Enhancing Computational Thinking in Information Science Engineering Education

In general, computational thinking is represented by a set of problem-solving strategies that includes the methods of expressing problems and the procedures leading to their solutions in a systematic way. In particular, with reference to Information and Communication Technologies, it involves the mental skills and techniques for designing algorithms that get computers do jobs for us, as well as the methodologies for explaining and interpreting the world as a complex of information processes.

Especially for the engineering community, computational thinking represents a key feature and a fundamental capability. Moreover, in this practitioner's area, the term should include the set of skills needed to transform real-life challenges into problems, which can be solved with the help of a computer, and to apply computer-based solutions to questions at hand. On one hand, this mindset is fundamental to almost every engineering task. On the other hand, teaching computational thinking represents the main challenge for engineer educators worldwide.

Therefore, when designing and implementing engineering courses and curricula, one has to consider an increasing number of skills and tasks. Information and Communication Technologies progress requires analysis, design, and creation of increasingly large and complex systems. Engineering working environments and tasks require interdisciplinary and multidisciplinary teams, as well as close communication between involved groups.

It is worth noting that the design of engineering courses and curricula must take these factors into careful consideration, whist the necessary skills must be implicitly or explicitly developed. Moreover, proper teaching techniques can enable students to develop or increase computational thinking through systematic introduction of simple and advanced computational tools, depending on the courses in the engineering curricula (Bachelor or Master Science degrees). In this way, the definition of suitable relationships among the courses in the engineering curricula enhances the application of the aforementioned principles to engineering courses themselves.

In particular for the Automatic Control area of Information Science engineering, simple strategies can be exploited for fostering computational thinking in education. Different teaching approaches can be used with reference to basic and elective courses of Automatic Control for BSc and MSc degrees. In particular, through the learned lessons and the experience acquired after years of teaching classes, the learning effectiveness can be enhance by exploiting three main tools.

The teaching activity must be supported by *(i)* a 'learning by doing' approach, which enhances the development of theoretical and practical skills proposed to the student at the same time. On the other hand, the use of *(ii)* 'real and realistic examples' taken from different engineering backgrounds helps to engage students and attract their interest towards difficult theoretical activities. Moreover, the design and the development of proper *(iii)* 'manual and semi-automated procedures' that are tailored to the considered application examples drive the students to learn the most appropriate engineering approach to solve practical problems.

Using proper software resources, such as MATLAB, which is a multi-paradigm numerical computing environment and proprietary programming language developed by MathWorks, can enhance the effective development of these teaching tools. In fact, MATLAB allows matrix manipulations, plotting of functions and data, implementation of algorithms, and creation of user interfaces, and interfacing with programs written in other languages. Although MATLAB is intended primarily for numerical computing, an optional toolbox allows the access to symbolic computing abilities. An additional package, Simulink, adds graphical multi-domain simulation and model-based design for dynamic and embedded systems.

In conclusion, the development of computational thinking can be enhanced by the introduction of an integrated use of the aforementioned teaching principles and computational tools intended to support the students in acquiring knowledge and developing problem-solving skills. At the same time, soft skills are also fostered to enhance communication and to improve the translation of real-world problems into the technical domain and technical issues back into common language.