DEVELOPING A RESPONSIVE AND RESPONSIBLE CURRICULUM FOR COMPUTER ENGINEERING

Simon Winberg

1University of Cape Town (South Africa)
Simon.Winberg@uct.ac.za

Abstract

This paper reports on the curriculum refinement process for a four-year BSc in Electrical and Computer Engineering degree programme at the University of Cape Town in South Africa. The curriculum changes are being made in collaboration with research institutions and companies interested in employing graduates of the programme. This paper focuses on developing an effective curriculum that is responsive in terms of equipping graduates with the qualities and practical expertise that the stakeholders value, while being responsible by ensuring graduates obtain the fundamental theoretical knowledge that underpins professional practice. The findings show that all the stakeholders want skilled and knowledgeable graduates. Graduates’ knowledge should not be limited to computer engineering theory, but should also include a breadth of contextual and practical situations. It was further found that the collaborators’ requests could be categorised into four main knowledge areas: 1) protecting the fundamental core; 2) essential professional expertise; 3) problem-solving in context; and 4) implementation tools and techniques. The term ‘T-Shaped people’ [1] provides a useful representation of the desired graduate qualities: the down-stoke of the ‘T’ represents disciplinary depth, while the cross-stroke represents contextual breadth in different contexts. The paper presents a graduate knowledge model as a useful visual tool to assist in curriculum selection to support students’ transition from university to professional practice. Conclusions reflect on stakeholders’ requests and compromises made, emphasising the consensus to ‘protect the fundamental core’ but allowing freedom in both development techniques and problem-solving experience in both low- and high-levels of contextualized practice.

Keywords: Curriculum design, computer engineering, engineering education, industry-university partnerships.

1 INTRODUCTION

Computer engineering is a discipline that integrates principles of electrical engineering and computer science for the purpose of building computer systems and computer-related electronic devices [2]. This paper focuses on the development of an effective university curriculum for computer engineers that is responsive in terms of equipping graduates with the qualities and practical expertise that industry stakeholders value, while being responsible by ensuring that graduates build the fundamental theoretical knowledge that underpins professional practice in this discipline.

This paper reports on the curriculum refinement process for a four-year BSc in Electrical and Computer Engineering (ECE) programme at the University of Cape Town (UCT), South Africa. This is an engineering degree, where the term ‘electrical’ in the programme title emphasises that graduates acquire both electrical and electronic knowledge as well as computer science and software engineering knowledge. This curriculum refinement process is intended to continue for the duration of the programme, following good practice as described by the European Science Foundation [3].

The structure of this paper is as follows: Section 2 presents the main reasons for the curriculum renewal and the collaborators involved in this process; Section 3 outlines the methodology; Section 4 presents on the findings. Part of the methodology involves developing a ‘graduate knowledge model’, which is presented in Section 5 and which served as a useful visualization aid during discussions. Section 6 concludes this paper with generalized reflections and recommendations for this form of curriculum renewal process.

2 BACKGROUND

Computer engineering concerns the design of specialized computers, often embedded or special-purpose computers, system level software, peripheral drivers, and development of computer-related
devices [4]. Computer engineers can access a wide variety of career options[5], from the specific practice of computer engineering (i.e., designing computer architectures and microchips) to broader computer-related careers, some of which are more often associated with computer science (e.g., software engineering and programming) and others more associated with electrical engineering and electronics (e.g. design of printed circuit boards and electric products).

At university, computer engineering students usually study a combination of electrical engineering and electronics subjects together with computer science subjects. In addition, these students learn specialized topics closely related to their chosen field, such as digital microcomputer design and use of hardware description languages [4]. Recently, the landscape of computer systems has changed [6, 7]; this is partly due to a 'brick wall' in terms of the speed of individual CPU cores and the exponential growth on power consumption as transistors double on a chip. As a result, computer architectures and software need to be made parallel to perform computations more quickly [8]. Computer designers also need to be skilled in establishing effective balance in computer size, weight and power consumption attributes, referred to as SWAP characterises [9] – an aspect that is becoming more important considering the growing popularity of mobile systems. Industry, research institutions and university research groups all want computer engineering graduates adequately prepared to address these challenges in the new landscape of computing.

The curriculum renewal process for the Electrical and Computer Engineering programme discussed in this paper was started for two main reasons. Firstly, it was started in response to South African industries' need for graduates suitably prepared for understanding and using the theories and technologies associated with the new landscape of computing. The second reason is that the Department of Electrical Engineering at UCT decided to start a department-wide renewal process of all the undergraduate programmes offered. This broader renewal process was needed to correct the misalignment that had occurred over the years between programmes. For example, certain prerequisite courses were no longer providing the fundamental theory and practices expected. A further reason for the renewal process came from the Engineering Council of South Africa (ECSA) in terms of recommendations for curriculum changes and degree certification procedures. This reworking of the curriculum has been underway since 2009.

While there are many curriculum renewal projects underway in the different undergraduate programmes offered by the department, this paper focuses on the curriculum renewal for the Electrical and Computer Engineering (ECE) programme. The ECE degree programme is one of three degree programmes offered by the department. On its own, this ECE renewal process is itself a large project that currently involves five faculty involved in teaching ECE courses, with additional representatives from the other degree programmes and from other departments. In order to refine the focus of this paper, the collaboration with industry and research institutions focused on the two final year core courses in the ECE programme.

In subsection 2.1 the collaborators from industries and research institutions are discussed, while Subsection 2.2 describes the ECE degree programme.

### 2.1 Industry and Research Collaborators

The ECE curriculum renewal process involved collaboration with both industries and research institutions in South Africa. Three industries and two state research institutions are involved in the collaboration.

Two of the industry partners are involved in development of special-purpose high-end embedded computer systems, developing both hardware platforms and covering a broad range of embedded software development, including coding in hardware description languages. The third industry partner is involved with development and programming of high performance computing systems. Not all the industry collaborators wanted their company name mentioned in publications; but all parties were willing to have their recommendations and decisions resulting from this collaboration made public.

The collaborators from the state research institutions (referred to as ‘research collaborators’ in this paper) were willing to have their respective organizations mentioned. The organizations were: the Karoo Array Telescope (KAT) Project, and the Council for Scientific and Industrial Research (CSIR).

### 2.2 Structure of the ECE programme

Students enrolled for computer engineering usually study a combination of electrical engineering and computer science courses in addition to the requisite physics and mathematics courses. The ECE
programme offered by the Department of Engineering at UCT is a four year programme. While students are expected to complete the programme within four years, many students take longer. Fig. 1 represents the recommended computer engineering curriculum as described by the Association for Computing Machinery (ACM) [2], which is generally followed in the ECE degree offered at UCT. The UCT ECE curriculum has, however, made some changes to the ACM's recommended structure, which are elaborated below in relation to the 2009 curriculum that was in effect prior to the current curriculum renewal process.

<table>
<thead>
<tr>
<th>Fundamentals</th>
<th>Focused Computer Engineering</th>
<th>Advanced</th>
</tr>
</thead>
<tbody>
<tr>
<td>Math, Science and Computing fundamentals</td>
<td>Prescribed CE Core Topics</td>
<td>Elective CE Topics</td>
</tr>
<tr>
<td>1 year</td>
<td>1.5 years</td>
<td>1.5 years</td>
</tr>
</tbody>
</table>

Fig. 1: Recommended computer engineering curriculum structure based on [2].

The first year, illustrated on the left of Fig. 1, focuses on students building a ‘core’ foundation of mathematics and science. This core, as explained by the ACM curriculum task force [4], is necessary for students to adequately engaged with subsequent courses that are more closely related to the specifics of electronics and computer engineering. Students enrolling for the programme should ideally have done computer science at the school level and should have excellent computer literacy skills. The students should also have some prior knowledge of programming and electronics either in formal school subjects or as informal extramural activities, or as a hobby.

The first year ECE subjects constitute: mathematics (including algebra, calculus, vectors, statistics and geometry), computer science (including algorithm development and programming), and physics 1. The first year includes some project-based learning and computer-based practice – and some of this experience was built into the electrical engineering 1 course. These practical components were incorporated into the first year for educational purposes, and to build students’ interest in the field and to retain students in the programme. There are no elective courses in the first year. The incorporation of ethics, engineering professionalism, and legal issues is stipulated for various reasons as explained by [10, 11]. The more general aspect of academic ethics, such as understanding and avoiding plagiarism, is incorporated into the first year curriculum.

In the second year, students learn more advanced programming, and develop their understanding of the design and implementation of computer hardware and aspects of software engineering. The second year curriculum, like the first year, is so tightly packed with essential courses that there is almost no leeway for elective courses. There is scope to allow some flexibility, such as an additional course. This structure corresponds closely to the ACM recommendations, as illustrated by the ‘CE Core Topics’ shown in Fig. 1.

In the third year the number of elective courses increases; 16 of the recommended 144 credits are elective courses, although students often overshoot this recommendation limit due to fourth year elective course prerequisites (each credit represents approximately 10 hours of coursework). In the model shown in Fig. 1, third year extends over parts of both the ‘focused computer engineering’ and the ‘advanced’ topics. But in terms of elective CE topics within UCT’s ECE programme, the scope for third year electives (11% of the credit year) is rather restrictive.

The fourth year courses cover advanced topics. Some of the courses are inter-disciplinary, to account for the likelihood of graduates needing to have knowledge of other fields to improve their employability and career prospects [10]. A wide range of electives are catered for in the curriculum to allow students to become specialized in particular areas, for example to explore additional mathematical fields or to do further computer-based subjects. A significant portion of the fourth year is devoted to a research project, in which students develop prototype systems and perform experiments and research followed by writing a comprehensive final year report (i.e., minor dissertation). The fourth year also has a variety of elective courses, in total 40 credits (or 29%) of the 136 credit year. Combined with the third year electives, the overall ECE curriculum has a minimum of 0.4 years of electives. While this is a bit
lower than the ACM’s recommendation of half a year of electives, there are certain cases in which students can take additional elective credits resulting in some students completing more than 0.5 elective credit years in the programme.

3 METHODOLOGY

The overall ECE curriculum renewal project is a large undertaking and involves a diverse range of inputs from different people; such as external examiners, accreditation teams and expert consultants among others. Research activities have been undertaken by some faculty members and researchers involved in the ECE programme. The scope of this paper has been narrowed down to focus on correspondence related mainly to students’ experiences in fourth year courses.

The methodology for the ECE curriculum renewal process began by identifying the main objectives for the process over a timeline during which the changes would be implemented and evaluated. The selection of collaborators was based on existing, mainly research-related collaborations that had been in place for many years.

The main objective of this renewal process was to update the curriculum, ensuring that up-to-date tools and technologies were incorporated. A second objective was to consider the input of the collaborators with regard to preparing graduates with the skills that they desire.

The timeline for the ECE curriculum renewal project was in line with the general reworking of all the degree programmes offered by the Department of Electrical Engineering at UCT. A timescale of five years was provided for this: starting mid-2009 and ending in 2014.

Data for the ECE curriculum renewal process was obtained in the following ways: meetings and telephone conversations with collaborators together with archived email correspondence. Minutes were taken during the meetings and notes concerning curriculum requests made during telephonic discussions were kept.

3.1 Gathering insights form collaborators

The industry representatives included both senior engineering professionals and more junior representatives who had graduated from the ECE programme and were working in the field. Three private industries are involved and two public research institutes. In consideration of confidentiality, the interaction has remained at the general level of graduate characteristics, identifying specialized fields and theories of interest, and issues of development tools and technologies in use by professionals.

The need to make compromises where emphasised from the start of the collaboration, specifically clarifying that the fundamentals (mathematics, physics, basic computing) were non-negotiable and that students’ learning experiences should not be restricted to the techniques and tools that the industry partners use. The collaboration was planned around a ‘win-win’ scenario in which all stakeholders gain benefits, specifically:

- The industry partners potentially saving training costs for new employees, with the advantage of new recruits having a deeper understanding of relevant applications areas;
- The research collaborators gaining new recruits that could more quickly become productive members of the research institution and/or company, with the added advantage of knowing how to use some of the tools used within the organisation;
- Students benefit by learning about the latest technologies and tools used in real projects, while also improving their career prospects. Additionally, students could improve their prospects for graduate study by considering the inputs from the research collaborators.

Meetings with collaborators occurred on a biannual basis, generally using a combination of face-to-face meetings with some parties calling in via Skype. Minutes for the meetings were produced. Handwritten notes or emailed summaries were made of telephonic discussions. Changes to the curriculum as well as reporting on the result of changes were discussed in meetings and other correspondence.

3.2 Analysing the data and refining the curriculum

The data analysis followed a process of making lists by reviewing minutes and email correspondence. Five lists were produced to group points into knowledge categories. Most of the input from the collaborators was within the ‘cognitive domain’ of curriculum objectives as defined by Henson [12]
curriculum development framework. Most of the requests corresponding to cognitive domain issues were related to either cognitive knowledge levels 3 or level 5 as defined by Henson [12]. Cognitive knowledge level 3 relates to application and a student’s ability to draw on theories and learned principles to solve problems in particular applications. Cognitive knowledge level 5 involves synthesis, the way a student draws on the prior levels (i.e., knowledge level 1 of disciplinary understanding all the way to level 4 analysis) to build a design solution or device. The collaborators provided less input in regards to Henson’s other knowledge domains of ‘psychomotor’ and ‘affective’. This is likely due to most of this knowledge being implicit in what is needed for students to learn the discipline.

The following knowledge categories were used for grouping the collaborators’ curriculum requests. The knowledge domain and knowledge level as defined by Henson [12] is shown in parentheses for each category.

1. Desired graduate qualities (affective / valuing);
2. Techniques (with theory support) that graduates should learn (cognitive/comprehension);
3. Application contexts and problems students should experience (cognitive/application);
4. Types of systems / projects students should do (cognitive/application); and
5. Important professional qualities (affective / organizing).

This reflective practice was done mainly by an individual researcher with some correspondence with faculty colleagues and with industry partners to clarify recorded issues. The requests were prioritised based on how often similar topics were mentioned, and also how strongly a particular request was emphasised. Particular courses in the ECE programme were identified that would be most influenced by the collaborators’ requests. Decisions from this investigation were shared with faculty members who were responsible for teaching these courses, using informal meetings. Relevant suggestions from industry were subsequently built into the courses, and done so at the discretion of faculty responsible for these courses and in the way that they chose to implement the requests if at all.

4 RESULTS

This section presents the findings from an analysis of the data obtained from the collaborators. Four main findings were determined, which are presented in sections 4.1 to 4.4 together with excerpts from the minutes and email correspondence with the collaborators.

4.1 Finding 1: protecting the fundamental core

All collaborators agreed that the fundamental knowledge of the computer engineering discipline needs to be protected. Core engineering knowledge will be made use of regardless of the type of career or application context in which the graduate will practice engineering. A large portion of this fundamental knowledge is the same across engineering disciplines, such as mathematics and physics. This understanding is expressed in the following email excerpt from an industry collaborator:

... I have interviewed more than 100 candidates for ... positions in both South Africa and the United States... Typically, candidates are chosen not for experience in any single technology or environment, but for a strong grounding in the fundamental concepts... A good candidate who knows C# can become productive in Java in a matter of weeks... (Industry collaborator 1)

In terms of ‘trade-offs’ between modifying the curriculum to support new tools versus sacrificing state-of-the-art development tools for greater depth of disciplinary knowledge, the following comments reflect collaborators' general feelings:

...I would strongly encourage any curriculum change to focus on providing a diverse range of ideas and experience rather than knowledge of any particular technology... (Industry collaborator 2)

With an understanding of programming fundamentals, they will be able to go into more detail with it if and when needed. (Research collaborator 1)

The general consensus was thus that the fundamental disciplinary knowledge areas need to be identified and protected during curriculum renewal, as well as when choosing technologies to incorporate into course.
4.2 Finding 2: the essential professional expertise

The most commonly discussed requirement in terms of graduate qualities was that of professionalism in terms of professional values and underpinning professional expertise. This was emphasised by both industry and research collaborators. These qualities were difficult to explain in clear terms. It is arguably a largely tacit quality that develops through experience as elaborated by [13] in terms of innovation in engineering work. Accordingly, the general consensus among collaborators was the need for a broad variety of problem-solving exercises. These did not necessarily need to be computer-based or part of a project. Also, a significant portion of individual work, as opposed to group projects, was considered essential to ensure that individuals acquire the necessary knowledge and expertise. A range of behavioural qualities were identified, many of which would be expected by professionals in general, such as “…taking responsibility for project work…” (industry collaborator 5), “…able to do time planning… and use of Gantt charts…produce parts lists … and work with budgets” (Research collaborator 3).

Professional qualities and knowledge bases were often expressed in relation to potential difficulties arising from too specific a focus within the programme, as the following contributions suggest:

I would very much like to see embedded-systems and lower level courses remain a big part of the ECE curriculum at UCT. While virtualization, virtual machines and dynamic languages are a big part of the industry landscape, a solid grounding in bits-and-bytes and the low-level functioning of a computer is still extremely important. All the abstractions we have built are leaky, and when those abstractions leak, a knowledge of low-level computing topics is indispensable (Industry collaborator 3).

It was furthermore emphasised that educational experiences within courses be appropriately focused. For instance, avoiding side-issues that may be interesting but that would involve knowledge that has a short ‘shelf-life’, as is expressed in the following comment:

The lifecycle of technologies like these have proven to be short. While exposing students to some of the ideas in these languages is important (LINQ, for example, is a fascinating technology), learning the details should not be the focus (Research collaborator 2).

The collaborators had a range of advice to offer with regard to group work. This involved some heated discussion, for which there was no clear consensus among the collaborators. In reflection on the 2009 curriculum that was used as a basis of comparison, some collaborators wanted less group work, to ensure that individuals gained experience in applying theories and tools; but others wanted more group work so that students were better prepared for working in development teams, as is typically the case in computer engineering workplaces. The comments below summarize some opinions on this:

…groupwork should be limited because there are too many ways that students can cheat… ending up with students not having grasped the way to use essential tools and development methods (Research collaborator 3).

…it is imperative that version control systems are covered… (Industry collaborator 4).

… the leadership quality is an important factor…which can be gained in appropriately structure group assignments (Industry collaborator 5).

Developing what was referred to as ‘quality’ designs (or design artefacts) was highlighted in the discussion. The definition of ‘quality’ was generally expressed vaguely, encompassing a range of unspoken expectations that professionals are sensitive to. Testing and the need to understand and construct effective testing frameworks were emphasised, as illustrated by the quote below:

Not enough emphasis is on testing and frameworks associated with testing, i.e. using an existing framework… or setting up… testing frameworks. (Industry collaborator 4)

4.3 Finding 3: engineering problem-solving in context

Based on collaborators’ inputs, it was clear that the curriculum needed to incorporate certain types of application contexts – or simulated work conditions – together with the application of tools and technologies related to solving problems within these contexts. It was emphasized that the application context and problems concerned be of a sufficiently complex nature. For example, “… simple PWM … and squeaking speakers are way too simple for third years” (Industry collaborator 4); “… they should
have exposure to signal processing techniques... and coming up with their own algorithms” (Research collaborator 3). The discussion and correspondence of contexts and problems were broad; some collaborators suggesting “… simple radar receiver processing using simulated data…” (Research collaborator 5), others suggesting “… sonar applications can be done using low cost parts…” (Research collaborator 4) and “…calculating locus and movement affects with… robot control programs” (Research collaborator 6).

While it there was consensus about gaining problem-solving experiences for complex applications, there was not consensus on specific contexts; but there was nevertheless a clear agreement in terms of the need for understanding problem solving for real-time applications and exposure to solving problems using real-time operating systems – as exemplified by the following contribution:

Based on my experience… it would be beneficial for the student to get more practical experience with Real-Time.... Other than that, I am of the opinion that ECE is an excellent degree and successfully equips the student… (Industry collaborator 6).

A need for some inter-disciplinary learning was identified, for which elective courses was proposed as the recommended approach to avoid causing core courses to lose their focus on major issues, as illustrated by the comment below:

… with complex problem-solving applications as well as inter-disciplinary applications… such as medical imaging and 3D modelling where many computer engineers find work… can be placed in an elective course… (Industry collaborator 2)

4.4 Finding 4: implementation specific techniques and tools

In the course of discussions, emails, etc. a cluster of specific tools and techniques were identified as being important. It is important to note that the industry and research partners did not advocate the need for very specific training, and often qualified their comment by including or suggesting related tools and techniques, or emphasising the importance of understanding how to use the tool effectively, such as the industry collaborator who commented that “Linux scripting … is an extremely useful tool if you know how to use it!” (Industry collaborator 1).

Thus even in learning the application of specific tools and techniques, it is important that students build up and draw on their reservoirs[14] of problem-solving, engineering expertise and fundamental knowledge, as expressed below:

I can’t stress how important it is to be able to extract information from databases and understand the correct standards for database layout… it is important especially if you wish for your product to allow for integration that your database is structured logically and efficiently. (Industry collaborator 3)

Taken together, the different collaborators used a wide variety of implementation techniques and tools in their respective workplaces. However, there were a number of tools and programming languages that were consistently mentioned or suggested during meetings and in correspondence; these included the following (these are listed approximately in the order of how frequently the term was used starting with the most frequently mentioned first): C, C++, GCC, Linux, Verilog, Python, VHDL, Xilinx ISE, Embedded Linux, Altera Quartus, Java/Embedded Java, Android, Visual Computer / Virtual Box, Linux Drivers, MySQL, MyHDL, Ubuntu, .NET, Win32/Win64, Microsoft Visual Studio, VX Works, Windows Embedded, SQLite, C#, Boost libraries, EDA hybrids, C2gates.

The collaborators presented various scenarios and case studies drawn on their own on-going projects and experiences when reflecting on specific tools and techniques that students may benefit from learning, some extracts from correspondence include:

Our products are built with java and... a tomcat server. We use mysql... and hibernate to communicate between java and the databases… (Industry collaborator 4).

Java [suggested] for its ability to run on multiple architectures (Industry collaborator 3).

... I am currently working on developing Firmware (only VHDL and a little C code) for radars. We deal primarily in coding with Aldech Active-HDL for block diagram implementation and simulation of VHDL code... use Altera Quartus for synthesis... (Industry collaborator 3).
5 KNOWLEDGE MODEL FOR RESPONSIVE GRADUATES

A visual modelling tool was constructed during the data analysis to assist with decision making on the emerging knowledge categories. The model developed involved four quadrants, corresponding to the four main knowledge areas identified in the findings. The model was used as a visual aid during follow-up discussion with collaborators, and with faculty involved in teaching ECE courses. The model is operationalized by placing a curriculum request at a point on the x-y plane where the x- and y-scales indicate the following: the horizontal position ranges from low situated knowledge (i.e. simple context or the need for little or no contextual structure) on the left, and high situated knowledge (i.e., complex contexts needing much explanation) on the right. The vertical position ranges from requests that need little in-depth disciplinary knowledge (e.g. simple mathematics or coding) at the bottom, to requests that require deep disciplinary knowledge at the top of the plane.

![Graduate knowledge model](image)

Fig. 1: Graduate knowledge model to assist with explaining characteristics of a curriculum requests.

6 CONCLUSIONS

The results of the data analysis identified the main graduate knowledge areas to which the curriculum requests corresponded, together with the collaborators recommendations on how these knowledge areas should – and in some cases should not – be influenced during the curriculum renewal process. These main knowledge areas were the: ‘pure disciplinary knowledge’, ‘professional expertise’, ‘context-based problem solving’, and ‘implementation techniques’. Based on the findings, a general consensus emerged around the need to protect the disciplinary knowledge – in so much that changes made to the curriculum and selection of different tools for practical experiences should not significantly impact the fundamental disciplinary knowledge that is learned by the student in completing the degree. Characteristics for professional behaviour and the need to achieving appropriate professional qualities was emphasised, and specified as a component of the curriculum that is more important than mastering the latest tools and should therefore not be adversely affected during the curriculum renewal process.

In Section 5 a model was presented that illustrates the differences between the main knowledge areas identified. The parts of the curriculum are interlinked, for example professional expertise is often learned through project work; thus individual learning activities cannot necessarily be isolated
according to these different knowledge areas. Generally, in structuring a curriculum, students experience transitions in their learning experiences, moving from learning theory (e.g., in lecturers) towards experimenting with implementation techniques in the laboratory. Fig. 2 shown below is an adaptation of the graduate knowledge model given in Fig. 1. This version of the model is annotated to include curriculum planning considerations.

![Graduate knowledge model annotated to emphasise a responsible curriculum.](image)

The upper quadrants of the model correspond to ‘responsible areas’, with the closed padlock icon implying the need to conscientiously protect these parts of the curriculum to ensure graduates develop the fundamental knowledge and professional expertise required in the discipline. The lower quadrants of the model correspond to ‘responsive areas’, for which the open padlock implies more flexibility and the potential to accommodate changes on a regular basis to respond to dynamic trends and changes in the popularity of tools as used in the profession. The central arrows with dashed lines imply the need for supporting the development of professional expertise with the other forms of knowledge that is learned during in the programme. The shaded arrows around the periphery suggest the need to consider alignment between sections of the curriculum, such as pure disciplinary knowledge leading the way towards implementation techniques, opening the way towards more complex used of application-based problem solving, culminating in building professional expertise. The arrows suggest a cycle where students develop increasingly more complex knowledge as the curriculum progresses.

6.1 General reflections

The findings of the collaborative process show that both the industry and the research collaborators want graduates who have primarily acquired a broad spectrum of skills and knowledge that spans a range of applications and implementation techniques. The collaborators confirmed the need for graduates to have a good depth of fundamental knowledge, both fundamental knowledge in terms of the specific computer engineering discipline, as well as adopting values and behaviour broadly associated to professional engineers. The students learning experiences should furthermore span a breadth of contextual and practical situations appropriate to application development trends.

The term ‘T-Shaped people’ [1] provides a useful description of the collaborators’ expectations for graduates the down-stoke of the ‘T’ representing a graduate’s depth of disciplinary knowledge and the cross-stroke representing breadth of knowledge and flexibility in managing problems in different situations and contexts.

Preparing ‘T-shaped’ graduates is crucial for computer engineers considering how rapidly the landscape of computer architectures and technologies is changing [6], combined with the way
professional computer engineers often move between significantly different applications (e.g., from factory line control systems to medical systems). Accordingly, students should likewise be exposed to solving problems in a variety of application domains to prepare them for professional practice. An effective computer engineering curriculum consequently needs to incorporate a variety of hybrid knowledge forms, which combines situated and more tacit knowledge of profession practice with the explicit knowledge of pure and applied disciplines associated with computer engineering – added with the more interdisciplinary application knowledge that professionals draw on to establish effective solutions for different applications. In order to provision students with this variety of skills and experiences, ECE programme at UCT has been adapted to incorporate more accurate project-based simulation of design work based on recommendations from the industry and research collaborators discussed in this paper.

REFERENCES


