Global Journal of Computer Science and Technology
Cloud & Distributed
Volume 12 Issue 11 Version 1.0 Year 2012
Type: Double Blind Peer Reviewed International Research Journal
Publisher: Global Journals Inc. (USA)
Online ISSN: 0975-4172 & Print ISSN: 0975-4350

Missing Elements of Computer Science Curricula 2013

By Muhammad Anwar- ur-Rehman Pasha & Shaheen Pasha
University of Sargodha, Sargodha, Pakistan

Abstract - Rapidly expanding computing domain has forced educational institutions to up-grade existing curricula of computing degree programs. Recently, a joint task force of Association for Computing Machinery and IEEE-Computer Society has published the Strawman Draft of Computer Science Curricula 2013. The Draft has introduced some new ideas to keep computing curricula modern and relevant. The recommended curricula have designed in the light of 6% response rate of the conducted survey. This paper has pointed out some important aspects which need attention to meet the challenges of the 21st century. These aspects include an Ad-hoc approach towards the core body of knowledge, incomplete curriculum guidelines, over-ambitious contents and learning outcomes. Some other missing aspects include computing dispositions, global education, 21st century skills, guideline for inclusion and the hidden curriculum. It is believed the recommendations of this paper may generate some thought provoking ideas to make the computing curricula more robust and effective.

Keywords : Computing Curriculum, Computer Science Curricula 2013, Computing Model Curriculum, Dispositions in Computing, Hidden Curriculum in Computing, Global Education in Computing, Flexible Computing Curriculum.

GJCST-B Classification: K.3.2

MISSING ELEMENTS OF COMPUTER SCIENCE CURRICULA 2013

Strictly as per the compliance and regulations of:

Missing Elements of Computer Science Curricula 2013

Muhammad Anwar-ur-Rehman Pasha [α] & Shaheen Pasha [σ]

Abstract - Rapidly expanding computing domain has forced educational institutions to up-grade existing curricula of computing degree programs. Recently, a joint task force of Association for Computing Machinery and IEEE-Computer Society has published the Strawman Draft of Computer Science Curricula 2013. The Draft has introduced some new ideas to keep computing curricula modern and relevant. The recommended curricula have designed in the light of 6% response rate of the conducted survey. This paper has pointed out some important aspects which need attention to meet the challenges of the 21st century. These aspects include an Ad-hoc approach towards the core body of knowledge, incomplete curriculum guidelines, over-ambitious contents and learning outcomes. Some other missing aspects include computing dispositions, global education, 21st century skills, guideline for inclusion and the hidden curriculum. It is believed the recommendations of this paper may generate some thought provoking ideas to make the computing curricula more robust and effective.

Keywords : Computing Curriculum, Computer Science Curricula 2013, Computing Model Curriculum, Dispositions in Computing, Hidden Curriculum in Computing, Global Education in Computing, Flexible Computing Curriculum.

I. Introduction

Traditionally, computing is used as an umbrella term to represent the following five disciplines:

1. *Computer Engineering* (*CE*) focuses on computing hardware and associated computing aspects.
2. *Computer Science (CS)* focuses on computing theory, methodology, innovation, development (programming) of technologies and applications, and applying computing to new disciplines.
3. *Information Systems (IS)* focuses on applying computing in organizations and organizational information management.
4. *Information Technology (IT)* focuses on solving organizational computing challenges by integrating technologies into solutions and deploying and maintaining the solutions.
5. *Software Engineering (SE)* focuses on developing large complex software systems.

Author α : Department of Computer Science & Information Technology, University of Sargodha, Sargodha, Pakistan. E-mail : marpasha@yahoo.com

Author σ : Division of Education, University of Education, Lahore, Pakistan. E-mail : drshaheenpasha@hotmail.com

Computing is a rapidly progressing domain. In recent years many significant developments have been made and many new concepts have been introduced. For example, "Computational Lens" (Karp, 2011) which articulates a new relationship between computer science and other sciences, "Ternary Computing" dealing with computing for the masses (Li, 2010), "e-Science" managing massive experimental data and collaborating via the Net, "Computational Thinking" (Wing, 2006; 2008), Cloud Computing (Li & Zhang, 2009), Biological Computing (Garfinkel, 2000), etc. In parallel, the integration of computing in other disciplines introduces new disciplines such as "Computational-x" (e.g., computational mathematics, computational physics, computational finance, etc.) and "x-Informatics" (e.g., bio-informatics, dental-informatics, clinical-informatics, etc.) (ACM & IEEE-CS, 2012). Many such developments compel the international community to update the curricula of computing degree programs to meet the needs of the time.

The practice of developing a model curriculum in the computing domain started in 1965 when the Association for Computing Machinery (ACM) for Computer Science curriculum published their recommendations (ACM, 1965). Since then the international community has developed many model curricula to keep computing discipline up-to-date. Recently, the Joint Task Force on Computing Curricula Association for Computing Machinery and IEEE-Computer Society has published the Strawman Draft of Computer Science Curricula 2013 (ACM & IEEE-CS, 2012). The recommendations made in this Draft have introduced some new ideas to keep computing curricula modern and relevant. The Draft has invited suggestions & recommendations from the international community to be included in the Ironman report going to be released in 2013. In this paper we have pointed out some short comings of the recommended curricula and made recommendations to make it more robust and effective. We believe the recommendations made in this paper may generate some thought provoking ideas for developing model curriculum for computing degree programs.

The organization of this paper is as follow. A review of the computing model curriculum development efforts is presented in the next section. Some important aspects of the Strawman Draft are outlined in the next section. Section 4 has identified some shortcomings of

Global Journal of Computer Science and Technology (B) Volume XII Issue XI Version I Year 2012

the curriculum recommended in the Draft. Concluding discussion and recommendations are presented in the last section.

II. Computing Model Curriculum Development

In computing domain, the history of model curriculum development started with the publication of the recommendations of the ACM for Computer Science curriculum (ACM, 1965). Since then many efforts have been made to keep the computing curriculum up-to-date. These efforts include, for example, Curriculum 68 (ACM, 1969), IEEE Computer Society Education Committee/Model Curriculum (1976), Curriculum recommendations for the Undergraduate Program in Computer Science (ACM, 1977) and Curriculum 78 (ACM, 1979), IEEE Computer Society Educational Activities Board/Model Program (IEEE-CS, 1983) and ACM Task Force's Report on the Core of CS (Dening & et al.,1988)

In 1991, ACM and IEEE-CS jointly published Computing Curricula 1991 for Bachelor's degree programs in CS and CE (ACM/IEEE-CS, 1991). In 2001, ACM and IEEE joint task force produced Computing Curricula 2001 (ACM/IEEE-CS, 2001) four distinct disciplines - CS, CE, IS and SE. In 2005, once again ACM & IEEE jointly published the Computing Curricula 2005 (ACM/IEEE-CS, 2005) which included IT as an independent discipline. The interim review effort (2008) (ACM/IEEE-CS, 2008) and the Strawman Draft of CS Curricula 2013 (ACM/IEEE-CS, 2012) are among the latest efforts to keep computing curricula modern and relevant.

III. CS Curricula 2013: The Strawman Draft

The Draft has provided a comprehensive revision of the existing curricula. It is prepared in the light of following guidelines, as reported in (ACM/IEEE-CS, 2012):

- The "Big Tent" view of CS to accommodate the challenges of emerging disciplines include more cross-disciplinary work new programs of the form "Computational Biology," "Computational Engineering," and "Computational X".
- Flexible models for different curricula without losing the essence of a rigorous CS education.
- To identify and describe existing successful courses and curricula to show how relevant knowledge units are addressed and incorporated in actual programs.
- To be applicable in a broad range of geographic and cultural contexts, understanding that curricula exist within specific institutional needs, goals, and resource constraints.

The recommended curricula are based on following ten principles:

1. Computer Science curricula should be designed to provide students with the flexibility to work across many disciplines.
2. Computer Science curricula should be designed to prepare graduates for a variety of professions, attracting the full range of talent to the field.
3. CS2013 should provide guidance for the expected level of mastery of topics by graduates.
4. CS 2013 must provide realistic, adoptable recommendations that provide guidance and flexibility, allowing curricular designs that are innovative and track recent developments in the field.
5. The CS2013 guidelines must be relevant to a variety of institutions.
6. The size of the essential knowledge must be managed.
7. Computer Science curricula should be designed to prepare graduates to succeed in a rapidly changing field.
8. CS2013 should identify the fundamental skills and knowledge that all computer science graduates should possess while providing the greatest flexibility in selecting topics.
9. CS2013 should provide the greatest flexibility in organizing topics into courses and curricula.
10. The development and review of CS2013 must be broadly based.

The Draft has organized the Body of Knowledge into a set of 18 Knowledge Areas: "

1. AL - Algorithms and Complexity
2. AR - Architecture and Organization
3. CN - Computational Science
4. DS - Discrete Structures
5. GV - Graphics and Visual Computing
6. HC - Human-Computer Interaction
7. IAS - Information Assurance and Security
8. IM - Information Management
9. IS - Intelligent Systems
10. NC - Networking and Communications
11. OS - Operating Systems
12. PBD - Platform-based Development
13. PD - Parallel and Distributed Computing
14. PL - Programming Languages
15. SDF - Software Development Fundamentals
16. SE - Software Engineering
17. SF - Systems Fundamentals
18. SP - Social and Professional Issues

Many of these Knowledge Areas are derived from CS curriculum 2001 (ACM/IEEE-CS (2001) and CS curriculum 2008 (ACM/IEEE-CS, 2008) but have been revised—in some cases quite significantly new.

The Draft has introduced three levels of knowledge description: Tier-1 Core, Tier-2 Core, and

Elective. Topics have been identified as either "core" or "elective". The draft suggests that a curriculum should include all topics in the tier-1 core and ensure that all students cover this material. Also, all or almost all topics in the tier-2 core should be taught to all students. It has also been suggested that the curriculum should include significant elective material as covering only the "core" topics is insufficient for a complete curriculum (ACM/IEEE-CS, 2008).

IV. ShortComings of the Cs Curriculum 2013

The Draft is prepared to keep the computing curricula up-to-date and relevant but the following aspects may raise questions about its effectiveness.

a) Low response rate

The Draft reports that "the survey was sent to approximately 1500 Computer Science (and related discipline) Department Chairs and Directors of Undergraduate Studies in the United States and an additional 2000 Department Chairs internationally. We received 201 responses, representing a wide range of institutions". In this case the response rate is just 6% which raises the question of reliability, validity and acceptability of its recommendations. Studies suggest that an achievable and acceptable rate is 75% for interviews and 65% for self-completion postal questionnaires (Arber, 2001; Sitzia & Wood,1998). Similarly, Mundy (2002) comments that "There's no magic figure on response rates. Higher is better: 60% would be marginal, 70% is reasonable, 80% would be good, 90% would be excellent" (p. 25). The recommendations made in the light of 6% response rate can only represent the point of view of a specific community. It cannot be generalized.

b) An Ad-hoc approach towards the core body of knowledge

The Draft has added two new knowledge areas in the core body of knowledge: "Information Assurance and Security" and "Parallel and Distributed Computing" as the survey respondents indicated a strong need of these topics. There is no doubt the identified areas are important but the concept of computing is evolving and expanding with an unprecedented pace. The approach of adding new concepts as they emerge will make the computing core over-crowded and unmanageable.

c) Incomplete curriculum guidelines

The Draft includes guidelines regarding knowledge areas, curricula and course exemplars, institutional challenges, key principles & professional practice, and characteristics of graduates. As a normal practice, an effective curriculum provides guidelines for students' learning, contents for learning, sequence of courses of study, instructional methods and activities, instructional resources, educational settings, evaluation methods for assessing student learning, accountability measures for teaching-learning processes, etc. (Talbot, 2004; HEC, 2012; UNESCO, 2012). Whereas, the recommendations of the Draft covers only few of these aspects.

d) Inconsistency in the use of terms 'Computing' and 'Computer Science'

A substantial amount of research efforts have been carried out to define the distinctive features and characteristics of five key disciplines of computing. In the Draft, the term "computing" and "computer science" are used interchangeably that make it unclear that the proposed recommendations are for 'Computer Science" degree program or for the whole spectrum of computing related degree programs. This aspect is making its scope ambiguous.

e) Over-ambitious contents and learning outcomes

Topics included in the defined knowledge areas can be considered over-ambitious and seems difficult to cover within the proposed time span.

f) Dispositions: an ignored aspect

The concept of dispositions has become an important element of an effective curriculum. It can be thought of as habits of mind or tendencies to respond to certain situations in certain ways. For example, curiosity, friendliness, bossiness, meanness, and creativity are dispositions, rather than of skills or items of knowledge (Katz, 1995). Preparing students for having the disposition to be a programmer is more important than having programming skills. This important aspect is missing from the proposed curriculum.

g) Other missing aspects

Global education, 21st century skills, inclusive education, and hidden curriculum are among the important aspects of 21st century education. These aspects have not been addressed in the Draft.

V. Discussion & Recommendations

Computing is a rapidly changing domain and will continue to change for the foreseeable future. Both institutions and faculty are striving to address how to meet the needs of the students studying in computing and other newly emerging disciplines as they are being considered responsible of producing well-rounded computing graduates equipped with professional competencies ready to work in a more holistic way than simply demonstrating technical skills. For this purpose they need a flexible curriculum model that would take a broader view of the field and provides guidelines to meet the challenges of 21st century education. The ACM and IEEE-CS joint task force's effort of producing the Straman draft of Computer Science Curricula 2013 is a valuable attempt in this direction. Yet below discussed

Global Journal of Computer Science and Technology (B) Volume XII Issue XI Version I Year 2012

aspects need to be considered before producing a final draft.

As discussed earlier, the Draft has increased the size of the core body of knowledge by adding new knowledge areas. In recent years many new concepts have been introduced and will continue in the foreseeable future. The approach of adding new knowledge areas in the computing core will make it unmanageable if new knowledge areas continue to emerge. The wisdom suggests that in place of increasing the size of the core, a more appropriate approach has to be adopted for accommodating new ways of thinking, application and evolution of computing. We believe, in place of increasing the size of the computing core, some common knowledge areas should be identified which could strengthen students' conceptual understanding required to study higher level computing concepts. These common knowledge areas should be equally important for both the students of core computing disciplines and the students studying in newly emerged fields. In this regard we recommend that the computing core should be based on following knowledge areas which are essential for a whole range of computing degree programs including "computational-x" and "x- informatics". These knowledge areas are:

1. Principles of Computing & Programming
2. Principles of Operating Systems
3. Principles of Database Systems
4. Principles of Software Engineering
5. Principles of Human Computer Interaction
6. Principles of Web Technologies.

Keeping a small core will allow institutions to include newly emerging areas like quantum computing, bilogical, cloud computing, etc. It will also allow them to produce their own brands through offering special topics or training. Branding in higher education is a topic of great interest among the higher education community (Brunzel, 2007; Lockwood & Hadd, 2007); Temple, 2006). We also propose the following curriculum structure for computing degree programs:

- Core Compulsory Courses (17%)
- Foundation Elective Courses (11%)
- Interdisciplinary Computing Supportive Elective Courses (11%)
- General Education Elective Courses (9%)
- Domain Specific Elective Courses (38%)
- Specialization/Major Elective Courses (9%)
- Capstone Project/Internship (5%)

For the selection of course contents "Selective Abandonment" strategy (Lovely & Smith, 2004) is strongly recommended as it allows teachers to prioritize the content of instructional material into three categories: essential material must be covered and have top priority, supportive may be dealt with in conjunction with other material or as a cooperative or independent learning experience, and extraneous material can be included as time allows.

It could be argued that we have eliminated the traditional core areas like computer programming, data structure and algorithms, data-communication, digital logic design and computer organization, etc. We believe these subjects have different standpoints in different domains. For example, low level computer programing is more useful for computer engineering students as compared to the students of information systems. Time has come to realize that to develop an appropriate mindset the students need to study material related to that particular domain (Pasha & Pasha, 2012). Such topics could be covered under the category of 'Domain Specific Elective Courses'. This way institutions can offer different contents to the students of different degree programs. Similarly, courses like discreet structures, data-communication, digital logic design and computer organization could be offered under 'Computing Supporting Elective Courses'. Science, Mathematics, etc. could be covered under 'Interdisciplinary supporting Elective Course'. Course like Philosophy, Psychology, Sociology, Comparative Study of Religions, etc. could be taught under 'General Education Electives'. The Capstone project will allow students to demonstrate the knowledge and skills they have learnt during the course of their study.

Jackson (2008) argues that higher education has a responsibility to help students to develop and promote their understanding and awareness of their own creativities, identity and lifelong learning experiences. He further comments "Preparing students for a lifetime of working, learning and living in uncertain and unpredictable worlds that have yet to revealed is perhaps one of the greatest responsibilities and challenges confronting universities all over the world." Katz (1993) argues that "One of the major questions to be addressed when developing a curriculum is, What should be learned?" One way to answer this question, as (Katz, 1991) explains, "is to adopt at least four types of learning goals, those related to knowledge, skills, dispositions, and feelings. The acquisition of both knowledge and skills is taken for granted as an educational goal, and most educators would also readily agree that many feelings (e.g., self-esteem) are also influenced by school experiences and are thus worthy of inclusion among learning goals. However, dispositions are seldom included, although they are often implied by the inclusion of attitudes (e.g., attitudes toward learning) as goals" (Katz, 1993).

The role of dispositions in computing education is very important. For example, having the disposition to be a programmer is much better that just having programming skills. Similarly, and, having the disposition to be a software engineer is much batter than just having software engineering skills. Katz (1995) pointed out that "Dispositions are not learned through formal instruction or exhortation. Many important dispositions are in-born in all children like the dispositions to learn and to make sense of experience." Many dispositions that most adults want children to acquire or to strengthen - for example, curiosity, creativity, cooperation, openness, friendliness—are learned primarily from being around people who exhibit them; they are strengthened by being used effectively and by being appreciated rather than rewarded (Kohn, 1993).

To strengthen the dispositions computing students should have, they must be provided with the opportunity to express the dispositions in their behavior. When manifestations of the dispositions occur, they can be strengthened as the students observe their effectiveness and the responses to them and experiences satisfaction from them. Dweck (1991) argue that an effective curriculum can strengthen certain dispositions by setting learning goals rather than asking teachers to set some performance goals. Therefore, it is strongly recommended that the forthcoming Iransman Draft must identify those dispositions which are essential for computing students and make part of the curriculum.

Hidden Curriculum is an important component of any educational program (Jackson, 1968). Hidden curriculum deals with the elements of socialization embedded in the curriculum and are imparted to students through daily routines, curricular content, and social relationships, yet are not part of the formal curricular content. Emile Durkheim views educational systems reflect underlying changes in society because the systems are a construct built by society, which naturally seeks to reproduce its collectively held values, beliefs, norms, and conditions through its institutions (Giddens, 1972). He further comments, "Society can survive only if there exists among its members a significant degree of homogeneity; education perpetuates and reinforces this homogeneity by fixing in the child, from the beginning, the essential similarities collective life demands". He also comments that socializing children to hold particular values such as those of "achievement" and "equality of opportunity" is necessary to this consensus and is the primary function of education (Giddens, 1972).

The Draft has addressed the issue of professional practices and considers it as a discrete area which has to be treated explicitly. We believe topics like professional ethics, soft skills, public speaking, critical thinking & reasoning, modern literacies, inter-personal attributes, entrepreneurship, attitude towards lifelong learning, other life & social skills should not be considered discrete items and to be taught independently. Such concepts should be threaded into the entire fabric of the curriculum and taught as a hidden curriculum. This approach will, on the one hand, make room for other valuable concepts. On the other hand, it will make students responsible citizen, ethically sound professionals, and sociable members of the society.

The biggest pitfall in selecting the contents and learning outcomes for any learning activity is to be over-ambitious for the time allocated. The over-ambitious contents and learning outcomes is another aspect of the Draft which must be addressed. Let's take the example of "Algorithms and Complexity (AL)" knowledge area. The Draft has proposed the following contents, learning outcomes and number of hours.

a) *AL/Basic Analysis [2 Core-Tier1 hours, 2 Core-Tier 2 hours]*

i. *Topics : [Core-Tier1]*

- Differences among best, average, and worst case behaviors of an algorithm
- Asymptotic analysis of upper and average complexity bounds
- Big O notation: formal definition
- Complexity classes, such as constant, logarithmic, linear, quadratic, and exponential
- Empirical measurements of performance
- Time and space trade-offs in algorithms

ii. *[Core-Tier2]*

- Big O notation: use
- Little o, big omega and big theta notation
- Recurrence relations and analysis of recursive algorithms
- Some version of a Master Theorem

iii. *Learning Outcomes*

1. Explain what is meant by "best", "average", and "worst" case behavior of an algorithm. [Knowledge]
2. In the context of specific algorithms, identify the characteristics of data and/or other conditions or assumptions that lead to different behaviors. [Evaluation]
3. Determine informally the time and space complexity of simple algorithms. [Application]
4. Understand the formal definition of big O. [Knowledge]
5. List and contrast standard complexity classes. [Knowledge]
6. Perform empirical studies to validate hypotheses about runtime stemming from mathematical analysis. Run algorithms on input of various sizes and compare performance. [Evaluation]
7. Give examples that illustrate time-space trade-offs of algorithms. [Knowledge]
8. Use big O notation formally to give asymptotic upper bounds on time and space complexity of algorithms. [Application]

Global Journal of Computer Science and Technology (B) Volume XII Issue XI Version I Year 2012

9. Use big O notation formally to give average case bounds on time complexity of algorithms. [Application]
10. Explain the use of big omega, big theta, and little o notation to describe the amount of work done by an algorithm. [Knowledge]
11. Use recurrence relations to determine the time complexity of recursively defined algorithms. [Application]
12. Solve elementary recurrence relations, e.g., using some forms of a Master Theorem. [Application]

Teaching of the above mentioned course contents and expecting the mentioned learning outcomes from students in just 4 hours seem unrealistic. We believe the proposed learning outcomes require more time on the part of both teachers and students for their completion than is mentioned. Knight (2002) argues that in the higher education contents should be offered in order to maximize the chance that learners will experience coherence, progression and deep learning. If the contents and outcomes are over-ambitious compare to the time available, these cannot go without compromising the essential characteristics of the learning experience (Barnett, et al., 2001; Pasha & Pasha, 2012a). Di Carlo (2009) argues that attempting just to cover the overcrowded course contents limit students to simply learning facts without developing the ability to apply their knowledge to solve novel problems. It puts an extra cognitive load on students (Chandler & Sweller, 1991). and makes both faculty and students overburdened (Gibbs, 1981; Ironside, 2004). As a result, the students' academic achievements get effected (Apple, 2001; Jones, 2008). For an effective learning students need to be engaged in higher order cognitive activities which are related to the upper half of Bloom's taxonomy (Bloom & David, 1959; Pasha & Pasha, 2012a).

The high pace of knowledge exploration, inventions of new technologies, and the convergence of computing and other disciplines, the emergence of new domains & disciplines have introduced new challenges to curriculum development for degree programs. These trends demand a flexible approach for curriculum development which not only meets the existing challenges but also have the potential to accommodate the future needs as well (Pasha & Pasha, 2012a).

We need to realize that the 21st century has been labeled as an era of knowledge economies which have manifested itself in many different ways like science and technology bonding has become stronger than ever before, innovation has become more important for economic growth and competitiveness, continuing education and lifelong learning have got unprecedented importance in organizational practices, investment in intangible assets has become more valuable than investments in fixed capital (Pasha & Pasha, 2012b). These trends have led to an increased competition in the business world (Utz, 2006). Also the relationship between knowledge and technology has become more evident. Although, the economic activities all over the world are increasingly becoming knowledge oriented but the degree of knowledge and technology integration into economic activity is now so great that knowledge & technology have been recognized as the drivers of productivity and economic growth (Kogut & Zander, 1992; Nonaka, & Takeuchi, 2002; Choo, 2002; Zítek & Klímová, 2011). In today's world, the basic economic resource - the means of production - is no longer capital, nor natural resources, nor labor. It is and will be the knowledge workers who possess high levels of education and/or expertise in a particular area, and who use their cognitive skills to engage in complex problem solving. Such knowledge workers will be the assets of the organization (Drucker, 2006). In this sense transforming computing students into valuable knowledge workers should be one of the key purposes of a curriculum (Pasha & Pasha, 2012c).

Time has come to realize the changing patterns of 21st century universities education which have removed the identity of place, the identity of time, the identity of the scholarly community, and the identity of the student community. For accommodating these changes, we need to understand the five contemporary competing epistemological pressures on the higher education curriculum. Brigges (2000) suggests that the future of the higher education curriculum will hang significantly on the way in which this competition is resolved:

1. The deconstruction of the subject, as reflected in, for example, the modularization of the curriculum;
2. The cross-curricular `key' skills movement;
3. The learning through experience movement and the shift of the seat of learning outside the academy;
4. The anarchic potential of web-based learning; and
5. The reaffirmation of the subject as the academic and organizational identity.

We believed, similar to other disciplines, people from computing domain must appreciate these challenging aspects and find practical ways to resolve these conflicts. We also believe giving considerations to the following aspects would make computing curricula more agile, responsive and accommodating:

The curriculum should:

- Equip students with 21st century skills;
- Include a hidden curriculum for teaching the elements of socialization & other life skills;
- Include the aspects of Global Education & Multicultural education;
- Promote inclusive education and define measures to meet the needs of the students with special needs;

- Allow institutions to integrate the concept of branding within their degree programs.

We believe that the recommendations made in this paper may provide some useful ideas to be included in the Ironman Draft which is going to be released in 2013 [6].

VI. Acknowledgment

We express our sincere and deep gratitude to our colleagues and professionals from teaching and software Industry who made it possible to streamline our thoughts for this research. We also pay our thanks to our institutions for providing the opportunity to conduct this research.

References Références Referencias

1. ACM, 1965. 'An Undergraduate Program in Computer Science-Preliminary Recommendations'. *Communications of the ACM*, 8(9), 543–552.
2. ACM, 1969. 'Curriculum 68: Curriculum Committee on CS Recommendations for Academic Programs in Computer Science'. *Communications of the ACM*, 11(3), 151–197.
3. ACM Curriculum Committee on CS, 1977. 'Curriculum recommendations for the Undergraduate Program in Computer Science', *SIGCSE Bulletin*, 2, 1-16.
4. ACM Curriculum Committee on CS, 1979. 'Curriculum 78: Recommendations for the Undergraduate Program in CS'. *Communications of the ACM*, 22 (3), 147-166.
5. ACM/IEEE-CS, 1991. *Computing Curricula 1991: Recommendations of ACM & IEEE-CS Joint Curriculum Task Force*, ACM Press, Baltimore, MD.
6. ACM/IEEE-CS, 2001. *Computing Curricula 2001: Recommendations of ACM & IEEE Curriculum Task Force on Computer Science*, IEEE Computer Society Press and ACM Press.
7. ACM/IEEE-CS, 2005. *Computing Curricula 2005: The Overview Report*, IEEE and ACM Press.
8. ACM/IEEE-CS, 2008. *Computer Science Curriculum 2008: An* Interim *Revision of CS 2001*, ACM/IEEE-CS Joint Interim Review Task Force Report, ACM Press.
9. ACM/IEEE-CS, 2012. *Strawman Draft: Computer Science Curricula 2013*. The Joint Task Force on Computing Curricula Association for Computing Machinery and IEEE-Computer Society.
10. Apple, M. W., 2001. *Educating the "Right" Way: Markets, Standards, God, and Inequality*. New York: Rout ledge Falmer.
11. Arber, S., 2001. Designing Samples. In: N. Gilbert (Ed.) *Researching Social Life*. London: SAGE Publications.
12. Barnett, R., Parry, G. & Coate, K., 2001. 'Conceptualising Curriculum Change', *Teaching in Higher Education*, 6(4), 436-449.
13. Bloom, B. S. & David, R. K., 1959. Taxonomy of Educational Objectives: The Classification of Educational Goals, by a committee of college and university examiners. In: *Handbook I: Cognitive Domain*, New York, Longmans, 1959.
14. Bridges, D., 2000. 'Back to the Future: the higher education curriculum in the 21st century'. Cambridge Journal of Education, 30(1), 37-55.
15. Brunzel, D. L., 2007. 'Universities Sell Their Brands'. Journal of Product & Brand Management, 16(2), 152-3.
16. Chandler, P. & Sweller, J., 1991. 'Cognitive Load Theory and the Format of Instruction'. Cognition and Instruction, 8, 293-332.
17. Choo, C. W., 2002.The Strategic Management of Intellectual Capital and Organizational Knowledge, Oxford University Press, New York, NY.
18. Denning, P. J. & et al., 1988. Report on the ACM Task Force on the Core of Computer Science. New York: ACM Press.
19. DiCarlo, S. E., 2009. 'Too much content, not enough thinking, and too little FUN!'. Advances in Physiology Education, 33(4), 257-264.
20. Drucker, P. F., 2006. Classic Drucker, Boston, MA. Harvard Business School Publishing Corporation.
21. Dweck, C. S., 1991. Self-theories and goals: Their role in motivation, personality, and development. In: Richard A. Dienstbier (Ed.), Nebraska symposium on motivation: Vol. 38. Lincoln: University of Nebraska Press. pp. 199-235.
22. Garfinkel, S. L., 2000. 'Biological Computing'. Technology Review, May/June, 71-77.
23. Gibbs, G., 1981. Twenty terrible reasons for lecturing, SCED Occasional Paper No. 8, Birmingham.
24. Giddens, A. ed., 1972. Emile Durkheim Selected Writings, Cambridge, England: Cambridge University Press.
25. HEC, 2012. National Reports on the Undergraduate Curriculum, Traditional and Contemporary Perspectives - Innovations in the Undergraduate Curriculum, Higher Education Curriculum, Available at: <http://education.stateuniversity.com/pages/1896/Curriculum-Higher-Education.html> [Accessed 28 June 2012].
26. IEEE-CS, 1976. A Curriculum in Computer Science and Engineering. IEEE Computer Society Education Committee/Model Curriculum Subcommittee Report, IEEE Computer Society.
27. IEEE-CS, 1983. Model Program in Computer Science and Engineering, IEEE Computer Society Educational Activities Board/Model Program Committee Report, IEEE Computer Society.

28. Ironside, P. M., 2004. '"Covering Content" and Teaching Thinking: Deconstructing the Additive Curriculum', Journal of Nursing Education, 43(1), 5-12.
29. Jackson, N., 2008. Tackling the Wicked Problem of Creativity in Higher Education. In: International Conference Brisbane at the ARC Centre for the Creative Industries and Innovation.
30. Jackson, P., 1968. Life in Classrooms, New York: Holt, Rinehart & Winston.
31. Jones, R. D., 2008. Student Engagement — Creating a Culture of Academic Achievement, International Center for Leadership in Education.
32. Karp, R., 2011. 'Understanding Science through the Computational Lens'. Journal of Computer Science and Technology, 26(4), 569-577.
33. Katz, L.G., 1991. Pedagogical Issues in Early Childhood Education. In: S. L. Kagan, (Ed.). The Care and Education of America's Young Children: Obstacles and Opportunities. Ninetieth Yearbook of the National Society for the Study of Education. Part I. Chicago: University of Chicago Press. pp. 50-68.
34. Katz, L. G., 1993. 'Dispositions as Educational Goals'. ERIC Digest.
35. Katz, L. G., 1995. Dispositions in early childhood education. In: L. G. Katz (Ed.), Talks with teachers of young children. Norwood, N: Ablex. pp. 380-232.
36. Knight, P. T., 2002. Being a Teacher in Higher Education, Buckingham: SRHE/OU Press.
37. Kogut, B. & Zander, U., 1992. 'Knowledge of the firm, combinative capabilities, and the replication of technology'. Organization Science, 3, 383-97.
38. Kohn, A., 1993. Punished by rewards: The trouble with gold stars, incentive plans, A's, praise, and other bribes. Boston, MA: Houghton Mifflin.
39. Li, D. & Zhang, H., 2009. 'Cloud Computing Beyond Turing Machine'. Communications of the China Computer Federation, 5(12), 8-16.
40. Li, G. (Ed.), 2010. Information Science and Technology in China: A Roadmap to 2050, Science Press Beijing and Springer-Verlag, Berlin.
41. Lockwood, R. C. & Hadd, J., 2007. 'Building a Brand in Higher Education: Why Business Practice-Particularly Brand Strategies – are Becoming Essential in Today's Universities'. Gallup Management Journal Online, 12, 1-6.
42. Lovely, S. & Smith, S., 2004. 'Selective abandonment: How and when to say no'. Principal Leadership, 5(3), 35-38.
43. Margolis, E., (2001). The Hidden Curriculum in Higher Education, Routledge.
44. Mundy, D., 2002. 'A Question of Response Rate', Science Editor, 25(1), 25.
45. Nonaka, I. & Takeuchi, H., 2002. The Knowledge-Creating Company. New York: Oxford University Press.
46. Pasha, M. A. & Pasha, S., 2012a. 'Rethinking of Computing Curricula in Higher Education in Pakistan'. International Journal of Advanced Research in Computer Science, 3(3), 111-115.
47. Pasha, M. A. & Pasha, S., 2012b. Essentials of Knowledge Management: Concepts, Theories and Practices. Innovators Knowledge Services, Lahore, Pakistan. Available at: < http://www. innovators. edu.pk/?q=node/139> [Accessed 28 June 2012]
48. Pasha, M. A. & Pasha, S., 2012c. A Pragmatic Approach for Implementing Knowledge Management in Pakistani Organizations Using Open Source Technologies, International Journal of Computer Application (Submitted for publication May 2012).
49. Sitzia, J. & Wood, N., 1998. 'Response Rate in Patient Satisfaction Research: An Analysis of 210 Published Studies'. International Journal for Quality in Health Care, 10: 311–317.
50. Talbot, M., 2004. 'Monkey See, Monkey Do: A Critique of the Competency Model in Graduate Medical Education', Medical Education, 38: 587-592.
51. Temple, P., 2006. 'Branding higher education: illusion or reality?'. Perspectives: Policy and Practice in Higher Education, 10(1), 15-19.
52. UNESCO, 2012. Graduate Employability in Asia, Asia and Pacific Regional Bureau for Education, UNESCO Bangkok.
53. Utz, A., 2006. Fostering Innovation, Productivity, and Technological Change: Tanzania in the Knowledge Economy. World Bank Institute, the International Bank for Reconstruction and Development/the World Bank.
54. Wing, J., 2006. 'Computational Thinking'. Communications of the ACM, 49(3), 33-35.
55. Wing, J., 2008. 'Computational Thinking and Thinking about Computing'. Philosophical Transactions of the Royal Society, 366(1881), 3717-3725.
56. Zítek, V. & Klímová, V., 2011. Knowledge Economy and Knowledge Infrastructure, Proceedings of International Conference on Applied Economics, pp. 821-928.