

Computational Features of the Thinking and the Thinking Attributes of Computing: on Computational Thinking

Wenchong Shi, Maohua Liu

College of Mathematics & Information Technology,
Hebei Normal University of Science & Technology, Qinhuangdao, China
E-mail: qhdshwch@126.com, ytulmh@gmail.com

Peter Hendler

EECS, University of Ottawa, 800 King Edward, Ottawa, ON, K1N 6N5, Canada
Email: peter_143@eecs.uottawa.ca

Abstract—The paper aims at revealing the essence and connotation of Computational Thinking. It analyzed some of the international academia's research results of Computational Thinking. The author thinks Computational Thinking is discipline thinking or computing philosophy, and it is very critical to understand Computational Thinking to grasp the thinking's computational features and the computing's thinking attributes. He presents the basic rules of screening the representative terms of Computational Thinking and lists some representative terms based on the rules. He thinks Computational Thinking is contained in the commonalities of those terms. The typical thoughts of Computational Thinking are structuralization, formalization, association-and-interaction, optimization and reuse-and-sharing. Training Computational Thinking must base on the representative terms and the typical thoughts. There are three innovations in the paper: the five rules of screening the representative terms, the five typical thoughts and the formalized description of Computational Thinking.

Index Terms—Computational Thinking, computational features, thinking attributes

I. INTRODUCTION

In 2005, the report submitted for the president by the President's Information Technology Advisory Committee *Computational Science: Ensuring America's Competitiveness* pointed out, although computing is a discipline, it can promote other discipline's development. The committee thought, in the 21st century, all the frontier researches, the most important in science and the most promising in economy, could be solved by advanced computing and computing science. [1]

In 2006, a Chinese-American Jeannette M. Wing put forward 'Computational Thinking (CT)' which caught the attention of the international academia. In America, Computer and Information Science and Engineering Division (CISE) launched the CISE Pathways to Revitalized Undergraduate Education in Computing

(CPATH) program in 2007 and the National Science Foundation (NSF) launched the Cyber-Enabled Discovery and Innovation (CDI) program in 2008. Stanford University and Carnegie Mellon University have implemented the reform of computer courses for this one after another. In China it also attracted many scholars' attention; the Coalition of 9 Universities (C9) put forward the proposal developing CT in computer elementary education of higher education institutions.

In 2011, The Information Strategy Researching Group of Chinese Academy of Sciences (CAS) thought, the discipline computer science and technology had been constructed as a very professional instrumental discipline. The narrow instrumentalism is very harmful for popularizing IT; cultivating CT is the effective approach resisting the narrow instrumentalism, it is also the base for solving other IT puzzles. [2]

In a word, CT is an important topic of the international academia. Research on CT is very meaningful presently. Grasping the essence of CT and enriching CT's theoretical system are of great benefit to people to cultivate and apply CT. This paper first analyzes the essential connotation of CT, finds some representative terms with the features of computing science, reveals the typical thoughts of CT and gives the formalized description of CT.

II. ESSENTIAL CONNOTATION OF COMPUTATIONAL THINKING

People paid attention to the relationship between thinking and computing hundreds of years ago. In the early 18th century, Gottfried Wilhelm Leibniz, the inventor of the multiplication machine and the founder of the binary numeral system, put forward Symbol Thought, Calculus Ratiocinator and Symbolic Logic. They explained that thinking's essence is computing for the first time; in 1950, Turing demonstrated the computing essence of mind in detail in the paper *Computing Machines and Intelligence*. Since then, that mind's

essence is computing has become the basic hypothesis of artificial intelligence theory, and it is another statement of the sentence 'thinking's essence is computing'. Academia also paid attention to the impact of the discipline to thinking and to worldview. At the end of the last century, Howard Gardner, the developmental psychologist of America, pointed out, 'Every subject is thinking and understanding the world in its own way', 'Communicating with the symbolism and style of a discipline, is developing discipline thinking' [3].

In 1997, the Calculation and Philosophy Branch of American Philosophical Society launched a national poll on 'How computers are changing philosophy'; the result indicated, 'A new philosophical paradigm has emerged.' [4] Chinese scholars echoed the view actively and thought 'Computing has become a new worldview' [5].

In 2011, the Turing Award winner Richard Karp put forward 'Computational Lens'. He advocated taking computing as a common way of thinking to solve the problem of every discipline [6]. Actually, he advocates computing philosophy.

The first modern computer scientist who had realized the influence of computation on thinking was Edsger Wybe Dijkstra. In the early 80's, this Turing Award winner talked in an article on programming, 'The tools we use have a profound (and devious!) influence on our thinking habits, and on our thinking abilities.' [7]

In 1992, Chinese scholar Huang Chongfu first defined CT from perspective of artificial intelligence [8]; in 2002, Dong Rongsheng et al. put forward Methodology for Computer Science & Technology [9]. This theory did not talk about its universal guiding significance to non-computer sciences, and could not be called computing philosophy, it did not yet mention the word 'Computational Thinking', but it has revealed the thinking attribute and the laws of computing.

In March 2006, Professor Jeannette M. Wing published *Computational Thinking in Communications of the ACM*. She thought, 'Computational Thinking involves solving problems, designing systems, and understanding human behavior, by drawing on the concepts fundamental to computer science. CT includes a range of mental tools that reflect the breadth of the field of computer science.' [10]; In 2010, she gave CT another explanation in an article written by the coauthors and her— 'CT is the thought processes involved in formulating problems and their solutions so that the solutions are represented in a form that can be effectively carried out by an information-processing agent'[11]; which is completely different from her earlier statement.

In 2010, Steering Committee of Computer Science and Technology of China Ministry of Education pointed out, Computational Thinking can be understood as considering solving problems by the basic approaches by which a computer solves problems, so as to construct a corresponding algorithm, basic programs and the like. The committee decomposed the professional ability of computer science and technology into 82 points; there are nine points of them that concern Computational Thinking. The percentage is more than 10%. The committee thought,

Computational Thinking ability mainly includes expressing the problems and solving them by some symbols, logical thinking and abstract thinking, formal verification, modeling, realizing class computation and model calculation, applying computer technologies and other things. [12]

Similar to Chinese committee above-mentioned, in 2011, the International Society for Technology in Education (ISTE) and Computer Science Teachers Association (CSTA) proposed the 'operational definition' of CT. Perhaps they thought CT defined by Wing is 'discommodious to operate', and interpreted it as a problem's solving process with six steps [13].

Above all, CT is nothing but discipline thinking or a computing philosophy. Scholars all over the world have a variety of expressions on CT. However, Wing's *Computational Thinking* is more vivid, more popular and more revealing. Einstein said, 'The more widely a concept, the more frequently it enters people's horizon, the more difficultly we understand the meaning of it' [14]. Explaining 'what is computation' or 'what is thinking' is very difficult. Understanding CT is still harder. The author thinks, since CT is abstracted from computing science, it must possess the obvious computing science's characteristics, and since 'computation' has been sublimated as 'thinking' already, it must have thinking attributes. So to master and research CT, it is necessary to start with two aspects—the computational feature of the thinking and the thinking attribute of computing, and it can not be restricted to a simple definition. Researching CT should aim at the idiomatic thinking methods of computing science; learn from CT's positive and beneficial factors, so as to serve for various social practices, which is the fundamental purpose of 'let the masses think like a computer scientist' [15].

III REPRESENTATIVE TERMS OF COMPUTATIONAL THINKING

What could embody CT's characteristics of computing science? Wing mentioned 'drawing on the concepts fundamental to computer science' in the paper *Computational Thinking*. CT is abstracted from computing science, and the basic concepts are the important part of computer science, so these concepts should be the best carriers of the characteristics of computing science. As far as what 'fundamental concepts' to 'draw on', Wing had not indicated them explicitly. In America, Alan Bundy has noted 'the infiltration of computational concepts into other disciplines' theories' [16]. Many social organizations have recognized 'Most efforts have not focused on fundamental concepts', and discussed some fundamental concepts [17]. In China, some scholars presented scattered narratives [18]. A scholar mentioned with suspicion two quite different views from each other of CT [19]. The result of the questionnaires publicized by a website to cultivate CT of a university also reflected the two views [20]. The first view includes 12 concepts and the second contains 25 concepts (see Table 1).

Slightly analyzing, one would find that the 12 concepts of View 1 come from Computing Curricula 1991 (CC1991) launched by ACM and IEEE society of America; the former 7 concepts of View 2 derived from *Great Principles of Computing* written by Peter Denning, an American computer scientist; the latter 18 concepts obviously come from Wing's original work, without increase or decrease.

In the writers opinion, there are many fundamental concepts in computing science, and searching for representative terms is critical. CC2004 has already admitted that computer science 'is hard to define as a single discipline' and 'covers a number of other important subjects' [21]. The 12 'core concepts' from CC1991 are just 12 'core problems', and the 'core concepts' are not equal to the 'fundamental' concepts. Besides, CC1991 formed more than 20 years ago. Nowadays, even the 'core problems' have already changed. On the other hand, the concepts presented by some authoritative publications can not be simply copied or misused; Even as to the special writings on CT, interpreting out of context and screening vocabularies from the original as the representative terms are ridiculous. The representative terms of CT should be established according to certain principles or standards. They should have the following features:

TABLE 1.

TWO VIEWS OF THE FUNDAMENTAL CONCEPTS

View	Concepts in the view
View 1	Binding, complexity of large problems, conceptual and formal models, consistency and completeness, efficiency, evolution, levels of abstraction, ordering in space, ordering in time, reuse, security, tradeoffs and consequences.
View 2	Computation, communication, coordination, recollection, automation, evaluation, design, reduction, embedding, transformation, simulation, recursion, parallel processing, abstraction, decomposition, Protection, redundancy, damage containment, error correction, systems recovery, heuristic (reasoning), planning, learning, scheduling, trade-offs.

(1) They must be the terms of computing science in order to embody the computational features. 'Efficiency', 'security' and 'conclusion' among others are not specialized vocabularies for computing science and not representative terms. The words 'abstract', 'reduction', 'binding', 'conversion' and so forth are far from the terms of computing science, they cannot become the representative terms either;

(2) They must be fundamental or basic terms, so that they are easier to understand, easier to research and popularize CT. It is probably because of this that computing science has surpassed and broken through the areas of computer science (CS) and computer engineering (CE) [22], but Wing still stressed the concepts' fundamentality;

(3) Their frequency of use must be high enough, so that they can embody the common features of computing science, and CT must be contained in the common features;

(4) They can reflect discipline conception or cognitive style. This can make them fundamentally embody the thinking attributes, so it is especially important;

(5) The intensions and extensions of them must be moderate. They can not be located at root or leaf node of the tree of the computing science knowledge. For example, the word 'computation' is very general and very abstract, it is difficult to grasp its connotation, and it can not be a representative term.

Admittedly, the five rules above-mentioned are relative to some extent. Especially the hierarchy where the node is directly affects the number of the representative terms. The key is that the representative terms should be based on computation theory or theoretical computer science and should not involve too many branches. Here, the writers give six nodes and list some representative terms (see Table 2). It is easy to see; the words such as 'structuralization', 'model' and others appear frequently in the table.

TABLE 2.

THE REPRESENTATIVE TERMS FROM 6 NODES

Nodes	Related branches	Representative terms
Data	Data structure, database	Stack, tree, queue, data type, integrity, consistency, structuralization, redundancy, data model
Algorithm	Algorithm	Exhaustion method, recurrence method, recursive method, divide-and-conquer method, backtracking, search, sort, complexity
Program	Programming	Encoding, pattern, modularization, structuralization, consistency, transportability, testability, optimization
Software	Software engineering and software process	Reuse, modeling, formal methods, structured method, object oriented, visualization
System	Operating system, information system, etc.	Process, threading, concurrency control, priority level
Network	Internet, web design, web programming	Protocol, the OSI model, topological structure, hyperlinks

It must be pointed out, the representative terms abstracted from computing science are helpful to cultivate or train CT step by step and on purpose. CT exists not only in the representative terms, but also widely in the methods or tools of every branch. It must not be exclusively confined to those representative terms to cultivate CT.

IV TYPICAL THOUGHTS OF COMPUTATIONAL THINKING

In another paper, Jeannette M. Wing pointed out; the essence of CT is abstraction and automation. 'We operate by mechanizing our abstractions, abstraction layers and their relationships.' The key is 'identifying appropriate abstractions and choosing the appropriate kind of computer for the task' [23]. According to this, most importantly, researching CT must find out the relevant 'computer' or abstract the corresponding 'mechanization'. Clearly, 'computer' or 'mechanization' is not a tangible

matter or system, it should be something intangible or invisible, and belong to worldview or methodology. Pat Philips thought of the essence of CT as ‘thinking about data and ideas, and using and combining these resources to solve problems’. He pointed out, CT means ‘think computationally’; it needs to ‘move technology projects beyond “using” tools and information toward “creating” tools and information’; he thought, to apply CT, five key questions, which are related to the power and limit, difficulty, method or approach, technological application, computational strategies, must be answered[24]. Clearly, he focused attention on the thinking attribute or thinking process as well.

In China, Wu Wenhui et al. proposed ‘the ways of Computational Thinking’ when introducing and interpreting the common skills in programming [25]. Among of them, ‘dividing and conquering’ (categorical of algorithm) is indeed one of the typical approaches of computer science; He Mingxin thought ‘separation of concerns’ as one of the important principles of CT. He pointed out that ‘divide-and-conquer’ and ‘modeling’ are its other expressions [26]. Zhao Lingzhong et al. thought greedy strategy, divide-and-conquer strategy, dynamic programming, backtracking and branch-and-bound strategy reveal CT very well [27]; Most of the textbooks of CT written by C9 have dedicated pages on algorithm knowledge. Obviously, many scholars have noticed the special significance of algorithm in CT.

Algorithm has its thinking connotations indeed. For example, recursion embodies the thought ‘applying repeatedly the operating rules of the former steps to the end’, and the divide-and-conquer algorithm embodies the thought of ‘breaking up the whole into parts, then solving them one by one’. However, the exhaustive algorithm, the recursive algorithm, the greedy algorithm and the like are usually used to solve mathematical problems; as the difficulty of an algorithm increases, its application is increasingly confined to narrower fields. For instance, iteration is often applied to numerical analysis; branch-and-bound algorithm is usually used for combination explosion problems. If the algorithm becomes the only way or tool for solving a specific problem, its thinking connotation would be weak. Therefore, ‘putting algorithm first’ is a narrow-minded view of CT. In the authors opinion, CT is thinking, and the result of thinking is thought. The outward manifestation modes of thinking should be scheme, stratagem or countermeasure. Researching the typical thoughts of CT is of great significance in understanding and mastering the thinking attributes of it, cultivating and making use of CT. The authors think, the related thoughts of CT exist widely in some representative terms. According to the representative terms appearing frequently in table 2, some typical thoughts can be concluded as follows:

A. Structuralization

In 1968, Donald Knuth wrote *The Basic Algorithms*, the first volume of *The Art of Computer Programming*. For the first time he systematically illustrated the logical structure and storage structure of data and their operation

techniques, and created the original system of data structure. This achievement made him the youngest Turing Award winner, and directly gave birth to the Donald E. Knuth Prize. Since then, the theory of data structure has become the theoretical basis of programming, compiling program, operating system, database system and the like. Structuralization has become an important thought in computing science and one of the most basic strategies.

Structuralization is formatting and standardization. It is helpful to the independent existence of things; it facilitates the access of objects; it benefits data’s reuse or being shared. There are many instances concerned in this in computing science. Binary system is structuralization and it is easy to design hardware and form electronic circuits. This is the most important theoretical foundation of computer hardware. On the aspect of data storage, data type is structuralization. Table in the relational model is also structuralization. A data item has its own name and data type. The establishment of data type of a data item means that its storage space occupied is determined and definite, which achieves the ordered storage of data and makes it easy to access data by its name. Data file’s structuralization directly leads to the transformation from the original file system to a database; In programming, the reasonable use of GOTO statement is helpful to keep the independence of the program segment, and it is the basic requirement of structured encoding; In the field of developing software, structuralization is a kind of traditional software development methods, structured analysis, structured designing, structured implementation are needed; Modularity is the basic requirement of structuralization. Structured Query Language (SQL) has become a standard, it can be independent of the various programming languages and DBMS software, and it has unified the access tools, simplified the access process, greatly promoted data’s cross platform operation, and has reached the highest level of the data reuse or data sharing. Therefore structuralization is the most important thought of computing science; it reflects the idea of formatting or standardization. It is an important part of CT.

B. Formalization

The 23 mathematical questions presented by David Hilbert in 1900 had been leading the development of mathematics and computer science. In the 1930s, in order to essentially illustrate what was calculable, Alonzo Church presented λ Operator, Alan Turing proposed Turing Machine and Stephen Kleenen introduced General Recursive Function. These theories laid the foundation of formalization thought. Since then, it has become the most basic method of computing science raising a question first and then revealing the essence of it with the most rigorous mathematical language, i.e. formal methods. Set theory, first-order predicate calculus and temporal logic are the basic theories of formalization; Model, logic algebra, process algebra and net are the basic tools of formalization. Here, the word ‘formal’ used to mean ‘standard’, ‘regular’ or ‘normal’. It reflects a rigorous scientific attitude.

Model is the basic means of formalization; it reflects the ideas such as abstraction and visualization so as to be understood and universal. The process of modeling or building model is an abstract thinking one. Graph model is one of the most widely used models, e.g. the relational model, the ISO/OSI model, etc., but there are also many text models, e.g. network protocols, etc. Actually, Turing Machines, Cellular Automata, λ Operator and general recursive function are the different forms of the models. Now, Tower of Hanoi almost becomes the intuitive model of recursive problems, five-dining-philosophers is almost the model of concurrency control.

The process of formalization is a typical thinking one. For example, there is a process thinking about consistency and completeness of the system in the formal specification of the formal methods(e.g., system modeling); the process of formal verification (such as simulation with software, software testing and so on) is the considering process of program's correctness. All of them fully embody the CT.

Furthermore, analysis indicates that programming language is formalization; algorithm is formalization as well. Stack, queue, tree and logic operation are all formalization. These concepts or methods have become the important tools of computing science. Moreover, even some words used in daily life(e.g., information, learning and safety) have opened up many new research areas by the formalized definitions. Formalization, so to speak, is an important technical system of computing science. The thought formalization has pioneered and developed computer science.

C. Optimization

In table 2, the concepts such as redundancy, complexity and optimization reflect the optimization ideas from different perspectives. Jobs originally done by human beings are finished by a computer with programs or software, so as to improve the work efficiency, it itself is optimization. Moreover, optimization thought has further directly changed the programs themselves, and improved the efficiency of programs' running. In the earliest programming, before the concept 'complexity' was well understood, people had already learnt lowering the time complexity by reducing the number of loops. Later, optimization ideas had come up to improve resource utilization. From 1st normal form (1NF) to 5th normal form (5NF) of the database theory, scientists have been managing to reduce data redundancy and save storage space, to make the data redundancy of a database lower and lower. After that, optimization has gradually changed the initial requirement for people into the necessary function of a system. For example, reducing complexity, reducing redundancy, choosing data type appropriately and so on once were the requirement for designers, with more and more optimization rules and requirements, a lot of software and hardware have been able to automatically implement some related performance. As a result, self-checking, system's fault tolerance and error correcting technology embody the thought of system's self-perfection; Varying-length string and variant data type reflect 'flexible' and 'adaptive'

thought. Their essence is also optimization. Later on, optimization had extended from the connotation or essence to the form or appearance gradually. As an instance for this, software's interface has developed from black and white character interface into colored GUI; nowadays, optimization has made static and monotonous figures become the dynamic and visual data... In a word, optimization is the basic thought of computing science; it also has led the research and development of the discipline.

D. Association and Interaction

Hyperlink in a web page embodies the association and interaction of files, media and location of information; Object-oriented programming embodies the association and interaction of objects and data; Transaction management of database(DB) theory embodies the association and interaction of events; By means of locks, a database achieves the concurrency control of the system; Referential integrity realizes the data's association and interaction, and some software also provide pointer's association and interaction.

Association-and-interaction is integrality. It is also correlation. It is helpful to improve the access speed, realize automation and ensure the validity of data. Analysis shows, 'binding' of View 1 in table 1 is also association-and-interaction. From human-computer interaction in OOP and GUI to wired communication of person-person by computers' software and hardware on the Internet, and then to wireless link and interaction, such as human-machine, machine-machine, human-matter and matter-matter on the Internet of things... The forms and contents of association-and-interaction are being constantly renovated; its idea is changing the world. The core thought of it is searching for a reliable interconnected mechanism to realize the integration and multi-win. People all over the world are benefiting from the thought.

E. Reuse and Sharing

Database technology makes data files separate themselves from the source program and exist independently; therefore they can be applied to other programs. Most importantly, this is helpful to be used by many users simultaneously. Wider coverage of Internet has greatly promoted resource sharing. The clipboard of Windows, the Cut, Copy and Paste operation of various software and the like are all helpful to reuse the data or the files. As to Backup and Restore of a database, although the direct aim is not at reuse or sharing, it is the maximum reuse or sharing when a hardware or software disaster happens, for without backup, restore would be impossible. So reuse or sharing is one of the most important thoughts of computing science. It is meaningful to reduce the repetitive development of resources or to improve resource utilization. The thought can be widely applied to many social or economic fields.

V. ESSENCE OF CT: FORMALIZATION DESCRIPTION OF IT

To explain the essence of CT, describing it in formal language is essential. CT concerns many branches of computing science such as data, algorithm and program and so on. It should be the union set of CT of every branch. The Computational Thinking of a branch is the function of the fundamental concepts, especially the representative terms. It must be noticed, methods and tools have their thinking connotations too. Here, the methods mean the steps or procedures by which computing science deals with problems. For Example, reuse can be realized through two steps—copy and paste; and clipboard is the tool. Template is also a tool for reuse. Analysis shows, command, icon and button are tools as well. These methods and tools are the materialization of the representative terms. With the representative terms only and without these methods and tools, a software or a computer cannot work, CT can not be perceived by people easily either. As some scholars pointed out, every tool is a component of a certain method[28]; each method is the combination of several job steps which need to be finished under the guidance of certain thinking and by means of some tools[29] [30]; a method is a pattern of outward manifestations of certain thinking[31]. So the idiomatic tools and methods of computing discipline are the important constituent parts of CT as well. Now, CT can be described in mathematical language as follows:

$$CT = CT_d \cup CT_a \cup CT_p \cup CT_{so} \cup CT_{sy} \cup CT_n \quad (1)$$

$$CT_x = F_x(Rt_x, M_x, T_x) \quad (2)$$

In equation(1) and equation(2), CT_x refers to the Computational Thinking of a branch, and Rt_x refers to the representative terms of the corresponding branch, $x \in \{\text{data, algorithm, program, software, system, network}\}$. F_x refers to the Computational Thinking function of the branch. M_x refers to the methods and T_x to the tools. A conceptual model of CT can be concluded as figure 1.

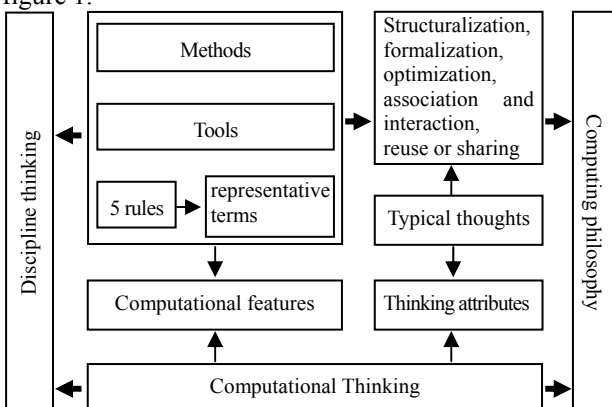


Figure 1. Conceptual Model of CT

VI. CONCLUSIONS

CT is the basic principle of computing science. To some extent, CT has been 'interred beneath layers of technology in our understanding and our teaching'. [32] The key to understand CT is grasping its computational

features and its thinking attributes. The representative terms are the carrier of CT. The computational features and thinking attribute are included in the representative terms. Structuralization, formalization, optimization, association-and-interaction and reuse are the typical thoughts of CT. In general, structuralization, formalization, association-and-interaction are the means or approaches; optimization, reuse and sharing are the goals. Training CT must start from understanding and applying the representative terms, the methods and the tools. In teaching computing science, after the meanings and usages of the terms are expounded, the concepts' ideological connotation and universality must be emphasized. When teaching algorithms, resting on their classical cases and instrumental meanings is inadvisable. The most important is clarifying their methodological meanings and realistic significances, and practicing them in everyday experiences.

REFERENCES

- [1] President's Information Technology Advisory Committee, *Computational Science: Ensuring America's Competitiveness*, <http://www.nitrd.gov/pitac/reports/200506/computational.pdf>, June 2005.
- [2] Chinese Academy of Sciences, *Information Science & Technology in China: A Roadmap to 2050*, Science Press, China, 2011.
- [3] Howard Gardner, *The Disciplined Mind: Beyond Facts and Standardized Tests*, Penguin Group (USA) Incorporated, USA, 2000.
- [4] Terrell Ward Bynum, James H. Moor, *The Digital Phoenix: How Computers Are Changing Philosophy*, Springer, USA, 1998.
- [5] Hao Ningxiang, "Computing: A New Philosophy Area", *Philosophical Trends*, vol.12, no.11, pp.32-36, 2000.
- [6] Richard Karp, "Understanding Science through the Computational Lens", *Journal of Computer Science and Technology*, vol.26, no.4, pp.66-68, 2011.
- [7] Edsger Wybe Dijkstra, "How Do We Tell Truths That Might Hurt?" *Selected Writings on Computing: A Personal Perspective*, Springer Verlag New York Inc., USA, 1982.
- [8] Huang Chongfu, "Applying Information Diffusion Principle and Computational Thinking Machine in earthquake engineering", *Beijing Normal University Press, China*, 1992.
- [9] Dong Rongsheng, Gu Tianlong, "Methodology of Computer Science and Technology", *Post & Telecom Press, China*, 2002.
- [10] Jeannette M. Wing, "Computational Thinking", *Communications of the ACM*, vol.22, no.9, pp. 23-28, 2006.
- [11] Jan Cuny, Larry Snyder, Jeannette M. Wing, "Demystifying CT for Non-Computer Scientists", *Work in Progress*, pp.246-249, 2010.
- [12] Steering Committee of Computer Science and Technology of China Ministry of Education, *On the composition and cultivation of the professional ability of the talents of computer science and technology in institutions of higher learning*, China Machine Press, China, 2010.
- [13] ISTE, *Operational Definition of Computational Thinking for K-12 Education*, http://www.iste.org/Libraries/PDFs/Operational_Definition_of_CT_sflb.ashx, 2012.
- [14] Einstein, *Collected Works of Einstein*, Science Press, China, 1989.
- [15] Jeannette M. Wing, "Computational Thinking", *Communi-*

- nications of the ACM*, vol.10, no.9, pp.23-28, 2006.
- [16] Alan Bundy. "Computational Thinking Is Pervasive", *Journal of science and practical computing*, vol.1, no.2, pp.67-69, 2007.
 - [17] Committee for the Workshops on Computational Thinking, *National Research Council, Report of A Workshop on the Scope and Nature of Computational Thinking*, The National Academies Press, USA, 2010.
 - [18] Li Lian. "On the Concepts and Challenges of Computational Thinking", *Chinese College's Teaching*, vol.6, no.1, pp.7-12, 2012.
 - [19] Tan Haoqiang, "Researching Computational Thinking Must Face Its application", *Computers Education*, vol.15, no.21, pp. 45-50, 2012.
 - [20] Computer Teaching Experimental Center of Xi'an Jiaotong University, "Questionnaire for College Computer Basic Course", pp.3-6, 2009.
 - [21] IEEE/ACM. *Computing Curricula 2004*, SIGCSE Norfolk, USA, 2006.
 - [22] IEEE/ACM. *Computing Curricula 2004*, SIGCSE Norfolk, USA, 2006.
 - [23] Jeannette M. Wing. "Computational Thinking and Thinking about Computing", *Philosophical Transactions of the Royal Society*, vol.1, no.3, pp.37-42, 2008.
 - [24] Pat Philips. "Computational Thinking: a problem-solving tool for every classroom", *Communications of the CSTA*, vol.3, no.6, pp.12-16, 2009.
 - [25] Wu Wenhui, Wang Jiande, *Advanced Tutorial for International Academician Programming Competition (ACM/ICPC) (Book 1): The Common Computational Thinking Ways in Programming*, China Railway Publishing House, China, 2011.
 - [26] He Mingxin, "Methodological Significance of the Separation of Concerns for software Engineering", *Computer Science*, vol.12, no.4, pp. 60-63, 2011.
 - [27] Zhao Lingzhong, "On Strategy of Algorithm Design and Computational Thinking", *Enterprise Technology and Development*, vol.8, no.8, pp.35-38, 2011.
 - [28] Cihan Varol and Coskun Bayrak, "Risk Assessment of Error-Prone Personal Information in Data Quality Tools", *Journal of computer*, vol.6, no.2, pp.155-164, 2011.
 - [29] Daniel Amyot and Gunter Mussbacher, "User Requirements Notation: The First Ten Years, The Next Ten Years", *Journal of Software*, vol.7, no.3, pp.747-758, 2012.
 - [30] Tyler C. McCandless, Sue Ellen Haupt, and George S. Young, "The Effects of Imputing Missing Data on Ensemble Temperature Forecasts", *Journal of computer*, vol.6, no.2, pp.162-171, 2011.
 - [31] Kasom Koht-arsa and Surasak Sanguanpong, "A Centralized State Repository Approach to Highly Scalable and High-Availability Parallel Firewall", *Journal of computer*, vol.8, no.7, pp.1664-1677, 2013.
 - [32] Peter J. Denning, "Great Principles of Computing", *Communications of the ACM*, vol. 46, no. 11, pp.15-20, 2003.
- Wenchong Shi** received a bachelor degree in mining from China University of Mining and Technology in 1992, a master degree in computer application from Yanshan University in 1998 and Ph.D degree in operational research and cybernetics from Hebei University of Technology in 2003.
- He is an associate professor of computer science at Hebei Normal University of Science and Technology. His research interests include social computing, social multimedia computing and software engineering. He has published over 30 research articles in data or image processing.
- Dr. Wenchong Shi is a member of China Computer Federation and ACM.
- Maohua Liu** received a master degree in computer science from Tianjin University in 2002 and a Ph.D degree in operational research and cybernetics from Hebei University of Technology in 2005.
- He is an associate professor of computer science at Hebei Normal University of Science and Technology. His research interests include social computing, social multimedia computing and software engineering. He has published over 20 research articles in software engineering.
- Dr. Maohua Liu is a member of ACM.
- Peter Hendler** received his M.Sc. (1998) and Ph.D. (2002) degrees in computer science from the University of Ottawa. After working for Mitel Networks as a senior researcher, he joined the School of Electrical Engineering and Computer Science of the University of Ottawa, where he is now Associate Professor in software engineering. His research interests include software engineering, requirements engineering, aspect-oriented modeling, and healthcare informatics. He has published over 20 papers in various conferences and in journals.
- Peter Hendler is a member of ACM.