Teaching Programming Based on Computational Thinking

Ying Li
School of Computer Science and Engineering
Beihang University
Beijing, China
liying@buaa.edu.cn

Abstract—Computational Thinking was an important way to study computer science and it explained the relationships between human, calculation and computer in terms of thinking. The computer science was raised from the theory of tools to the height of thinking. This paper introduced three levels of Computational Thinking: Computational Thinking in computer science, Computational Thinking in computing science and Computational Thinking in all subjects and analyzed the thinking characteristic of calculation and the calculation attribute of thinking. Through Computational Thinking, human can solve problems by using computer more efficiently. Therefore, the cultivation of Computational Thinking's ability was particularly important in teaching. But Computational Thinking cannot be taught without carrier and the programing course was the best way for they were closely related with each other. The paper took the Maze Problem as an example to express the differences between human and computer in the aspects of storage structure, thinking mode, solving procedure, etc. and it compared the human brain's multi-dimension and multi-forms characteristics with the computer's ne-dimension and one-form characteristics. The paper thought the goal of teaching was not to learn knowledge itself, but to form problem solving thinking

Keywords—Computational Thinking; Calculation; Human Brain; Programing

I. INTRODUCTION OF COMPUTATIONAL THINKING

In 2006, professor J. M. Wing, dean of School of Computer Science Department in Carnegie Mellon University, published an article in Communication of the ACM and first proposed the concept of Computational Thinking (CT). She said: " Computational Thinking involves solving problems, designing systems, and understanding human behavior by drawing on the concepts fundamental to computer science". [1] The literal meaning of CT indicated it neither computing thinking, nor computer thinking. In fact, CT referred to the scientists' problem-solving methods which were used for or based on calculation. CT was a way of thinking rather than a kind of technology and it was contained in the computer knowledge and human thinking. Therefore, CT belongs to the areas of scientific methodology. According to the application range of CT, it can be divided into three levels: **CT in computer science**, **CT in computing science** and **CT in all subjects**. CT in computer science mainly used the computer's principles and methods to solve real problems and it was the most basic and the most essential content of CT; CT in computing science was a way of thinking to convert the real problems into a finite computing process based on the computer's automatic calculation; CT in all subjects focused on how to solve problems in other subjects, production activities and human society by applying the computer science knowledge and computing theory and it had the characteristics of generality and diversity. [2] In this paper, we mainly analyzed the CT in the first level-CT in computer science, aimed at making learners recognize the difference between the practical problems in the real world and the ideal models in the computer world and understand what can be solved by computer's calculation and what can be only implemented by the human's thinking. Through some specific thinking training, learners can obtain a better solution by using computer more efficiently and more easily. CT can give full play to advantages of human and computer, respectively and can remove their cooperative obstacles.

CT first expounded computer science from the perspective of thinking. And based on this view, computer was considered as an implementation way of thinking rather than a kind of tools and computer science was regarded as a subject to develop learners' thinking ability rather than to teach them the use of software. Every discipline had its own way of thinking. First, CT was a way of academic thinking and due to the fact that it was abstracted from computing science, it had some obvious characteristics of calculation. Second, CT regarded calculation as a way of thinking, which made calculation have the attribute of thinking. Therefore, CT can be analyzed from two aspects: thinking with the characteristics of calculation and computing with the attribute of thinking. 1) Thinking with the characteristics of calculation: when human used the computer to solve problems, their way of thinking must satisfy the conditions of computer's calculation, which indicated human should think about problems in accordance with the computer's computing methods, computing rules and computing processes. In CT, the subject of thinking was human and the subject of calculation was computer and the key was to convert the real problem into a finite computing process. The conversion process was as follows: First, human abstracted the real problem into a formational expression by using algebraic language or other specification languages. Then, computer transformed the formational expression into some strings by the character conversion method according to the predetermined rules. Finally, the results can be calculated depended on the computer's strong computing ability. In the process, human's thinking showed the characteristics of calculation, which indicated the conversion of formational expression must be limited and its semantic must be clear and the conversion process must be implemented mechanically. 2) Computing with the attributes of thinking: computer was invented by human and its strong capability of calculation was only valuable when executing the algorithms designed by human. And algorithm itself was a way to express one's thinking. Calculation was the implementation of thinking and thinking was the realization of calculation. All the computing activities were completed under the guidance of human's thinking. Computer solved the problems by simulating human's thinking indirectly, approximately and limitedly. [3] CT explained the thinking behind computer from the perspective of computing, e.g. using divide-and-conquer thinking to decompose the complex task into some simple tasks; using recursion thinking to translate code into data and translate data into code; using division thinking to abstract the real problem into some solvability steps; using iteration thinking to execute the same or similar process repeatedly.

In conclusion, thinking made computer have the ability to solve problems. CT was a way of thinking, which was used to connect the real world and the computer world and with the guidance of CT, human can understand and use computer efficiently. Furthermore, human can understand and reform nature based on the principle of calculation. Therefore, everyone should have the ability of CT.

The rest of the paper is structured as follows: Section 2 instructed the related researches of CT. Section 3 presented the relationships between CT and programing course. Section 4 analyzed the differences between CT and human brain when solving problems. Section 5 proposed some methods to teach programing based on CT. Finally, section 6 showed some conclusions and outlines future work.

II. REALATED WORK

CT represented the development of thinking mode in the field of information technology. But CT was not a new concept. To cultivate thinking ability under the support of leaning computer had been proposed for a long time. Early 80s in the last century, the study of LOGO programming language, advocated by Papert, a professor at the Massachusetts Institute of Technology, was a typical representative. [4] In 2000, DiSessa, an education researcher who was one of the developers of the LOGO, proposed the concept of Computational Literacy and he emphasized the role of the computer- as a tool and medium for exploring other disciplines. [5]

Later, in 2006, Computational Thinking was proposed by J. M. Wing, which had profound effects on computer science and the related researches had risen to unprecedented heights. J. M. Wing thought Computational Thinking had two important characteristics: abstraction and automation, which included the methods of recursion, parallel thinking, decomposition, ect. [1] Her definition emphasized the role of Computational Thinking, which had universal sense. J. M. Wing also described the extension of Computational Thinking by analyzing the value of computer science's methods and principles in the aspect of solving problems. In 2011, J. M. Wing regarded the Computational Thinking as a process of thinking further. She thought the core of Computational Thinking was to find and abstract problems, represent the solution and calculate the result by means of information technology and methods. This definition focused on the methods of solving problems. [6,7]

In 2011, International Technology Education Association (ISTE) collaborated with Computer Science Teachers Association (CSTA) gave an operational definition of Computational Thinking. In the definition, Computational Thinking was regarded as the process of solving problems. [8] When using Computational Thinking, it needed to find a problem which can be solved by computer and to abstract data through model and simulation and to calculate the result automatically through designing algorithms. [9] Further, the solution of one problem can be applied to other similar problems. The definition proposed by ISTE and CSTA described the process of solving problems based on Computational Thinking and introduced a clear procedure. [10] In Sep. 2012, the UK's Department for Education published a new computer course plan, "Computing Programmes of study" based on Computational Thinking. The Royal Society considered Computational Thinking as different levels of calculation to recognize the real world and the knowledge of computer science was applied to differentiate and analyze the natural systems and artificial systems. [11] Later, ISTE and CSTA gave a more detailed definition and they thought Computational Thinking should include nine parts: Data Collection, Data Analysis, Data representation, Problem Decomposition, Abstraction, Algorithms and Procedures, Automation, Simulation and Palatalization. [12] In Feb. 2014, College Board in the United States released a course outline of Computer Science Principles, which indicated the course was designed in accordance with the practices and concepts of Computational Thinking for high school students. [13] Coincidentally, Google applied Computational Thinking into its product development. He thought Computational Thinking was the combination of problem solving skills and technology, which was regarded as problem decomposition, pattern recognition, model creation and algorithm design. [14]

Computational Thinking had also drawn more and more attentions of educators and computer scientists. E.g. Settle and Perkovic in DePaul University designed a conceptual framework of Computational Thinking based on the The Great Principles of Computing proposed by Peter J. Denning, which indicated Computational Thinking involved the aspects of calculation, communication, cooperation, memory, automation, evaluation and design. [15] Shuchi Grover and Roy Pea summarized some elements of Computational Thinking, which included the abstraction and creation of model, systematic information processing, symbol system and its representation, the algorithm expression for flow control, structured problem decomposition, iteration, recursion, parallel processing, conditional logic, efficiency and performance constraints, abnormal monitoring and troubleshooting. [16] Professor Richard m. karp proposed a project "computational lens", which considered calculation as a general way of thinking. The human's reconstruction of the nature and society was described through this broad sense of Computational Thinking, involved in information processing, algorithm execution, time-space complexity, etc. And this new view of Computational Thinking had converted the computer science from the initial numerical calculation tools to a general thinking mode. [17]

III. THE RELATIONSHIP BETWEEN COMPUTATIONAL THINKING AND PROGRAMMING COURSE

CT was a thinking method and CT ability was a human's problem-solving ability to solve problem by using automatic computing devices under the guidance of CT. The basic goal of computer teaching was to develop learners' thinking abilities to make them solve the real problems by applying CT consciously. But thinking was a human brain's indirect and abstract reflection of the essence of objects and their inner connections and it was also a main form of the human intellectual activities. How to explicitly express the thinking activities internalized in human brain and to make learners see and touch the abstract computer knowledge with non-physical properties, carrier was very important. The programming course was a basic computer course, which had a close relationship with CT. Edsgar Dijkstra, one of the famous computer experts had published a book "The Teaching of Programming i.e. the Teaching of Thinking" in 1976 and he said: the tools we used had affected the ways and habits of our thinking, thus they will also profoundly affect our thinking ability. Therefore, CT and programming had a closed relationship. CT provided some efficient thinking methods to programing course and programing course provided a practical carrier to the cultivation of CT ability.

A. Cultivating CT ability was the goal of programming course

The contents, e.g. abstracting problems, creating models, designing algorithms, verifying results, were taught in programing course by applying the methods of CT, e.g. heuristic inference, separation of concern, simplification. The goal of the programming course, making learners understand how to solve the real problems by computer, was the embodiment of training CT ability.

In traditional programing course, due to lack of the specific and effective guidance of thinking, the contents taught were limited into the programming language itself, the problems solved were the ideal models which were abstracted and processed in advance and the methods applied were lack of universality, engineering quality and practicalness. It led the teaching effect differed very far from the educational goal. As we know, the programing languages were different from each other, but the problem solving and thinking methods were the same or similar. The introduction of CT made the course pay more attention to teaching the ideas and methods of thinking by using the problem-driven teaching method, which taught computer knowledge according to the process of finding a real problem, associating human's solving method, mapping it into computer's model and designing algorithms. The teaching method based on CT cannot only make learners understand the collaborative relationship between human's thinking and computer's computing, but also make them master the general methods of thinking summarized by CT. CT connected the real problem with the computer model efficiently. The programing course based on CT can teach the thinking process explicitly, which were just understood by the leaners themselves before. And learners can solve all kinds of real problems through the study of only one programing course.

B. Programming course was an effective carrier for CT:

In fact, CT was a form of thinking which was shown in the process of solving problems. CT was implemented by the automatic and mechanical calculation with the aim of solving problems. The expression of CT and the cultivation of CT ability should be supported by the knowledge of compute. And the programing course, with the features of digitization, computerization and programmation, was the effective carrier to train the ability of CT. It cultivated the learners' abilities to find problem, analyze problem, design algorithm, compare performance and solve problem through teaching the specific methods and principles of the computer. Programing fit perfectly with the thinking process of CT. And the abstraction and calculation concerned by programing course were just the core contents of CT. Therefore, programing course can most and best reflect the thinking process of CT.

1) Programing was the most appropriate way for expressing CT.

CT should be described through some languages or words. And it was meaningless without explicit expression. Further, the expression of CT must follow some strict rules and specific language formats, otherwise it cannot be understood. Programing language itself was a kind of formal expression with the characteristics of determination, finiteness and mechanization. Programming language can express CT accurately and the expression can be understood by computer easily.

2) Programing course included the thinking methods of CT.

Programing course converted the unknown problems into the known problems by applying CT's methods of reduction, transformation, simulation, etc. Programing course simplified the complex problems into simple ones by applying structural-programming and function based on the CT's Separation of Concerns approach. Programing course solved the uncertainty problems based on the CT's heuristic approach. Programing course improved the calculation speed by using CT's parallel approach and the evaluation of time and space proposed by CT was also the important index for measuring the performance of algorithms; Programing course focused on the robustness and reliability and algorithms can be recovered in worst-cases through CT's methods of prevention, protection, redundancy, fault-tolerant, damage containment and error-correction, etc.

3) The practice in programing course was the important way for training CT ability

The cultivation of CT ability needs to experience the real case and the obtained ability had to be tested in practice. Practice teaching can make learners experience the process of writing program and forming CT. Through lots of thinking training and practice, programing course made the superficial and abstract thoughts and knowledge tap into learners' inner thinking systems and learners can solve the problems by applying the appropriate CT methods and effective computer knowledge actively or unconsciously. The practice in programing course can internalize the knowledge into ability and can merge the programing with CT.

In conclusion, with the application of CT, the programing knowledge ran throughout the whole process of solving problems for the ultimate goal of forming a complicated thinking network, which took the knowledge as nodes and the CT as connectors. CT made the contents of programing course approach the real problem as far as possible and it expanded the human thinking and computer knowledge from computer to the real world by breaking the limitations between human, the real world and the computer science. CT and programing had a natural relationship, which indicated the cultivation of CT ability needs the support of programing and programing needs the guidance of CT. Therefore, it made sense to introduce CT into programing. It was necessary to teach knowledge along with to teach its contained thinking. In this way, it can strengthen knowledge through thinking and can cultivate thinking thorough knowledge. Ultimately, it can both help learners learn programing language and cultivate their CT ability.

IV. Understanding the Difference Betwwen Human and Computer Based on Computational Thinking

The introduction of CT can both help learners to better understand the differences between human brain and computer and make them more easily convert the thought exited in human brain to the programs executed by computer. In this paper, we took the maze problem as an example to show the differences between human brain and computer, as shown in Figure 1.

1) Different behaviors to store information

If we want to use computer to solve problems, the first thing was to input and store information into computer for further processing. But human brain and computer had great difference in storage. Human brain was regarded as a complicated system with the characteristics of multi-dimension and multi-forms, but computer was considered as a simple system with the characteristics of one-dimension and one-form.

Human brain stored the information based on a neural network composed of neurons. The multi-dimension indicated the information was stored in several neural sub-networks with multi-dimensional storage space and different functions (vision, hearing, sensing, thought). The multi-forms indicated the things in the real world were abstracted, encoded and organized into diverse forms stored in human brain, including sense organ information, cognitive information and consciousness information. Sense organ information was received directly by different sense organs; cognitive information was represented by selecting the appropriate language, words and symbols to process and integrate the sense organ information; consciousness information was formed by further understanding and abstracting the sense organ information and cognitive information, which was used to control human's thinking activity. Therefore, the storage space of human brain was multidimensional and the storage type was multi-forms. As shown in Figure 1(a), the maze of a real problem was found by human and it can be simplified and stored into human brain in the form of a graph, as shown in Figure 1(b). Conversely, the storage system in computer was one-dimension and one-form. The one-dimension indicated the memory construction was linear and the data can be accessed according to a sequential numeric address. The one-form indicated all the things will be converted into binary digits and can be represented in only one data form.

The different storage behaviors between human and computer had brought huge difficulties for representing the objects and their relationships. And the first task to use computer was to convert the multi-dimension and multi-forms information stored in human brain into one-dimension and one-form binary digits stored in a linear space. In the conversion, Dimensionality Reduction Thinking (DRT) of CT played a key role. DRT simplified the complex problems in the real world into some simple ones which can be solved by computer through reducing the dimension of thinking space from human's multi-dimension to computer's one-dimension. In the process of reducing dimension, the loss of information and the error of the expression should be considered. In teaching, the learners should understand and master how to store the objects and their relationship in the computer's linear space by using DRT.

- Representing the objects by using DRT:

The objects can be divided into two types according to its priority and they were digital type and non-digital type. And the digital type can be further divided into numeric type and Boolean type. Due to the representation and storage of digital information between human and computer were the same or similar, the computer can directly store it into its memory space in binary form without dimensionality reduction. But for the non-digital information, e.g., text, graphic, image, sound and video, we must convert their complex forms stored in human brain into a simple digital form by using digitalization method. In the programing course, learners just need to understand some simple quantization process and the more important thing was to master a method of pattern-programming. They can established a mapping relationship between the objects in the real world and the data types in the computer and can select the correct and appropriate built-in data types of programming language to describe all the things. E.g., in the maze problem, the number of nodes and the length of path can be represented directly with the data types of int and double. But the map construed by all nodes will be converted into the connectivity between two nodes, which was represented by 0 or 1, as shown in Figure 1(c).

- Representing the relationships by using DRT:

When using computer to solve problems, it should not only store the objects, but also store their relationships. According to interaction between objects, the relationships can be divided into four types. They are set, linear, hierarchy and net. Due to the characteristics of multi-forms, human can directly store all kinds of complicated relationships, which can be represented intact and human can know all of the relationships at a time. But computer was a linear system and was only suitable for representing the simple relationships of set and linear. The hierarchy and net must be divided and converted from one-to-many or many-to-many nonlinear relationships to one-to-one linear relationships by using DRT. The all relationships between the objects can only be represented indirectly and gradually with the help of the relationship between the precursor object and the successor object. The whole relationship can only be known step by step. E.g. the map path in maze can be abstracted and represented with array and the connectivity between nodes can be indicated by 0 or 1, as shown in Figure 1(e). And the search path was obtained by the path between two nodes.

DRT is an important thinking method of CT, which made learners understand the different storage and representation between human brain and computer. The essence of DRT was to use simple structures instead of complex ones, which indicated it converted all kinds of data types (digital type and non-digital type) into binary form and stored all kinds of relationships (set, linear, hierarchy, net) in a linear structure. When simplifying the complex information existed in the real world into the simple form stored in the computer, learners should also know the time and space consumed by DRT

2) Different approaches to solving the problems

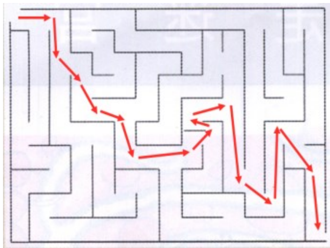
When the objects and their relationships were stored in the computer, it was to choose the right methods to solve problems. Human brain and the computer had big differences on the information processing. Human brain can obtain the whole information all at once and can process them concurrently. But the computer can only access part of the information and can only process them serially.

For human, the eyes were the main organ to receive information and it can get the complete picture of object during moving, which made it possible to solve problems from a global and unitary perspective. E.g. In the maze problem, human brain can implement global search by reading the whole map at a time. But computer did not have the function of human eyes. At some point, it can only access the information of the next path directly, which was associated with the current node. In order to get the result, computer can only calculate the search path step by step according to some problem-solving procedure under the constraint of the heuristic condition. Therefore, human brain can obtain the results through automatic memory and global search, but computer can only get the closer and closer result by means of iterative computation with the methods of task decomposition, repetitive calculation and procedure storage.

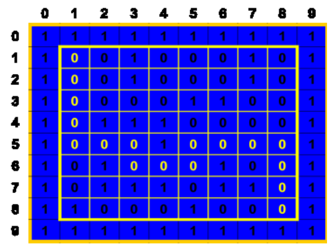
In addition, human brain can deal with several tasks concurrently, which was called multitasking. Therefore, human brain can read the maze map and research the efficient path, simultaneously. The basic reason was human brain was a distributed processing system and it can trigger multiple neurons to process information in parallel. But to the ordinary single-core computer, it processed the information serially according to the instruction sequence, which was considered as a means of centralized processing. Computer can only do one thing at the same time.

How to convert human's distributed parallel processing method to the computer's serial processing was the difficulty in programing teaching and it was the reason of introducing CT into programing teaching. In the class, teachers should both teach the computer's methods of solving problems according to its own procedure and introduce the reason from the perspective of CT. As what mentioned before, computer's memory was a linear structure, which resulted in its problem-solving methods without global view and overall view. Computer can only solve the problems by means of task decomposition and repetitive calculation. Therefore, loop was computer's main way to solve the problems and the main body of each algorithm was realized through loop. Loop was also an implementation of heuristics. It corrected the next input with the current output by using the same or similar computing method and obtained the result through iteration. The key of loop was to find the loop variable and loop expression. The realizing process was: First, by using Separation of Concerns, the integral solving process in human brain was decomposed into several sub processes in computer, which can be executed in single-step; Second, the loop variable existed in each of the sub-processes was found through observing, which was updated steps by steps and played a key role; Third, the general loop expression, which was appropriate for the whole sample space, was generalized by analyzing the execution process of some special samples, respectively. E.g. in the given maze problem, the global search process in human brain was divided into several sub processes by taking each node as a unit. Then, the search path was determined by the connectivity of each node, which was changed in each loop. Therefore, the node was the loop variable. Finally, the loop expression can be obtained by searching some path from the starting node. For each node, the effective path can be found according to some specific order, e.g. left-down-right-top, as shown in Figure 1(d).

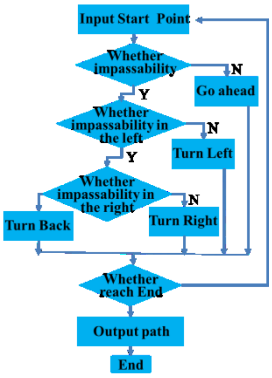
(a) Maze problem in the real world



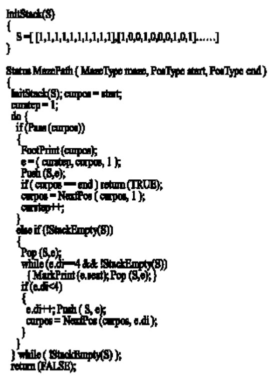
(b) Maze info got by human brain



(c) Maze digitized into 0 and 1



(d) Rlow chart of Maze



(e) Programing of Maze

Figure 1 The process of solving Maze problem

3) CT connected the real world with the machine world

The goal of teaching programing was to make learners understand how to solve problems by using computer knowledge. CT played an important role of "Bridge" to closely connect the independent and abstract knowledge in the machine world with the objects and their relationships in the real world. E.g. CT connected the objects (digit and non-digit) with different data types, connected the relationships (set, linear, hierarchy, net) with different data structures and connected the real problems with different algorithms (Greedy, Sort, Dynamic programming). In this way, when solving the real problems, learners can consciously and effectively find an appropriate storage type, data stricter and algorithms. Further, through the training of CT, human can solve problems by unconsciously applying the classic methods in computer science and it cleverly achieved the application fusion between human and machine and the knowledge fusion between different subjects.

V. TEACHING PROGRAMING BASED ON COMPUTATIONAL THINKING

The goal of teaching was learning and the learning of programing did not only understand the programing language, but mastered the approaches to solving the real world problems. The introduction of Computational Thinking made teachers to teach programing from the perspective of human's thinking. They taught the process of converting the real problems into the algorithms and programs by exploring the methods of CT behind the knowledge. With the help of CT, leaners can get full training in all aspects and they can solve the real problems independently by using the way of CT.

1) Teaching programming combined with human's thinking

The teaching content had been adjusted and reorganized according to human's thinking process. When human solved a problem, they first recognized the objects involved in the problem through their types, shapes and other physical characteristics, and then analyzed their relationships. After getting the basic information of the problem, human will discover the essence and law of objective things through the use of logic. Finally, they thought of the way to solve the problem based on their prior knowledge. Similarly, when using computer to solve the problem, it will find the appropriate data type and data structure to represent and store the object and their relationships, respectively. Then the conditional control statements were applied to represent the logical relationship between the objects and control the procedure of solution. And the algorithms were formed in the process of implementation. When solving problems, the relationships between human, computer and CT were show in Figure 2.

2) Changing the way of teaching and learning

To develop the learners' CT abilities, the roles of teachers were adjusted. The teacher was no longer the body of the course, but the guider to help students to acquire knowledge, the counselor to answer questions, the helper and organizer to manage learners' learning activities. Teaching was the process to help learners transform problems, construct knowledge, decompose tasks and provide the learners with the space of their all-round development and practice.

CT had changed the role of leaners from passive learning to autonomous learning, which realized learners' subjective status by means of advocated initiative, cooperative and probing learning methods. In the study, learners should involve their emotion and acquire the positive emotional experience. Through the training of CT, learners can understand the methods of Separation of Concerns, simplification, transformation, simulation and heuristics in the experience of solving the problems. Further, learners can build their own knowledge system and have some creative problem-solving abilities. CT accelerated the generation of thinking form the accumulation of knowledge.

VI. CONCLUSION

CT had a profound effect on computer science. The teaching based on CT was different from the ordinary teaching, which paid more attention to explain the process of solving problems and the formation of thinking. To learners, their simple memory and mechanical calculation were dramatically inferior to those of computer. Therefore, they should not focus on how to memory the computer knowledge or to imitated the computer's way of solving problems, but on how to form their own thinking and consciousness through learning. CT cannot be taught without carrier and the basic knowledge and basic methods were the foundation of teaching CT. It was an effective way to cultivate the CT ability through programing. But learning the methods of CT or mastering the skills of programing was not the basic goal of computer education. We should focus on the characteristics and essence of CT, not on its forms.

The introduction of CT into computer teaching had put forward new requirements for teaching and learning. To reorganize and redesign the computer knowledge according to the methods of CT were a meaningful and challenging work.

REFERENCES

[1] Wing J M. *Computational Thinking*[J]. Communications of the ACM. 2006, 49(3), 33-35.

[2] Chongli Jiang. *My view of Conputational Thinking*.[J] China University Teaching. 2013, 9: 5-10.

[3] Chongwen Shi. T*he Calculaiton of Thinking and the Thinking of Culculaiton*[J]. Computer Science. 2014, 41(2): 11-14.

[4] A. Nagchaudhuri. *Development of mobile robotic platforms as freshman engineering design project*. 34th Annual Frontiers in Education (FIE). Oct. 2004, 9-13.

[5] Jonathan Ozik, Nicholson T. Collier. *The ReLogo agent-based modeling language*. 2013 Winter Simulations Conference (WSC). Dec. 2013, 1560-1568.

[6] Wing J M. *Computational thinking and thinking about computing*. Parallel and Distributed Processin (IPDPS). Sept. 2011, 3-7.

[7] Wing J M. *Computational thinking*. Symposium on Visual Languages and Human-Centric Computing (VL/HCC) . April 2008, 1-5.

[8] Peter J. Denning. *Great principles of computing*[J]. Communications of the ACM, 2003, 46(11): 15-20.

[9] Cuny J, Snyder L, Wing J M. Demystifying CT For Non-Cmoputer Science.[J] Work In Progress, 2010, 8(3):30-36.

[10] Aravind Mohan, Shiyong Lu, Ke Zhang. *Towards an Online Service for Learning Computational Thinking Using Scientific Workflows*. Services

Computing (SCC), 2015 IEEE International Conference on. June 2015, 340-347.

[11] Kuo-Chuan Yeh, Ying Xie, Fengfeng Ke. *Teaching computational thinking to non-computing majors using spreadsheet functions*. 2011 Frontiers in Education Conference (FIE). Oct. 2011, 1-5.

[12] Benjamin Worrell, Catharine Brand, Alexander Repenning. *Collaboration and Computational Thinking: A classroom structure*. Visual Languages and Human-Centric Computing (VL/HCC), Oct. 2015, 183-187.

[13] Ross J. Toedte, Mehmet Aydeniz. *Computational thinking and impacts on K-12 science education*. 2015 Frontiers in Education Conference (FIE), Oct. 2015, 1-7.

[14] Wang Chun-Hong, Wang Qiao-Ling. A study of the Classified teaching of "Fundamentals of Information Technology" course based on computational thinking. 3rd Communication Software and Networks (ICCSN), May 2011, 702-704.

[15] Aaron Bauer, Eric Butler, Zoran Popović. *Approaches for teaching computational thinking strategies in an educational game: A position paper*. Blocks and Beyond Workshop (Blocks and Beyond), Oct. 2015, 121-123.

[16] Peter B. Henderson. *Ubiquitous Computational Thinking*[J]. Computer. 2009, 42(10): 100-102.

[17] Karp R. *Understanding Science Through The Computaional Lens*[J]. Journal of Computer Science And Technology. 2011, 26(4), 68-75.

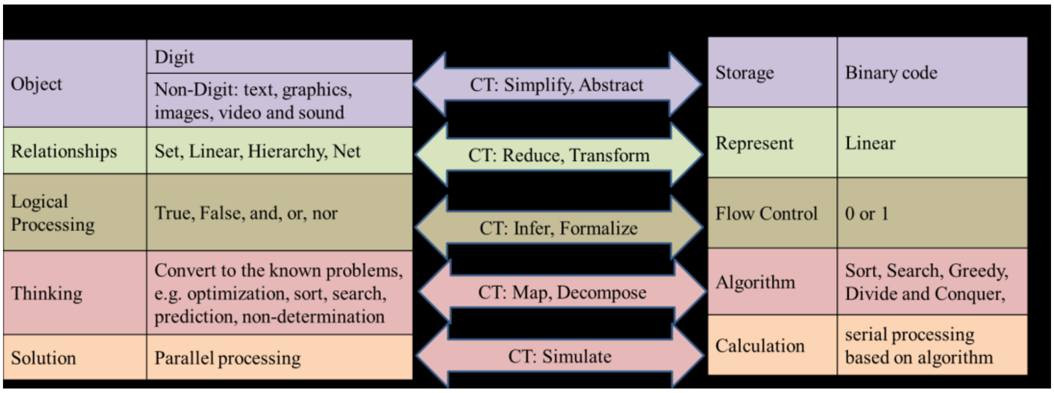


Figure 2 The relationships between Human, Computer and CT