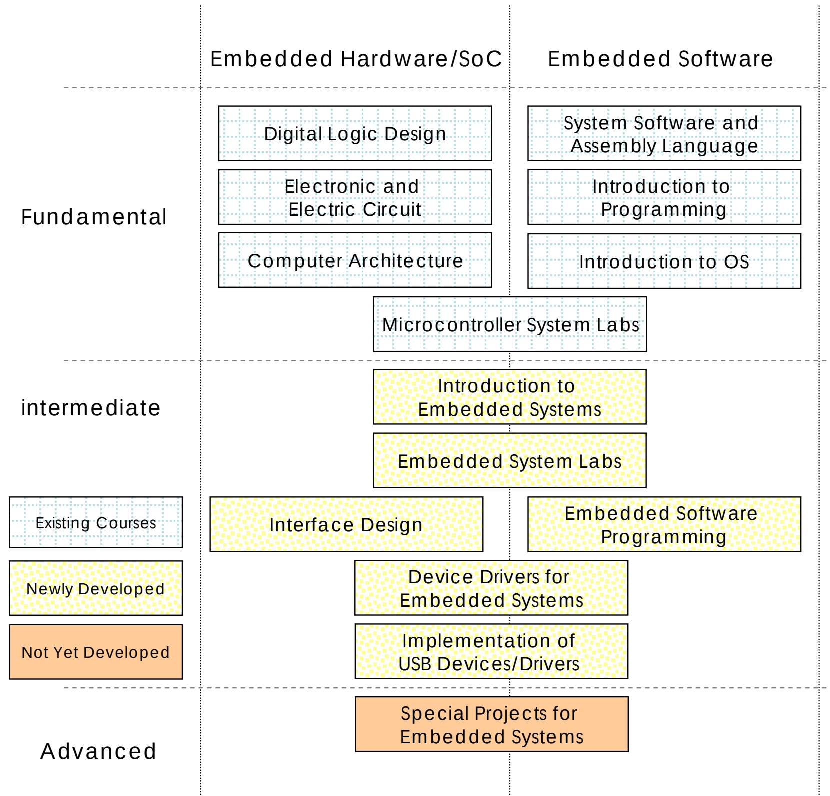

Figure 4. Embedded software curricula for technical universities

Table 1. The classification of labs by platforms

Platform	**OS**	**Topic**	**Label**
Creator (S3C2410)		LCD Display	[001-02-01]
	Linux 2.4[001-02-02]	MP3 Player	[001-02-03]
		RT-OS module FireLinux	[001-02-04]
NET-Start!W3001 (ARM S3C4510B)	uClinux 2.0 [001-03-04]	Bootloader	[001-03-03]
		Switch and 7-segment	[001-03-05]
		Interrupt Handlers and Time Delay	[001-03-06]
		DMA	[001-03-07]
	uC/OS [001-03-08]	Scheduler: Priority-inversion Issue	[001-03-09]

Table 2. The classification of labs by operating systems

OS	Platform	Topic	Label
Linux	Creator (S3C2410)	Porting Linux 2.4	[001-02-02]
		MP3 Player	[001-02-03]
		RT-OS Module FireLinux	[001-02-04]
	TI OMAP (5910)	Porting Linux 2.4	[001-04-01]
		IPC (Share Memory & Semaphore)	[001-04-01]
		PThread Programming	[001-04-01]
		Embedded GUI Programming	[001-04-01]
		Networked Digital Camera	[001-04-01]
uClinux	NET-Start!W3001 (ARM S3C4510B)	Poring uClinux 2.0	[001-03-04]
		Bootloader	[001-03-03]
		Switch and 7-segment	[001-03-05]
		Interrupt Handlers and Time Delay	[001-03-06]
		DMA	[001-03-07]