July 4–8, 2015
Vilnius, Lithuania

Association for
Computing Machinery

Advancing Computing as a Science & Profession

ITiCSE-WGP'15

Proceedings of the 2015 ITiCSE Conference on

Working Group Reports

Sponsored by:

ACM SIGCSE

Supported by:

Bilkent University

Association for
Computing Machinery

Advancing Computing as a Science & Profession

The Association for Computing Machinery
2 Penn Plaza, Suite 701
New York, New York 10121-0701

ISBN: 978-1-4503-4146-2

Additional copies may be ordered prepaid from:

ACM Order Department
PO Box 30777
New York, NY 10087-0777, USA

Phone: 1-800-342-6626 (USA and Canada)
+1-212-626-0500 (Global)
Fax: +1-212-944-1318
E-mail: acmhelp@acm.org
Hours of Operation: 8:30 am – 4:30 pm ET

Printed in the USA

Preface to the Working Group Reports of the 2015 Innovation & Technology in Computer Science Education Conference

Each year, the ITiCSE conference Call for Participation (CFP) includes a call for working group proposals, which are a unique opportunity for computing educators, from different countries, to come together and collaborate on a topic of common interest.

For ITiCSE 2015, the CFP invited any intending group leader(s) to submit a two-page proposal. Those proposals were evaluated by the working group coordinators, on the basis of likely interest in the suggested topic and the qualifications of the intending leaders. The decisions were made jointly with the conference chair and the working group coordinators. This year, nine applications were received, and seven of those proposals were accepted.

After the registration for working groups was closed, the working groups coordinators asked the working groups leaders to start working on their topic well before the conference. The groups arrived the conference venue couple of days before the conference started and continued working throughout the ITiCSE conference. Therefore the working groups had five days to work face to face on their projects. At the end of the conference, each working group was asked to submit a mature report to the coordinators.

After the conference, the working groups had approximately month and a half to finalize their report. At the end of August, all the seven working groups submitted reports for peer review. Each of the working group reports was evaluated by three or four referees. On the basis of this reviewing process, all seven working group reports were accepted (with revisions) for publication. The finalized reports where submitted accompanied by a letter that reflected the changes that where done according to the reviewers' comments, and were reviewed again by the working group coordinators.

As ITiCSE 2015 approached, we had some working group members indicate that they would not be attending the conference, but they wanted to remain members of their group. As per the philosophy of the ITiCSE conferences and the direction of SIGCSE we did not agree to those people remaining members. For the ITiCSE conference when people propose a working group, or apply to join a group, it must be with the intention of coming to the conference. Intrinsic to any conference is the concept of meeting face to face.

The concept of an ITiCSE working group has evolved, and there is now a great deal of tacit knowledge on how these groups are run, managed, and coordinated. We followed previous recommendation that at least one of the working groups leaders has prior experience as a working group member. Two of the groups exceeded the number of ten participants (one of 13, and one of 17) that was a fine limit in our opinion. Those working group leaders insisted on keeping that size and by that to expand the international knowledge gain. After submitting the final versions, those two working groups' leaders reported that all the participants contributed to the work, and that it was hard to manage a group of this size.

In closing, we would like to thank the 2015 Working Group leaders and all participants for their dedication, commitment and enthusiasm to their research and final reports. We would also like to thank the ITiCSE 2015 committee for their help in organizing the working groups, especially those committee members responsible for providing the necessary rooms and facilities at the conference venue.

Noa Ragonis
Beit Berl College and Technion-ITT, Israel

Päivi Kinnunen
Aalto University, Finland

Table of Contents

Working Group Presentations

- **Challenges and Recommendations for the Design and Conduct of Global Software Engineering Courses: A Systematic Review** 1
  Tony Clear *(Auckland University of Technology)*, Sarah Beecham *(University of Limerick)*, John Barr *(Ithaca College)*, Mats Daniels *(Uppsala University)*, Roger McDermott *(Robert Gordon University)*, Michael Oudshoorn *(Wentworth Institute of Technology)*, Airina Savickaite *(Vilnius University)*, John Noll *(University of Limerick)*
- **Educational Data Mining and Learning Analytics in Programming: Literature Review and Case Studies** 41
  Petri Ihantola *(Tampere University of Technology)*, Arto Vihavainen *(University of Helsinki)*, Alireza Ahadi *(University of Technology, Sydney)*, Matthew Butler *(Monash University)*, Jürgen Börstler *(Blekinge Institute of Technology)*, Stephen H. Edwards *(Virginia Tech)*, Essi Isohanni *(Tampere University of Technology)*, Ari Korhonen *(Aalto University)*, Andrew Petersen *(University of Toronto Mississauga)*, Kelly Rivers *(Carnegie Mellon University)*, Miguel Ángel Rubio *(University of Granada)*, Judy Sheard *(Monash University)*, Bronius Skupas *(Vilnius University)*, Jaime Spacco *(Knox College)*, Claudia Szabo *(The University of Adelaide)*, Daniel Toll *(Linnaeus University)*
- **A Global Snapshot of Computer Science Education in K-12 Schools** 65
  Peter Hubwieser *(Technical University of Munich)*, Michail N. Giannakos *(Norwegian University of Science and Technology)*, Marc Berges *(Technical University of Munich)*, Torsten Brinda *(University of Duisburg-Essen)*, Ira Diethelm *(Carl von Ossietzky Universität Oldenburg)*, Johannes Magenheim *(Paderborn University)*, Yogendra Pal *(Indian Institute of Technology Bombay)*, Jana Jackova *(Matej Bel University)*, Egle Jasute *(Vilnius University)*
- **Concepts in K–9 Computer Science Education** 85
  Erik Barendsen *(Radboud University)*, Linda Mannila *(Åbo Akademi University)*, Barbara Demo *(University of Turin)*, Nataša Grgurina *(University of Groningen)*, Cruz Izu *(University of Adelaide)*, Claudio Mirolo *(University of Udine)*, Sue Sentance *(King's College London)*, Amber Settle *(DePaul University)*, Gabrielė Stupurienė *(Vilnius University)*
- **New Horizons in the Assessment of Computer Science at School and Beyond: Leveraging on the ViVA Platform** 117
  Daniela Giordano *(University of Catania)*, Francesco Maiorana *(University of Catania)*, Andrew Csizmadia *(Newman University)*, Simon Marsden *(University of Portsmouth)*, Charles Riedesel *(University of Nebraska)*, Shitanshu Mishra *(Indian Institute of Technology Bombay)*, Lina Vinikienė *(University of Vilnius)*
- **Multinational Perspectives on Information Technology from Academia and Industry** 149
  Mihaela Sabin *(University of New Hampshire)*, John Impagliazzo *(Hofstra University)*, Hala Alrumaih *(Al Imam Mohammad Ibn Saud Islamic University)*, Brenda Byers *(Canada's Association of IT Professionals)*, Daina Gudoniene *(Vilnius University)*, Margaret Hamilton *(RMIT University)*, Vsevolod Kotlyarov *(Peter the Great Polytechnic University)*, Barry Lunt *(Brigham Young University)*, James W. McGuffee *(Northern Kentucky University)*, Svetlana Peltsverger *(Kennesaw State University)*, Cara Tang *(Portland Community College)*, Barbara Viola *(Viotech Solutions Inc.)*, Ming Zhang *(Peking University)*
- **What's in a Name? International Interpretations of Computing Education Terminology** 173
  Simon *(University of Newcastle)*, Alison Clear *(Eastern Institute of Technology)*, Janet Carter *(University of Kent)*, Gerry Cross *(Mount Royal University)*, Atanas Radenski *(Chapman University)*, Liviana Tudor *(Petroleum-Gas University of Ploiesti)*, Eno Tõnisson *(University of Tartu)*

Author Index 187

A Global Snapshot of Computer Science Education in K-12 Schools

Peter Hubwieser
Technical University of Munich
School of Education
Arcisstrasse 21,
80333 München, Germany
+49 89 289 29110
peter.hubwieser@tum.de

Michail N. Giannakos
Norwegian University of Science and Technology, Department of Computer & Information Science
Sem Saelands vei 9,
Trondheim, NO-7491, Norway
+47 73590731
michailg@idi.ntnu.no

Marc Berges
Technical University of Munich
School of Education
Arcisstrasse 21,
80333 München, Germany
+49 (89) 289 29112
berges@tum.de

Torsten Brinda
University of Duisburg-Essen, Faculty of Business Administration and Economics, Schützenbahn 70,
45127 Essen, Germany
+49 201 183 7248
torsten.brinda@uni-due.de

Ira Diethelm
Carl von Ossietzky Universität Oldenburg
Department of Computing Science
26111 Oldenburg , Germany
+49 441 798 2990
ira.diethelm@uni-oldenburg.de

Johannes Magenheim
Paderborn University, Faculty f or Electrical Engineering, Computer Science, and Mathematics
Fürstenallee 11,
33102 Paderborn, Germany
+49 5251 60-6341
jsm@uni-paderborn.de

Yogendra Pal
Indian Institute of Technology Bombay
Educational Technology IDP
Mumbai, 400076, India
+91 (22) 2572 2545
yogendra.pal3@gmail.com

Jana Jackova
Matej Bel University
Department of Informatics
Národná 12,
974 01 Banská Bystrica, Slovakia
+421 48 446 7128
jana.jackova@umb.sk

Egle Jasute
Vilnius University, Faculty of Mathematics and Informatics
Akademijos str. 4,
LT-08663 Vilnius, Lithuania
+370 5 219 3050
egle.jasute@gmail.com

ABSTRACT

In two special issues of the ACM journal “Transactions on Computing Education” (TOCE), 14 extensive case studies about the various situations of Computer Science Education (CSE) in K-12 schools in 12 countries (respectively states) were collected. During the work at the ITiCSE 2015, we have performed a deductive qualitative text analysis on these case studies in order to extract the most useful information. As a category system, we applied some selected categories of the Darmstadt Model that was developed by the working group “Computer Science/Informatics in Secondary Schools” at the ITiCSE 2011. Based on the coding results, we summarized information about the different fields of Computing Education at schools, the intended goals and competencies, the taught content, the applied programming languages and tools and the different forms of assessment and teacher education. Despite the limitations of the analyzed articles, representing just snapshots of complex situations from the specific viewpoint of the respective authors, we were able to collect some interesting results.

CCS Concepts
• **Social and professional topics**→Computer science education

Keywords
Computing Education; competencies; teacher education.

1. INTRODUCTION

While Israel [14] and many eastern-European countries [40], [10] were offering rigorous CS courses at schools for several decades already, several other countries have shifted the focus of Computer Science Education (CSE) in primary and secondary schools (shortly K-12) from computer and ICT applications towards rigorous academic computing just recently, see e.g. [4], [7], [20].

Several initiatives and projects were launched to foster this change. For example in the US, the activities of the CSTA yielded quite ambitious standards in 2011 [38] and an interesting comparison of CSE over the 50 states in 2010 [47]. The 10K Initiative aims at educating a substantial number of formally educated CS teachers, [13], while the recent approaches of “Computational Thinking” [48], “CS Principles” [2] and “Exploring CS” [16] aim at fostering the consolidation of K-12 CSE. Simultaneously, the British Royal Society has published its revealing report “[43] “Shutdown or Restart”, which has inspired the broad and promising initiative “Computing at Schools” [7], while in New Zealand brand new computer science standards for schools were introduced from 2011 to 2013 [4].

While these numerous and different initiatives produce many interesting ideas and amazing outcomes, the respective publications, distributed over several conference proceedings or journals, are not easy to compare. Due to the comparably short paper format of most of the proceedings, the majority of these papers almost provide only snippets of the respective situation. Therefore, two special issues of the ACM TOCE journal ("Transactions on Computing Education") were dedicated to Computing Education in K-12 schools, aiming to collect a number of extensive case studies, see [31], [30]. Additionally, a pilot study was published in advance [20]. In summary, all these case studies cover about 350 pages of text describing the regarding situation in 12 countries or states all over the world.

For our work during the ITiCSE 2015 in Vilnius, we intended to analyze and summarize all this information, considering the "most important" aspects for CSE in K12 schools. After a close discussion of potential research questions (see Section 4), we analyzed this text corpus regarding the fields of Computing Education at schools (e.g. ICT, IT, CSE), the intended goals and competencies, the taught content, the applied programming languages and tools, the different forms of assessment and the different forms of teacher education. This paper presents the outcomes of this analysis. Our results could be used, for example, by national stakeholders arguing in favor of a subject of CS, by curriculum designers who have to decide which approach a coming national initiative should follow, by researchers developing a framework for further studies about CSE in primary and secondary schools, or by teacher educators as a 'look over the fence'.

Nevertheless, we are well aware that the results of our analysis are based on case studies that do not necessarily represent the complex situation of CSE in these specific countries, particularly in federal states like US. Furthermore, the case studies represent the specific views of their authors that don't always represent the real situation. Additionally, things are rapidly changing in CSE, thus the current situation may have changed already when this report is published.

2. BACKGROUND

As already mentioned, we have used the 14 case studies of the two special issues of TOCE on CSE at K-12 schools (and the pilot article for these issues [20]) as text corpus for our analysis. In the call of the special issues, the authors were asked to write their studies according to the categories of the *Darmstadt Model* (DM), which had been developed by a working group *Informatics in Secondary Education* (WG ISE) at the ITiCSE 2011 [22]. Additionally, we used this model to generate the research questions for our work (see Section 4). In consequence, the Darmstadt Model represents the focal theoretical background for our work. Additionally, the WG report on the DM [22] provides a lot of references and explanations respecting the categories of the model. Therefore, if some reader may miss any theoretical background for our categories, she or he may have a look in the WG report on the DM [22].

The DM was developed starting from the so-called *Berlin Model* [19] (in English see [45]), which intends to describe the process of lesson planning by the following categories:

1) *Anthropogenic* or *Socio-cultural Preconditions*
2) *Decision Areas: Intentions, Content, Learning and Teaching Methods, Media* and
3) *Anthropogenic* or *Socio-cultural Consequences.*

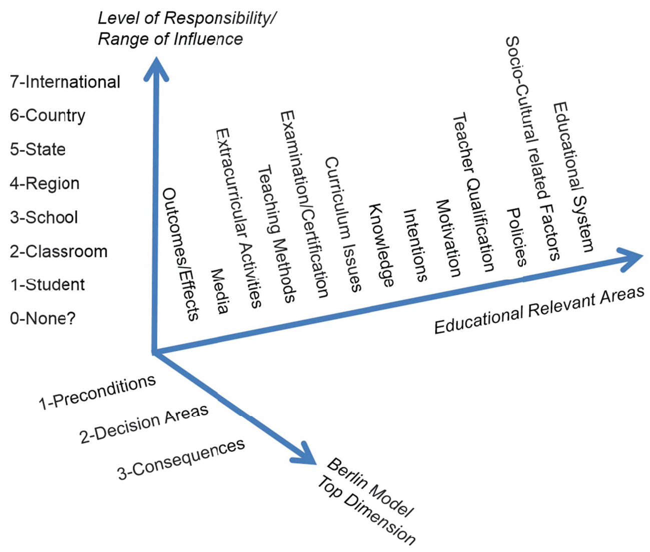
Based on this model, the WG ISE had performed a qualitative text analysis of five selected case studies about CS in secondary education. It turned out that this model was not sufficient for the description in several aspects. The central issue was that the top level categorization of the Berlin model turned out to be strongly dependent on the influence range of the relevant persons (e.g. the authors). Therefore, the WG ISE introduced two additional dimensions: *Berlin Model Top Category* and *Level of Responsibility/Range of Influence.* Further, some relevant and important categories were missing (e.g. curriculum issues) in the Berlin Model and thus were added to the new model. This resulted in a three-dimensional category system, which was named *Darmstadt Model* in honor of the location of the ITiCSE conference 2011:

– Dimension 1 (*Level of Responsibility/Range of Influence*) determines the decision level of the regarded stakeholders. According to the position of the respective person in the school-system, the following subcategories are suggested: *Student/Pupil, Classroom, School, Region, State, Country,* and *International.*

– Dimension 2 (*Berlin Model Top Category*) is formed by the categories of the first level of the original Berlin Model: *Preconditions, Decision Areas* and *Consequences*.

– Dimension 3 (*Educational Relevant Areas*) describes issues that are directly relevant for educational activities. It comprises the remaining subcategories of the original Berlin Model that have turned out to be relevant in our context (e.g. *Intentions*) and additionally several other categories that had emerged during coding (e.g. *Educational System*). In Figure 1 all categories of this dimension are displayed.

The genesis and structure of the model was described in detail in the respective working group report of the ITiCSE 2011 [22]. After some slight adoptions, the current version of model was presented in [21].



1.Figure 1. The Darmstadt Model

3. RELATED WORK

In 2012, the British Royal Society (BRS) published its groundbreaking report *Shutdown or Restart* [43]. This report raised important issues on K-12-CSE, caused significant political awareness, and most importantly, lead to initiatives of reintroducing Computing in schools with rigorous contents of Computer Science (CS).

The *Computer Science Teachers Association* (CSTA) of the ACM is one of the most active stakeholders of K-12-CSE. Besides its curricula, it published two flagship articles about the state of K-12-CSE in the USA. The report *The New Educational Imperative* [39] provides a comprehensive look at high school CSE in the USA and around the world. The well-known study *Running on Empty* [47] summarizes the state of K-12-CSE in USA by examining and mapping current learning standards in core subject areas. The study revealed that roughly two-thirds of the states have CSE standards for secondary education, with few and scattered courses dedicated to CSE.

A working group at the KOLI conference in November 2011 investigated the international situation of K-12-CSE. The group conducted a survey among experts that reflected on their national state of CSE in school and of computer science teacher education [36].

In 2013, a joint working group from Informatics Europe & ACM Europe, consisted of experts from academia and industry, published its report *Informatics education: Europe cannot afford to miss the boat* [24]. The report claims to "build on the considerable body of educational research and experimentation on digital literacy and informatics education developed over the past decades in Europe, the US and elsewhere". It summarizes the situation of K-12-CSE in several countries and gives some comprehensive recommendations.

In addition, several shorter relevant case studies were published recently in the proceedings of the ISSEP conferences ("Informatics in Schools: Situation, Evolution, and Perspectives"), see e.g. [17], [11], for example about Poland [41], France [44] and Venezuela [12]. Some of these were partly based on the DM.

4. RESEARCH QUESTIONS

To systematize the evaluation process, we considered all the research questions that were raised in the editorial of our second special issue of the TOCE journal [23]. After a survey among the participants of the working group about their research intentions, we decided to focus our evaluation on the questions that seemed to us particularly interesting and promising respecting the content of the case studies:

(1) Which terms are used to describe the *different fields of computing education at schools* (ICT, CSE, Computing, etc.)? How are these terms understood in the context of the different case studies?

(2) Which *goals* and which *competencies* are intended in K12-CS Education?

(3) Which *learning content* is delivered in K12-Education?

(4) Which *programming languages* and *tools* are used in K-12 schools?

(5) Which *assessment forms* for students are applied in the different countries or states?

(6) Which level of *teacher education* is required in K-12 CSE? What types of additional teacher training are provided in K-12 CSE?

According to our opinion, the answers on these questions could provide a rather comprehensive overview of the case studies, because this set of questions covers three of the four *Decision Areas* of the *Berlin Model* (see Section 2): *Intentions* by question 2, goals *Content* by question 3 and *Media* by question 4. The missing of the fourth *Decision Area* (*Learning and Teaching Methods*) could be justified by the argument that this is a more general, not very subject-specific field.

Please note that Section 8 on the results of our work is structured according to these 6 questions.

5. THE TEXT CORPUS

As already explained in the introduction, our text corpus consisted of the 14 articles that had been published in the two special issues on CSE in schools of the ACM TOCE journal [31], [30] and the pilot article for these issues [20]. Table 1 displays the resulting list of papers including the references and their number of pages, ordered by their length.

Table 1. The case studies of the special issues of TOCE

Code	Country/State	Reference	Pages
BY	Germany/ Bavaria	[20]	41
SGD	USA (SGD)	[33]	31
IN	India	[32]	36
NZ	New Zealand	[5]	31
NRW	Germany/ NRW	[27]	22
GEO	USA/ Georgia	[18]	29
FR	France	[3]	27
KO	Korea	[9]	22
SW	Sweden	[34]	25
UK	UK	[8]	22
FIN	Finland	[28]	18
US-IS	USA & Israel	[15]	18
RUS	Russia	[26]	10
IT	Italy	[6]	6

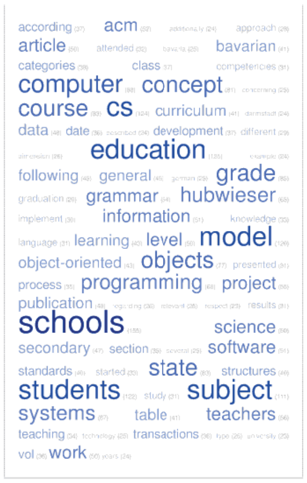
In summary, these 14 papers provide rich information about the situation of CSE in K12 School in 12 countries or states on 338 pages of text. In principle, every paper describes the situation in a certain country or a state like Bavaria or North Rhine-Westphalia (NRW) in Germany.

Yet, three papers have a specific focus. The paper GEO [18] informs about the goals, content and outcomes of a six-year *project named Georgia Computes!* aiming to improve computing education across this US-state. The paper SGD [33] presents the extracurricular project *Scalable Game Design* that is based on specific software systems and has impact in several US-states. The paper [15] compares the situation in the USA and Israel. In consequence, we have coded the latter in two different directions, separating the information about these two countries. Nevertheless, as the educational systems in the US are very different from state to state, we refrain from drawing any conclusions about the whole country from these three papers. Exceptions are several conclusions about the pure occurrence of an aspect anywhere in these states, e.g. regarding the use of a specific term in the context of computing at schools, of a certain programming language or of a certain goal for computing at schools. Nevertheless, the reader of this report should keep in mind that the analyzed information about the USA was restricted to these three very specific papers, and thus is far from being complete in any sense.

Although UK paper [8] tries to catch situation in the whole country, where England, Scotland, Wales and Northern Ireland have their own education systems, the authors refer mainly to England/Wales (their education systems "are broadly similar" [8, Section 4, p. 6]).

Though most of the analyzed papers have a quite similar conception as case studies, their scope and direction are quite different. In

Figure 2, the different word clouds of the papers BY [20] and IT [6] may give an impression of these differences. While the Bavarian paper is emphasizing concepts, modeling, object-orientation and a (compulsory) subject of CS, the Italian is focused on Informatics (understood in the sense of ICT education there) and technology.



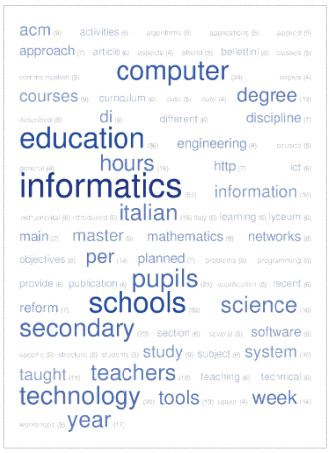


Figure 2. Word clouds of papers BY and IT

6. CODING PROCESS

For the qualitative text analysis of the case studies, the methodology of Mayring [29] was chosen, who had combined several techniques for systematic text analysis to a very elaborated and systematic research process.

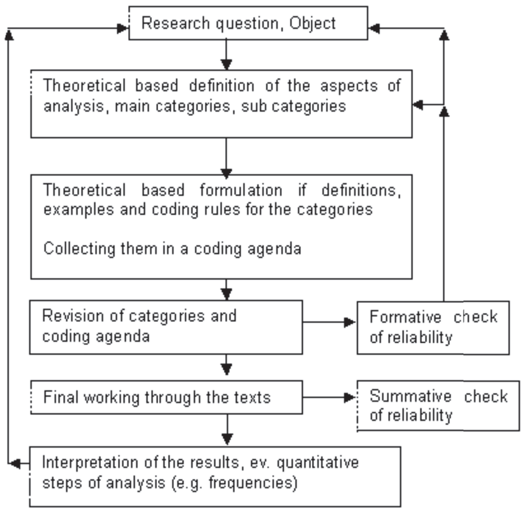
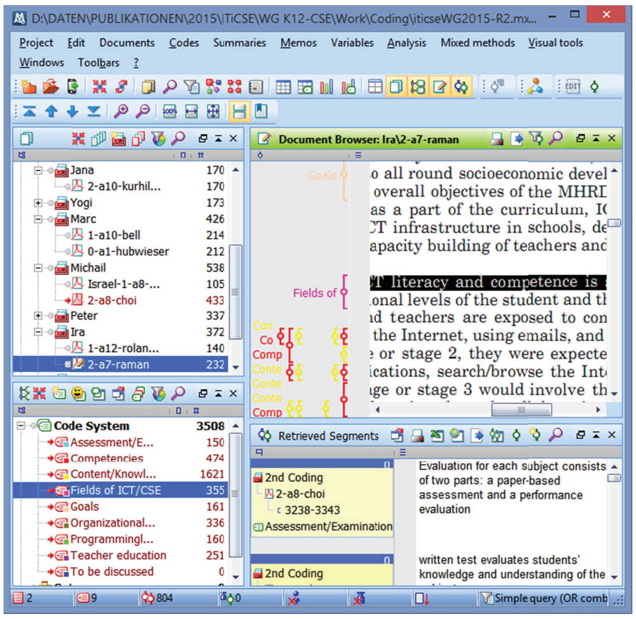

Research question, Object
Theoretical based definition of the aspects of analysis, main categories, sub categories
Theoretical based formulation if definitions, examples and coding rules for the categories
Collecting them in a coding agenda
Revision of categories and coding agenda
Formative check of reliability
Final working through the texts
Summative check of reliability
Interpretation of the results, ev. quantitative steps of analysis (e.g. frequencies)

Figure 3. Mayring's step model of deductive category application [29].

According to Mayring, the category system can be either derived deductively from a suitable existing theory or developed inductively from the text corpus during the analysis. The first strategy incorporates also the revision of the existing category system, as displayed in Figure 3. For our purpose, we derived the category system from our research questions (see Section 4).

For computer support, we chose the coding software *MaxQDA* (www.maxqda.com), which provides elaborated features for coding, integrating results from different coders and analysis of the results. Figure 4 shows the results of our coding work in the MaxQDA window. The screenshot gives an impression, how the coding process was performed from a technical point of view. In particular, we used one common MayQDA file, which contained all coding results of all group members on all documents (see upper left section of the window). This file was updated after each coding step. By this way, we could easily provide an overview of all codings of the same category for all coders (upper right part of the window)



Jana	170
2-a10-kurhil...	170
Yogi	173
Marc	426
1-a10-bell	214
0-a1-hubwieser	212
Michail	538
Israel-1-a8-...	105
2-a8-choi	433
Peter	337
Ira	372
1-a12-rolan...	140
2-a7-raman	232

Code System	3508
Assessment/E...	150
Competencies	474
Content/Knowl...	1621
Fields of ICT/CSE	355
Goals	161
Organizational...	336
Programmingl...	160
Teacher education	251
To be discussed	0

o all round socioeconomic devel
overall objectives of the MHRI
as a part of the curriculum, I(
CT infrastructure in schools, de
apacity building of teachers and

T literacy and competence is
onal levels of the student and tl
nd teachers are exposed to con
the Internet, using emails, and
e or stage 2, they were expecte
ications, search/browse the Int
ge or stage 3 would involve th

2nd Coding
2-a8-choi
3238-3343
Assessment/Examination

Evaluation for each subject consists of two parts: a paper-based assessment and a performance evaluation

2nd Coding

written test evaluates students' knowledge and understanding of the

2 9 804 0

Figure 4. Screenshot of the applied coding software MaxQDA.

Regarding the coding strategy, we have started with coding four papers in pairs. As soon as we realized that everybody was applying the categories quite similarly, we switched to single-person coding. At the end of this first coding run, we have coded 2284 relevant text passages over all categories. Interestingly, all the case studies were complete in the sense that we coded all categories in every document.

Table 2 displays the statistics of the codings of the first coding run. The second column displays the number of coded text segments over all documents. Obviously, the coding frequency is varying substantially over the categories. *Content* produced by far the most codings (1053). Yet, taking the length of the codings into account, it turned out that these codings were quite short in average (27 characters), while those of *Teacher education* were more than 10 times longer (298 characters in average).

This revealed that the different coding frequencies between these two categories were mainly caused by their granularity. In the first case mostly words (out of content lists) were coded, while in the second case mostly sentences or even paragraphs were found. Assuming that the complexity of the information represented by the coded segments is growing with their length, this means that the coded information on *Teacher education* was far more complex compared to *Content*. Out of this reason, we decided to choose different forms for the presentation of the respective results, see Section 8.3. and 8.6. If we regard the total sum of coded characters as a measure for the information richness, Clearly, *Teacher education* is the leading category. On the other end, the information on Programming languages or -tools is the sparest, restricting on the pure detection the usage of any language or tool in the respective country or state.

Table 2. Coding results for the analysis of the case studies

Code	Number	Average Length*	Total Length**
Teacher education	170	298	49154
Organizational Aspects	247	140	33913
Content/Knowledge	1053	27	28543
Fields of ICT/CSE	174	165	28073
Competencies	308	71	21572
Assessment/ Examination	98	176	16159
Goals	127	106	13199
Programming languages/-tools	107	15	1582

* number of coded characters including spaces
** sum of coding length over all codings

To assess intercoder reliability, we chose four papers that seemed to form the most "representative" subset (FR, IT, KO and NZ, representing nearly 30% of all papers), to be coded in a second run by a different researcher. As the working time during the conference was limited, we had to proceed in the evaluation of the coding work. Thus we decided that some members of the group should perform the analysis of the coding results based on the codings of the first run, while other members should recode the selected documents. As we intended mainly to detect statements that referred to our categories in the papers and not to draw any quantitative conclusions, we regarded this strategy as acceptable. At the end of the second coding process, we had coded 1224 text snippets of the four recoded documents.

7. INTERCODER RELIABILITY

Usually, intercoder reliability is assessed by applying a suitable quantitative measure taken from the literature. In his overview, von Eye [35] describes three different coefficients. The simplest one just expresses the raw agreement by the ratio of the agreements relative to the sample size. The other two coefficients are known as Cohen's kappa and the kappa-coefficient of Brennan and Prediger (for detailed description see [37]). Both coefficients calculate the ratio of agreement relative to the sample size and adjust the value by a ratio of agreement by chance. Another coefficient, called Krippendorff's alpha was proposed by Krippendorff [25]. It calculates the ratio of observed disagreement relative to the expected disagreement.

All these coefficients have some assumptions in common. First, the rated objects (usually documents) have to be independent. Second, the coders operate independently, and third, they assume the categories to be independent, mutually exclusive and exhaustive. The first two assumptions are fulfilled in our case, but our categories are neither independent, mutually exclusive nor exhaustive. As a consequence, all these coefficients are not applicable in our case. Therefore, in the absence of alternatives, we have to restrict our investigation of intercoder reliability to the pure percentage of equally coded text elements among all coded elements.

Principally, these text elements can be represented by documents or by codings. The former is simple to calculate and unambiguous, but makes no sense in our case because all our double-coded documents were assigned to all our categories. In consequence, we have to calculate agreement percentage among all coded text elements.

Fortunately, our software MaxQDA offers a feature to calculate intercoder agreement based on the character overlap of text segments. Additionally, the overlap percentage of the compared text segments can be adjusted. To investigate the variation of the agreement percentage with the overlap, we calculated three agreement values with overlap percentages of 10%, 50% and 90%. For instance the overlap 50% means that two coded text segments are regarded as the agreement, if the two segments have 50% of the characters of one of both segments in common. Unfortunately, the software does not contain any documentation about the base value for this percentage. Yet, the results are quite plausible. As expected, the agreement percentages decrease with an increase of the overlap.

Table 3. Intercoder agreements according to MaxQDA

Document	10% Overlap	50% Overlap	90% Overlap
FR	38%	30%	26%
IT	66%	50%	42%
KO	51%	44%	38%
NZ	39%	31%	24%

Obviously, our intercoder agreement was not really good. To explain this, we inspected the different codings in all double coded documents more closely, which revealed several reasons.

It became evident that the first, introductory pages of the case studies caused most of the differences. Sometimes, one coder coded these pages in detail with many different categories, while the second coder did not code anything there. Maybe this was because he or she supposed that every information contained in these pages would appear later in the paper anyway. As an example, Figure 6 shows the "document portraits" of the two coding sessions for the case study from Korea (KO), as generated by MaxQDA. The small squares represent coded segments in the text, where each color stands for a certain category.

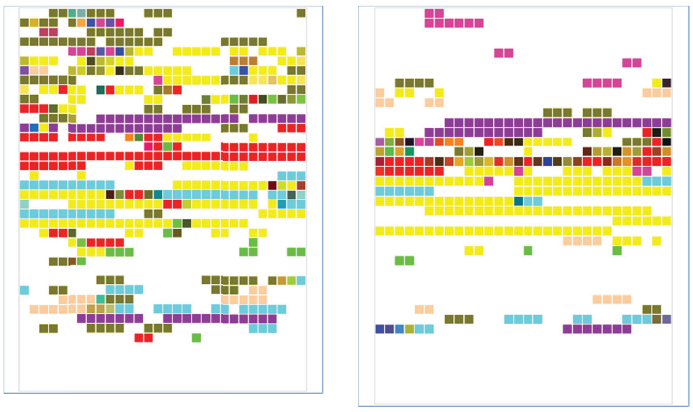


Figure 6: Document portraits of the case study of Korea, first and second coder

To detect variation among the categories, we calculated the agreements in the different categories at 50% overlap. It turned out that the categories *Competencies*, *Programming languages or -tools* and *Content/Knowledge* were comparatively easy to code, while *Goals, Organizational Aspects* and *Fields of Computing Education* were much more difficult. This might be caused by the following reasons.

First, we found a substantial variation in the length of the coded text segments in several categories (see Table 2), for example, in *Content* or *Programming languages or -tools*, where some codings comprised only one word like "Algorithms" or "CS", while others covered a whole sentence or even paragraph.

Second, there were certain "boring" effects among the coders, such as in the category *Fields of Computing Education* (pink squares in Figure 6). Although we had taken this into account already (by restricting the coding of sentences containing "ICT" or similar, to the first few or the major occurrences in a certain document), this effect contributed to the bad agreement results. The same issue occurred for *Programming languages/-tools* (light green in Figure 6), where certain denominators were coded only at their first or second occurrence. Nevertheless, this category was consistently coded in comparison to the others, presumably because it didn't offer much room for interpretation.

Third, despite our detailed coding rules, some categories still left room for interpretation. For example regarding *Organizational aspects* (olive squares in Figure 6), we agreed to code only statements that point to the present situation in that country or state. Yet this was sometimes really hard to tell. Regarding the category *Goals* (salmon-colored in Figure 6), the most relevant information was contained in complex sentences instead of being addressed explicitly, which might explain the poor agreement percentage.

Despite these differences between the first and second coding run, we are convinced that we found and coded most of the relevant information from all these case studies. After all, we intended to get as much valid information as possible out of the texts, but not necessarily in all cases together with its precise position.

Nevertheless, to improve the validity of our results, we performed an additional feedback loop, asking the leading authors of all case studies to reconfirm our coding results for their country or state where this seemed feasible during the time we had. At the end we received feedback on our codings of *Content/Knowledge*, *Teacher education* and *Assessment/Examination* from 9 of 12 countries or states: BY, FR, IS, RUS, KO, NZ, SW, UK, and GEO.

8. EVALUATION OF CODING RESULTS

As already explained in the preceding section, the following results were derived by analyzing the 2284 codings of the first coding run (See Table 2). The analysis was executed by several teams of group members in parallel to the second coding run.

8.1 Fields of Computing Education at School

8.1.1 Analysis Procedure

In a first step, the codings of the category *Fields of Computing Education* were paraphrased according to the suggestions of Mayring [29]. Depending on the coded segment, this paraphrasing included shortening and rephrasing the respective text section. If codings were too short to make sense on their own (e.g. one-word codings like ICT), they were recoded by including their context in the original document. Afterwards, the extended codings were paraphrased again if necessary. Table 4 shows some examples.

Codings were ignored for the further analysis, if they:

- apparently had no relation to the research questions,
- addressed only details on a lower level, but not the fields of computing education (e.g. programming knowledge, that appears almost everywhere in very different forms),
- referenced only general definitions or usage of the terms in external literature without relating to the addressed country or state,
- were presented in national language without English translation,
- did not refer to the regarded country or state.

Table 4. Performed actions during paraphrasing

Coding	Paraphrase	Performed Actions
we use the term CS to denote the German term "Informatik" which refers to the academic discipline as well as to the subject in schools.	The term CS is used to denote the German term "Informatik" which refers to the academic discipline as well as to the subject in schools.	Rephrased
In 1995 a new school curriculum was introduced that included "technology" as a learning area, and ICT was defined within that area with a broad and literal view of it being about "information" and "communication," but there were concerns that schools were not engaging with the topic	ICT is defined with a broad and literal view about information and communication	Shortened and rephrased
the need to go beyond the dichotomy between ICT and CS in the perspective of an informatics education for all.	CSE should integrate ICT and CS	Clarified the meaning and rephrased
Information and Communication Technology (ICT)	ICT should be integrated in all existing subjects but not as an independent subject	Recoded by including original context and rephrased

After the rephrasing of the codings we categorized the resulting text snippets according to their type. This resulted in the following nine categories: *usage of terms, focus, synonym, definition, organization, concerns, goal, educational goal,* and *misunderstanding*. The codings of the categories *goal* and *educational goal* were analyzed separately (see Section 8.2.4) and thus skipped in this Section. Table 5 presents the remaining categories, an explanation and an exemplary coding. Based on these categories, the rephrased text passages were rewritten to represent the original content in an abstract way. On the basis of these abstractions, the following results were derived. In summary, we found 40 different terms for fields of computing education:

Terms referring to the academic discipline: computer science (CS), informatics.

Terms referring to processes within the discipline: automatic data processing, computer programming, computing, computing and natural science, programming.

Terms referring to the technology aspect: computer and technology, computer technology, digital technology, information and communication technology (ICT), information technology (IT), media technology, multimedia, multimedia (and web) technology, technology, digital technologies.

Table 5. Categories for the fields

Category	Explanation	Exemplary Coding
Usage of terms	The terms are described in detail. In particular, there is an explanation for how a specific term has to be understood.	ICT can be seen as a discipline as well as a tool
Focus	The focus of a specific field is explained.	CSE is mainly concerned with algorithmic problem-solving
Synonym	Terms that are used synonymously in this country/state	CS is used as a synonym for Informatics.
Definition	The document provides an explicit definition of the term	CS is the academic discipline that encompasses the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society.
Organi-zation	The terms for the fields are used for organizational purposes. For example, school subjects are named with a field term or the fields are used in curricula.	CS and Programming were replaced by the subject ICT.
Concerns	In some documents concerns about the usage of the terms are mentioned	The interchangeable use of the terms CS, Informatics, ICT, and digital literacy creates the illusion that CS is already being taught and integrated at the school level and as a result, efforts to improve the situation for CS education at school often ends with giving more importance to digital literacy or ICT
Misunder-standing	Misunderstandings of terms are addressed.	It is important to differentiate between ICT and CS

Terms explicitly referring to education: computer application in education, computer(-related) education, computer studies, computing education, computer science education (CSE), information (and communication) technology education, informatics education, informatics practices, programming education, technology education.

Terms referring to educational goals: computational thinking, computer expertise, computer knowledge, computer literacy, computing awareness, computing fluency, computing literacy, digital knowledge, digital literacy, information and communication technology literacy, information knowledge, information literacy.

8.1.2 Local Application and Understanding of Terms

The following Table 6 presents all the terms we found in the codings of the category *Fields of Computing Education* for each regarded country or state. Apparently, in some countries or states many different terms are used, in others only a few.

Table 6. Terms for the Fields by Country/State

Country/ State	**Terms found in the codings of the category "Fields"**
Bavaria	CS, Informatics
Finland	CS, ICT, Programming education
France	CS, CSE, ICT
India	CS, CSE, Digital literacy, ICT, Informatics, Informatics practices, IT, Multimedia and web technology, Multimedia technology
Israel	Computing, CS, CSE, Informatics
Italy	ICT
Korea	Computer education, Computer literacy, Computer-related education, Computing education, CS, CSE, ICT, ICT education, ICT literacy, Informatics education, IT education
NRW	CS, CSE, Digital literacy, ICT, IT
NZ	CS, Digital technologies, ICT
Russia	Computer literacy, CS, Informatics
Sweden	Automatic data processing, Computer and technology, Computer application in education, Computer expertise, Computer knowledge, Computer programming, Computer technology, Computing, Computing and natural science, Computing awareness, Computing fluency, Computing literacy, CS, CSE, Digital knowledge, Digital literacy, Digital technology, ICT, Informatics, Informatics education, Informatics literacy, Information knowledge, Information literacy, IT, Media technology, Multimedia, Technology, Technology education
UK	Computational thinking, Computer studies, Computing, CS, Digital literacy, ICT, ICT education, IT, IT education, Programming, Technology
GEO	Computing education, Computer literacy

Additionally to the summarizing results presented above, we found many very different applications and interpretations of the variety of terms that describe the fields of computing at school. In the following, we present a choice of these finding in the context of the respective country or state.

In the paper about *North Rhine Westphalia (State of Germany),* the term CSE is either used to refer to the training of ICT skills as well as to the application of methods and principles of the academic discipline CS. The term CS is used to refer to the academic discipline as well as to the subject in schools. According to the paper, digital literacy is taught through other subjects with a focus on software applications, media or even basic concepts of Computer Science. CSE changed its focus a number of times throughout its history. While early approaches centered around algorithmic problem-solving or the usage of standard applications, current concepts focus on aspects that are believed to contribute to general education, such as the relation between real-world phenomena to CS concepts, sociotechnical information systems, and the comprehension and modelling of IT systems. Some actors in the field use the term CS as a synonym for digital literacy. This results in the concern that CSE, which only focuses on ICT skills, leads to an inaccurate or at least incomplete understanding of CS.

The paper about the situation in *France* states that CSE should integrate aspects of ICT and CS. In addition, the authors describe the need for ICT and CS to agree upon a common set of progressive goals and activities.

In the *Finnish* paper the term CS refers to the scientific aspects of the discipline. The term ICT focuses on the use of computers and computer applications, the term programming education on technology aspects. Moreover, the paper refers to the opinion that ICT should be integrated in all existing subjects and should not be taught as an independent subject.

In the paper about the situation in *Georgia* (USA), Computing education is described to focus on courses in computer literacy and the use of computers.

According to the *Italian* paper, vocational education focuses on the potential of ICT as a means to reorganize companies.

In the paper about the situation in *New Zealand*, the authors mention that the term programming was replaced by the term CS to broaden the view on the discipline. ICT is defined with a broad and literal view about information and communication. Moreover, it is stated that ICT can be seen as a discipline as well as a tool. The term digital technologies refers to standards that allow schools to address programming and CS aspects.

In the paper about *Bavaria (State of Germany),* CS is described to focus on various aspects of automated information processing and to encompass the study of computers and algorithmic processes, including their principles, their hardware and software designs, their applications, and their impact on society. The term Informatics is used as a synonym for CS.

The description of the situation in *Israel* informs the reader that the terms CS, Computing, and Informatics are used synonymously by some in the context of CSE.

In the *Korean* paper, the term CSE includes computer literacy, ICT literacy, and CS. Computer Science was not related to knowledge about computers and ICT. Some use the term computer education as a synonym for computer-related education, others use IT education, ICT education, CSE, and Informatics education as a synonym for Computer education. There is a subject named "Computer" and another one named "Information society and computer", which focus on ICT literacy, therefore the term "Computer" is used for computer-related education. The term computing education was related to ICT literacy, but it is also used to express a focus on CS principles and concepts.

In the paper about the situation in *Russia*, the term CS (or synonymously Informatics) encompasses fundamental and technological aspects. The term computer literacy has changed its meaning over time from programming to operating a computer as a user due to the spread of personal computers and the development of application software. In elementary schools, elements of Informatics are taught within the subjects "Mathematics" and "Technology".

In the paper about the situation in the *UK,* the term Computer Studies was used for a school subject which covered CS aspects including hardware, logic, binary, programming, and various other aspects of computers. CS required a thorough grounding in logic and set theory. This subject was replaced by the subject ICT. There was an ICT syllabus, which predominantly focused on using computers and technology (without giving insight into how it works) to foster IT skills and digital literacy, although there was still a mention of programming. The authors report the misunderstanding that ICT and CS would label the same subject. The term digital literacy is also used to encompass computational thinking, which includes logical reasoning, problem-solving, debugging strategies, algorithmic thinking, and programming. Finally, the term computing is used to cover digital literacy, information technology, and CS. The subject ICT has been renamed Computing. There is also the position that digital literacy is to be embedded in every subject.

In the paper describing the situation in *Sweden,* the concept of "Computing" is used for the current school setting, replacing Informatics and CS. The term "Information technology" (former known as "Information literacy" and "Informatics") is associated with technology education. ICT includes the usage of multimedia and information technology in creative processes and applications, the "flow of information", how to use digital technology, and how to develop critical thinking about the information available on the internet. IT-related courses in the field of Technology are oriented towards information and media technology and include computer communication, programming, digital media, web development, and computers and ICT. According to the paper, some use Informatics as a synonym for computing, information technology, information knowledge, computer knowledge and automatic data processing as well as for CS. Additionally, some use the terms computing awareness, computing literacy, information literacy, and computing fluency and computer expertise as a synonym for Informatics literacy. In former times, Informatics education primarily focused on programming knowledge related to automatic data processing while CSE used to focus on hands-on training. Additionally, Informatics focused on societal factors and less on technological aspects of programming. Later, Informatics education shifted its focus to computer application in education. This was caused by concerns that Informatics should not be treated as a cognitive framework in itself. Furthermore Informatics should cover "Programming" which is not included by standard.

The situation in *India* as described in the analyzed paper refers to ICT as six themes: connecting with the world, connecting with each other, interacting with technology, creating with ICT, possibilities in education, and reaching out and bridging the divide. Amongst others, there is a course on foundations of information technology, which includes knowing the hardware components of a computer and the different types of software as well as operating systems. The subject "Informatics Practices" is oriented toward getting familiarized with computer systems and developing web-based applications using HTML and Java. According to the paper, some use CS, Informatics, ICT, and digital literacy as synonyms for each other. The authors of the paper mention the concern that the interchangeable use of the terms CS, Informatics, ICT, and digital literacy creates the illusion that CS is already being taught and integrated at the school level.

8.1.3 General Results

Apparently, the terms that describe the fields of computing at schools are interpreted very differently. For example, in some countries CSE covers ICT and CS (NRW, France, and Korea) or CS is an academic discipline (NRW, Finland, Bavaria, Russia, UK). For Computing education there are also varying definitions. In the US and Korea Computing education includes courses in computer literacy and the use of computers. Furthermore, in Korea, there is a focus on CS principles and concepts, while in the UK digital literacy, information technology, and CS is included. ICT is understood as the usage of computers and software (Finland, UK) or as a discipline and tool (NZ). In Sweden ICT includes the usage of multimedia and information technology in creative processes and application (Sweden). Digital technologies, as used in NZ, includes programming and CS aspects, while digital literacy used to encompass computational thinking, which includes logical reasoning, problem-solving, debugging strategies, algorithmic thinking, and programming. Finally, computer literacy sometimes includes programming and operating a PC as a user as in Russia.

The difficulty to find a common understanding for all these terms is illustrated by their varying synonymic application. Figure 6 displays some of the terms as nodes and their synonymic usage in some countries/states as edges. The edges are labelled with the country/state where these terms are used synonymously.

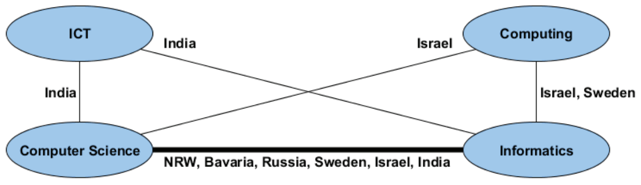


Figure 6. Synonymously used terms in some countries.

Furtherly, the names that are used to denominate school subjects are very different: Computer Studies (UK), Computing (UK), ICT (UK), Informatics (NRW, Bavaria), Computer (Korea), Information society and computer (Korea), Computing and natural science (Sweden), Computer Practices (India), Multimedia Technology (India). Digital literacy (NRW, UK)/ ICT (Finland) with a focus on using software and technology is taught through other subjects. In Russia, even elements of Informatics are taught through other subjects (in elementary school). The focus of education shifted in a number of countries in history (e.g. NRW, UK, NZ, and Sweden).

Additionally, we found a set of normative postulates: ICT should not be taught in a separate subject (Finland); Computing should be taught integrated in the different subjects (Sweden). Finally, ICT- and CSE have to agree upon a common set of progressive goals and activities (France), Informatics should not be considered isolated from other fields and should cover "Programming" (Sweden), and the interchangeable use of the terms CS, Informatics, ICT, and digital literacy creates the illusion that CS is already being taught and integrated at the school level (India).

8.1.4 Discussion

The most interesting result was that there are so many different terms that all claim to denote Computing (including Computer Science and ICT) at schools in any way. Apparently, these terms are applied in a very different way. In addition, sometimes even within a certain countries or state, two or more different terms are used to refer to the same field. Conversely, the same term may denote different fields in different countries or states.

Apparently there is an urgent need for a standardization of all these terms, otherwise serious misunderstandings among stakeholders of different countries seem inevitable.

8.2 Goals and Competencies

8.2.1 Disambiguation

Originally, we had chosen this research question (see Section 4) with the aim to collect and summarize information about the different intentions of CSE in schools. The category *Goal* should label general, long-term *global objectives* in the sense of Anderson & Krathwohl [1]: "Complex, multifaced learning outcomes that require substantial time and instruction to accomplish", for example "creative use of IT" or "problem solving". On the other hand, we understood *Competencies* in the sense of Weinert [46] as "the cognitive abilities and skills possessed by or able to be learned by individuals that enable them to solve particular problems, as well as the motivational, volitional and social readiness and capacity to use the solutions successfully and responsibly in variable situations." Anyway, a competency description should make clear precisely what the students should be able to do (e.g. "to write a Java program that implements a certain algorithm").

8.2.2 Coding Results

As described above, the first coding run had produced 127 codings in the category *Goals* and 308 in *Competencies*. Yet, it turned out during the coding process that the authors of the case studies often had very different understanding of these two terms. For example "understand how CS shapes our world" could have been a goal for one author, but a competency for another (or for our coders). Our coders frequently met the dilemma to code a text snippet describing an intention according to our understanding or as labeled by the authors. As a consequence, we had a number of inconsistent codings, mostly goals in our sense that were labeled as competencies. Thus, during the analysis process, we have shifted these codings to the category *Goals* as described in the following sections.

8.2.3 Analysis of Competencies

First, we exported the 308 codings of the Category *Competencies* from MaxQDA into a spreadsheet in order to paraphrase and normalize them. Next, a pair of researchers normalized the verbs; put them at the beginning of the statements and positioned objects and adverbs within the statements correctly in order to classify them as competencies. Afterwards we split those statements, which contained 'and' respective 'or' operators into two separate ones.

In a subsequent step, we identified very general statements, which might be rather goals or general objectives of CSE than competencies, e.g. "conduct computational thinking". By this way, 77 goal descriptions (originating from 62 codings) were moved to the category *Goals* and analyzed later (see next Section).

Further, we identified statements, which obviously are neither competencies nor goals (for example "ability to use ICT is not sufficient to solve given problems") and marked them as 'questionable'

After conducting this process, we received altogether 249 competence statements. As this list is too long to be displayed in this paper, we have extracted 15 randomly chosen competency statements to illustrate the results.

Table 7. Randomly chosen competency descriptions (15 of 249)

Nr.	**Competency Definition:** the students are able to ..
9	understand how CS shapes our world
23	be aware of the persistence of digital information on interconnected spaces
53	program sequential search, binary search
93	evaluate an interface in the light of usability heuristics such as those given by Nielsen (useit.com)
114	evaluate alternative models in order to choose one of them
125	write an algorithmic solution for the problem
132	share information
145	understand and express algorithms
146	design and write algorithms
147	understand of alignment and sort methods
175	analyze the model and the simulated object (process)
184	interpret results obtained during simulation of real processes
199	use logical reasoning to explain how some simple algorithms work
228	code a solution method in a language
231	connect to the Internet

Up to now, due to limited personal resources, we have recoded some of these competency definitions automatically using MaxQDA by searching the knowledge element “algorithm(s)”. Figure 7 shows the competencies that refer verbatim to this term.

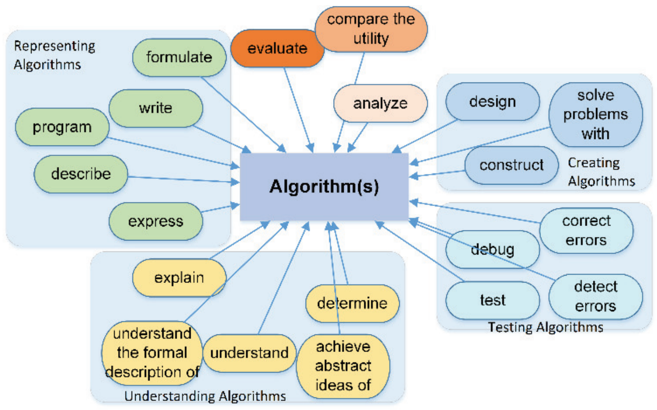


Figure 7. Cognitive processes referring to algorithms

As displayed in Figure 7, the coded competency definitions describe many different cognitive operations on Algorithms that partly can be grouped into the four super-categories *Representing*, *Understanding*, *Creating* and *Testing* Algorithms. Additionally *Evaluating*, *Comparing* and *Analyzing* seem quite close to each other, forming a fifth category (to be named). These results suggest that these five super-categories could represent a competency each in the sense of Weinert [46]. Furthermore, the more narrow skills contained in each of these super-categories (e.g. *Formulate*, *Write*, *Program*, *Describe*, *Express* in *Representing*) could operationalize these competencies by suggesting respective test items, e.g. about formulating an Algorithm. In future work to evaluate the list of 249 competencies regarding other knowledge elements by qualitative data analysis.

8.2.4 Analysis of Goals

Similar to the competencies, the evaluation of the goals was started by processing the respective codings according to Mayring [29], which included the splitting of several codings, if they had addressed more than one goal. This work was done by two members of the group. Table 8 shows some examples for this processing step.

Table 8. Examples for the Paraphrasing of goals

Coding	Paraphrase
bridging the digital divide across socioeconomic and geographic segments	bridging digital divide
mediate and facilitate the appropriation of capabilities and habits of importance for problem solving and communication	problem solving, communication

Table 9: Categories of addressed goals

Goal	Addressed by
digital literacy (including use and handling of tools)	FI, USA, BY, KO, RUS, UK, SW, IN, IT, NRW, NZ
computational thinking (including algorithmic and logical thinking)	FR, FI, USA, IS, RUS, UK, KO, SW, IN
problem solving	NRW, USA, IS, KO, RUS, UK, SW, IN
understand basic concepts of CS and IT	NZ, BY, IS, KO, SW, IN, FR, IT
career preparation and choice	NRW, SW, BY, IN, FR, IT, KO
support awareness of social, ethical, legal and privacy issues and impact of CS	NRW, KO, FR, RUS, UK, SW, NZ
general education to participate in society responsibly	NRW, BY, KO, SW, IN, RUS,
prepare for university	NRW, KO, SW, IN
student development	FR, IT, RUS, NRW
attract and motivate female and male students	SW, IS, KO
create IT	USA, IS, NZ
holistic view	RUS, SW, IN
connecting to real world contexts	NRW, KO
creative use of IT	UK, KO
limits and risks of CS	SW, BY
support communication about IT	BY, SW
support maths and science	IS, SW
apply IT in other subjects	BY
deeper knowledge of CS	SW
growth of knowledge society	IN
modern and relevant curriculum	IS
picture of CS and programming in society	SW
representing thinking processes	KO
rise and discover talent and attitude towards CS	KO

In a second paraphrasing step, the codings were reduced once more. As an example, all statements referring to any kind of computer literacy, ICT literacy or similar were paraphrased as “digital literacy". Every item referring to the idea that every student should learn CS like "IT for all" was paraphrased as "general education" and everything like "bridging the gap" or "digital divide" to "participation in society". Sometimes one text part that was coded for goals resulted in several single statements for goals, because it covered different aspects, e.g. "computational thinking" and "preparing for university".

This step resulted in 174 goals that stemmed from the original 127 codings of this category. As described above, 77 additional codings, originally coded as competencies, were moved to this category, resulting in 203 statements about goals. Finally, the statements about goals were categorized and unified, as far as this was possible without loss of information in consideration of their mapping to the case studies. Afterwards, some categories were deleted after going back to the original text because the coding resulted in a re-categorization as a competence or content (e.g. "programming" in the meaning of knowing or applying programming abilities or concepts to produce software). This resulted in 24 categories of goals. Table 9 shows these categories and relates them to the case studies.

In our eyes, these findings provide detailed information on how Computing at schools is justified in the regarded countries or states. Additionally, the information expressed by Table 9 could be used by stakeholders that look for arguments in favor of CSE in schools. Finally, the number of case studies that address a certain goal can be taken as a measure for the consensus among countries or states regarding this goal.

8.3 Content/Knowledge

Originally 1053 text snippets were coded in the category *Content* (see Table 2). After some syntactical normalization and removal of doublets, 816 codings remained. These were recoded inductively, which produced 19 content categories (see Table 10).

These categories were constructed according to the following rules:

1) If a coding didn't fit to any category then

- a new content category was created or
- one of the categories was extended (to comprise more codings) and accordingly renamed.

2) If a coding could be assigned to more than one category, these categories were joined into one category.

3) If less than 3 coded elements remained in a category, this category was deleted and their elements were moved to another related category that was (if needed) suitably renamed.

At the end each category had at least three elements and each element of coding was assigned to one of these 19 categories.

During this process, we detected and deleted 54 codings that didn't describe any learning content or knowledge. All the remaining 762 codings could be assigned to one of the newly defined categories that are displayed in the first column of Table 10.

Finally, the content categories were mapped back to the documents, where the original content codings had been found. Yet, due to the complicated procedure that was applied to derive these categories, we didn't consider that the results were sufficiently valid. To ensure this validity, we sent the resulting table to the original authors of the coded case studies and asked them to cross-check the corresponding entries in the respecting column. Unfortunately, only 8 authors of relevant papers responded. Table 10 shows the results for the corresponding 8 countries or states. Yet, despite this validation process, the reader should be aware that this Table represents only a snapshot in time. Some of the respective curricula might have changed already at the time when this paper appears.

In our opinion, the content landscape displayed in Table 10 represents the core information of our "global snaphot". Despite its limitations, it shows which learning content is addressed by the 8 countries or states that had responded to our confirmation request.

Table 10. Content/Knowledge categories covered by the curricula of some countries/states

Content Category	BY	FR	IS	KO	NZ	RUS	SW	UK
Algorithmic concepts	X	X	X	X	X	X	X	X
Application systems	X		X	X	X	X	X	X
Artificial intelligence				X	X			
Computer and Communitation device	X	X	X	X	X	X	X	X
Computer human interaction				X	X			
Computer Networks	X	X		X	X	X	X	X
Data protection	X				X	X		
Data security	X			X	X	X		
Data structures	X	X	X	X	X	X		X
Database systems	X		X	X	X	X	X	X
Digital media	X			X	X		X	
Ethical issues	X			X	X			
Information and digitalization	X	X	X	X	X	X		X
Mathematical aspects of CS	X	X	X	X	X	X		X
Modeling	X	X	X	X	X	X		
Object-oriented concepts	X		X	X	X	X	X	
Operating systems	X	X	X	X	X	X	X	X
Problem solving	X	X	X	X	X	X	X	X
Programming Issues	X	X	X	X	X	X	X	X

Additionally, despite the limitations of this result, the 19 content categories might eventually represent a complete overlap of what CS content is taught in schools currently.

8.4 Programming languages and tools

As described in Section 6, we have found 107 text snippets that provided information about the applied programming languages or tools (see Table 2). In a first step, we listed and separated these findings in four sub-categories:

- A) Hardware based systems, e.g. LEGO Mindstorms or Raspberry Pi,
- B1) Educational programming environments with own language, e.g. Alice or Scratch

– B2) Educational programming environments based on other languages, e.g. BlueJ or Java's Cool.

– C) Professionally used programming languages, e.g. C# or Java.

Table 11. Educational programming environments with own language (B1)

Environment	Used in K12-Schools of
Scratch	USA, NZ, Korea, UK, NRW
Kodu	USA, Korea
LOGO	Korea, UK
AgentCube	USA
AgentSheet	USA
Alice	USA
Blockly	USA
Game Maker	USA
Micro Worlds	NRW
Robot Karol	Bavaria
Squeak Etoys	Korea

Table 12. Educational programming environments based on other languages (B2)

Environment	Used in K12-Schools of
BlueJ	Bavaria
Greenfoot	USA
Java's Cool	France
Jeroo	USA

Table 13. Professionally used programming languages (C)

Language	Used in K12-Schools of
Java	France, Finland, USA, NZ, Israel, Sweden, India, Bavaria, Korea
C++	USA, Israel, Sweden, India, Korea
Python	France, NZ, Korea, India
AppInventor	USA, NZ
BASIC	Finland, Sweden
C	Israel, Korea
C#	USA, NZ
HTML	France, India
JavaScript	USA, NZ
Pascal	Finland, Israel
VisualBasic	USA, NZ
ALGOL	Sweden*
COBOL	Sweden*
FORTRAN	Sweden*
Logic Programming	Israel
PHP	NZ
VB script	India
XML	India

*Due to the historical parts of the case study about Sweden

Regarding hardware-based systems of category A our case studies told us that educational robots are used in Korea and Raspberry-Pi in UK classrooms.

The Tables 11-13 display the application of languages or tools in the different countries according to the information in our text corpus.

Besides the systems listed above, some environments were created locally that provided self-paced guided learning to students for various programming languages. These learning environments allow students to work on a set of small problems using instructions at each step, run the source code and answer question to present next set of instructions. Some schools in Korea are using code.org (www.code.org) and the "cyber home learning system" (www.virtualschoolsandcolleges.eu) in their K-12 computer science curriculum, while some schools in NZ are using "code avengers"or "interactive python" (codeavengers.com, interactivepython.org)

8.5 Assessment/Examination of Students

The 98 codings of the category *Assessment/Examination* were analyzed by one member of the team (See Table 2). Afterwards the results were reviewed and discussed by the whole working group. The analyzer read all the coded text snippets and sub-categorized all relevant information. By the end of this process the following five sub-categories emerged: *Level* (of education where the assessment takes place), *Topics*, *Certification*, *Count for admission*, *Application* (universal or not). In addition, we introduced one more category *Additional information*.

Afterwards a second round of analyzing was conducted, applying these sub-categories to describe the assessment used in each regarded country or state. To validate our results, we have asked the authors of the case studies for feedback and accordingly double checked most of the case studies. Unfortunately, the Indian case remained unchecked. To compensate this, it was controlled by the Indian member of our working group. Table A1 in the appendix displays the results of this categorization over all countries or states. In addition, the following text explains the local assessment or examination practices in more detail.

In the final graduation from *German* grammar schools (called Abitur), there are formally decided standards for all subjects, called *Einheitliche Prüfungsanforderungen in der Abiturprüfung* (EPA). In *Bavaria,* CS can be chosen for written as well as for oral examination. But only a small number of students enroll in CS courses, and even fewer students take one of their final examinations in CS. Topics to be covered by higher secondary education final CS exams are object-oriented modeling (including programming), entity relationship modeling, automata, algorithmic modeling, functional modeling (optional), rule-based modeling (optional), formal languages, computer-human interaction, privacy, security, computer architecture, computability, (practical) efficiency, and societal issues.

In *France*, Informatics and Internet Certificate (Brevet Informatique et Internet - B2i) is a novel object in the educational system. It doesn't represent an official examination, but a certification of competences. Another certificate of competences named C2i, assesses students' competences in ICT at the bachelor's degree level. It certifies a basic set of rather technical competences organized in five domains. Some of these skills are specific for organizing student assessment to IT project development: "Describe and explain a situation, system or program"; "Design and implement an IT solution in response to a problem". Others are more general: "Collaborate effectively within a team in a project" and "Communicate in writing and orally". One of the main big challenges in CSE is assessing students' learning outcomes, French ISN syllabus includes information about skills, which is quite informative about what is expected from students.

In *New Zealand* the National Certificate in Educational Achievement (NCEA) is the main school-leaving assessment; NCEA is of particular importance since it provides assessment that could count towards academically oriented qualifications. Assessment standards include a set of generic technology standards, but has specific standards in four areas of technology: Construction and Mechanical Technologies (focuses on making and knowing how to make products and devices), Design and Visual Communication (focuses on where visual literacy and creative thinking is developed, using visual communication techniques), Digital Technologies (focuses on applying and knowing about computer science, electronic and digital applications), and Processing Technologies (focuses on formulating and knowing how to formulate processed products) Student work is assessed by a government agency (NZQA), which tracks student performance and makes general statistics publicly. The programming standards are all "internal" which means that they are assessed by the teacher, with external auditing (moderation) to ensure consistency. The internal assessment of programming means that the teacher can observe the process that a student uses to program, including how independently a student is working once they have learned to program. The computer science standards are all "external" which means that they are assessed by an anonymous appointed marker (or group of markers), organized by NZQA. Due to various resource constraints and timetabling issues, assessment for the external computer science standards could not be done by standardized tests or exams, and so they are assessed by submitting written reports or projects.

In *Israel* the three years of high school culminate in a set of matriculation exams. The matriculation exams are crucial for admission to most Israeli universities. Alternatively, unique and innovative students' assessment techniques are applied.

In *Korea*, the evaluation for each subject (including CS) consists of two parts: a paper-based assessment and a performance evaluation. The written test examines students' knowledge and understanding of the subject and the performance evaluation assesses students' ability to perform specific tasks. In the CS subject, the paper-based assessment mainly includes understanding of CS knowledge and the performance evaluation assesses students' problem-solving ability, coding skills, and other abilities. Students get their grade each semester, which combines the results of the two evaluation methods. Each teacher adjusts the ratio of the two evaluation methods, actual performance is generally regarded as more important in the informatics subject. In addition to the school evaluation, there is another important test for further study, the College Scholastic Ability Test (CSAT). In CSAT only the career exploration course has computer-related contents and it is just for vocational high school students. Subjects within the curriculum or obtained a certificate related to the computer, which was reflected on the school record, could use it for a university entrance examination in addition an information literacy license was issued to students, and the fact was recorded in their record of school life. The certification is issued to:

- those who completed a course of an informatics-related subject
- those who completed informatics-related education using extracurricular time
- those who passed the information literacy test
- those who got computer-related certificates.

Many universities utilized this license as a means for admission. Another evaluation system is the so called Test of Practical Competency on IT (TOPCIT). It aims to assess the competency of IT human resources. This system can provide detailed achievement goals to the students, and also can be a measure of students' competency.

In *Georgia (USA)*, CS is available as an Advanced Placement (AP) course.

In *Russia*, the Unified National Examination (UNE) in Informatics is not compulsory for all school graduates. It is necessary, for admission to Informatics-related study programs. The exam tasks are grouped into 3 blocks: "Mathematical foundations of Informatics", "Algorithmization and programming", and "Information and computer technology." The examination covers the key topics from the Informatics school syllabus. The tasks include analysis of algorithms, developing computer programs, etc. The evaluation is conducted by the experts of regional examination boards based on standard assessment criteria.

In the *UK (England)*, alongside the developments within the National Curriculum, and encouraged by the Computing at Schools (CAS), the awarding bodies (who establish school qualifications across the country), have developed General Certificate of Secondary Education (GCSE) qualifications (taken at age 16) in CS (also called Computing), to augment the existing suite of qualifications which include GCSEs in ICT and more vocational coursework-based qualifications in ICT. To date, there are four GCSE programs in Computing/Computer Science available to schools.

In *India*, questions in the exam test the depth of understanding of the constructs of a programming language. In particular, students are expected to write one full program in Python, 5 SQL queries, present previously done project work, and document all the exercises done so far. As per the blueprints of the various CS-related subjects, 20% of marks are allotted to test the "Knowledge" instructional objective, 30% is allotted for testing the "Understanding" objective, and 50% is allotted to testing the "Application" objective. Hence, the exams allot 70% marks to theory and 30% to the practical.

8.6 Teacher education

The K-12 Educational Task Force of the Association for Computing Machinery (ACM) advocates for CS teacher training programs that prepare the CS teachers with the necessary pedagogical skills to convey the information to the students at the appropriate level [38]. Along the same lines, the US Computer Science Teachers Association (CSTA) report on CS teacher certification, advocate for the establishment of a Computer Science Praxis exam that will assess teachers' knowledge of CS concepts and the teachers' knowledge of pedagogy [42]. Common concern in many countries is the difficulty of ensuring that teachers have the technical, content, and pedagogical knowledge needed to teach CS; and this knowledge is continually refreshed and upgraded [15].

Regarding teacher education, we coded 170 text snippets in our text corpus (see Table 2). One researcher inspected all these and sub-coded relevant information. By the end of this procedure, the following sub-categories had emerged: *Required degrees*, *Training, Examination/Certification and Informal teacher education activities*. Afterwards a second round of reading was conducted, applying these categories to summarize the teacher education information of each case study.

Similarly to the categories *Content* (see Section 8.3) and *Assessment/Examination* (of students, see Section 8.5), we were aware that the quite complicated evaluation procedure of the case studies might produce invalid results. Therefore, we have asked the authors of the case studies for feedback again and adopted our results accordingly (except Finland, India, Italy). At least the results for India were checked by the Indian member of our working group.

Table A2 in the appendix displays the results of the evaluation of teacher education issues over all countries or states. In the following text, additional information about details of the local situations is provided.

In *NRW* (Germany), the first phase of teacher education is followed by 18-month of in-service training. During this phase, prospective teachers observe interns in a school and attend seminars in a teacher training college. The in-service training concludes with an examination 1) on the practical and theoretical knowledge of the prospective teachers by carrying out a lesson in each of the two subjects and 2) on the basic knowledge of public schooling, as well as basic subject-specific didactic elements by an oral exam. There is also a tradition of statewide in-service training events for CS teachers. These events are offered both by in-service teachers and academics, and cover topics ranging from pedagogical content knowledge and regulation issues for centralized examinations to best-practice reports.

In *France* the duration of the (CS) teachers' training (on campus, distance, or mix) ranges from 0 to 120 hours per year. Some districts offer a 2-year training course, which brings the total duration to 240 hours, personal additional work not included. CS graduates or those with teaching experience in high-level CSE may be allowed to teach without this additional training. The main certification is based on a set of rather technical competences organized in five domains. Level 2 certification (C2i) aiming at certifying professional competences for a series of professions: law, health professions, engineers, etc. and also teachers. The Ministry of Education announced that this certificate would be compulsory for teachers to receive tenure.

Teachers in *Finland* are required to have a master's degree (5 years, a total of 300 ECTS). Primary education teachers (grades 1–6) have a master's degree in general classroom teaching, whereas teachers for grades 7–9 and grades 10–12 have a master's degree with a major in the subject. In addition to the major, teachers are required to complete a minor both in school pedagogy (60 ECTS including training in an actual school) and in a second school subject to be taught (60 ECTS). Since there is no national CS curriculum, teacher training for CS is not well defined.

The key requirement for teaching in *Italian* secondary schools is a master's degree. Technology subjects can be taught by instructors with a master's degree in Architecture, Urban Planning, Industrial Chemistry, Engineering and other technology oriented degrees (but not Informatics!). CS subjects can be taught by instructors with a master's degree in Informatics, Physics, Mathematics and some Engineering degrees, like information, industrial, telecommunication, electronics and aerospace. It is often required CS teachers to receive their primary teacher certification in some discipline other than CS and then meet additional requirements to receive a supplemental CS endorsement – although there are no consolidated practices for Informatics education.

Teachers in *NZ* can teach CS subjects by holding a post-graduate education course that is available for teachers to learn by distance, and obtain a formal qualification in teaching computer science. There is an informal training and advising network; this network is consisted of university contacts, one academic at each of the country's eight universities was nominated as a local contact who could provide advice to local teachers, including finding senior students who could speak to classes.

In general, teacher students in *Bavaria* are educated and examined for one specific school type only (e.g., Gymnasium). For all types of schools, the teacher students have to pass two centralized examinations. The first one qualifies for the subjects to be taught, the second one on pedagogy and psychology. There are certain restrictions for the number, extent, and combination of the subjects that depend from the school type. Concerning Gymnasium, the students have to choose two subjects that are implemented on an equal level in their course of study. CS can be combined with mathematics, physics, English, or Economics and Law.

In *Israel* an undergraduate degree in CS is a mandatory requirement, as is formal teacher training and a formal requirement for a mandatory CS teacher certification. In addition there are mandatory in-service courses for teachers. CS teachers' preparation study program includes workshops, and new guidelines all intended to assist teachers in addressing the content and pedagogical challenges they face. There is a dedicated center which has the responsibility and funding to provide in-service workshops to assist the teachers to cope with these curriculum changes. A center for teachers is also responsible for ongoing improvements through pre-service and in-service workshops and annual conferences, and with building a strong community of practice.

In *Korea* teachers graduate from the universities of education or teachers' colleges, after that they have to go through evaluation for teacher qualification and are provided with a teacher certificate for each level of school. Only those who have a corresponding teacher's license may become teachers in elementary and secondary schools. Universities of education teach all the subjects that students in elementary schools learn, including CS. When prospective teachers complete the educational course of the department of computer education at the teacher's colleges, they can obtain a certificate for the secondary "informatics–computer" subject; only those are qualified for the informatics teacher recruitment exam. This is an annual national-level teacher recruitment examination, which includes a written test about the knowledge of the subject, pedagogical knowledge, essay, interview, and a demonstrative lesson. In order to maintain teachers' knowledge and skills, there is a 60 hours of training program running each year during holidays.

In *Georgia* (USA) the teachers who are allowed to teach computer science are mostly business teachers. They typically do not have much experience in advanced mathematics, science, or computer science. Within the *GeorgiaComputes* initiative, the professional development for in-service teachers was made through the Institute for Computing Education (ICE).

In *Russia* there is a requirement for basic degree in the subject, a Bachelor program with a single specialization in Informatics lasts four years, alternatively CS teachers can have a Bachelor program with a dual specialization, for example, "Mathematics and Informatics, which lasts 5 years. On any case there is an additional teaching practicum. Many teachers with other specializations come to Informatics after having taken professional courses offered by a variety of teacher-training institutions. Several programs have been developed to support in-service teachers training. For instance, pedagogical institutes, universities, and special institutes of professional training for teachers offer courses where innovative methods of Informatics teaching are instructed. Informal training is also supported from different societies of Informatics operate in various cities and regions.

In the UK, there are a variety of different ways in which to train as a teacher. The UK paper [8] discusses primarily *England* from this point of view. The most common is holding a postgraduate certificate in either primary or secondary education which lasts for one year, has a large component of actually teaching in the classroom and requires an undergraduate degree (for secondary education, usually in the subject in which one wants to teach). Alternatively, some undergraduate degrees in education are available which also give students qualified teacher status. Another option for prospective teachers is in-school training. Further, there is a very well organized network for supporting CS teaching. The CAS/BCS Network of Computer Science Teaching Excellence was started to address challenges and develop a network, building on the successful Computing At School regional network, that aims to support a thousand secondary schools in three years (a third of the total number of state-maintained secondary schools in England); with the appropriate regional coverage and support, the aim is for this network is to become self-sustaining at the end of the three years. With support from the Department for Education, BCS, The Chartered Institute for IT, Microsoft, Google, CPHC, OCR and AQA the Network was launched. At the heart of the Network of Teaching Excellence in Computer Science are a number of principles:

- face-to-face delivery,
- local delivery,
- teachers teaching teachers

Teacher training in *Sweden* does not offer courses in computing wherefore it is unusual to find teachers offering the content in school.

In *India* there is a focus on in-service training of teachers for the use of IT/ICT in education. The availability of trained and qualified personnel to teach CS subjects has also been a challenge, since in India there are few and scattered CS teachers. In particular, STEM teachers have always been assigned priority, followed by other teachers having some knowledge of computers. Moreover, there is a clear absence of formal training and certification programs for CS teachers in both the preservice and in-service phases. Schools in India require a minimum qualification of a 3-year Bachelor's degree and a 1-year Bachelor's in Education (B.Ed.) degree to teach grades 9 and 10 (low secondary). There is an additional requirement of a Master's degree in a related subject to teach grades XI and XII (high secondary). Empowering teachers with ICT training and tools, Intel's Teach India initiative kick-started in Mumbai, Bangalore, and Delhi through its Intel Teach program aimed at preservice and in-service teacher training.

9. DISCUSSION AND CONCLUSION

Originally, this work was intended to contribute to the discussion of the importance of rigorous CSE in K-12 schools. We wanted to distill this contribution from the 14 case studies on CSE in K12 schools that were published in the two special issues of the ACM TOCE journal on CSE in K12 schools. Our intention was to collect and classify different design decisions under different contexts of CSE in K-12 schools.

As we identified from the analysis of the case studies, there are very different ways to implement CSE in K-12 and many design decisions to be considered by different actors. In particular, we looked at the 14 case studies through the lens of the intended goals and competencies, content areas, programming languages and tools, forms of assessment as well as means and standards for training, certifying and keeping updated CS teachers. As expected, the analysis showed that the 14 approaches differ substantially in many respects. Thus, an important outcome of our work might be the detailed description of the communalities and differences of the investigated approaches regarding these fields.

Although our findings might provide meaningful outcomes for design decisions and their consequences, this study has certain limitations. As in all qualitative studies, the source of data has contingent effect on the results; this limits the extent of the generalization of findings to other states or countries. However, our intention was not to provide a complete and fully generalized model but to offer a snapshot of diverse CSE in K-12 case studies which might provide insights to different actors (e.g., stakeholders, curriculum designers, or teachers) and allow them to take informed decisions. A second limitation is caused by the lack of information that was sufficiently specific, e.g. regarding learning content. Aiming to reduce this limitation as far as possible, we communicated with the authors of the case studies and asked them to provide with certain missing information or to check some of our results. As we received this feedback from the majority of the case studies, this limitation was reduced as much as possible. The third limitation of this study comes from the fact that the analyzed case studies are based on the respective situation in 2013. For example, in Korea a new specification is being announced that will include computing as a subject in all primary schools from 2018.

Nevertheless, in our opinion the outcomes of our study could serve as useful guidance in the design process of CSE in K-12 schools. Additionally, we have worked out several overviews that might be helpful for stakeholders that aim to introduce or improve CSE in schools.

First, we have identified, described and compared 40 different terms that are applied to describe the fields of or around CSE. Also, we have investigated and compared the application and understanding of these terms in the respective countries (except USA) or states. This work might be used in the future to improve the understanding between stakeholders of different countries or states or even to sharpen the definition of CSE in a certain context.

Second, we have identified and normalized a list of goals and another list of competencies that are nominated in the case studies to describe the aims of CSE in schools, which could be used to inspire colleagues that are looking for arguments in favor of CSE in schools. The latter might be also applied to support the definition of future competency models.

Third, we have collected a large list of content elements, which we categorized in 19 content categories. This might be taken as a checklist for future curriculum development, or as a tool for the comparison or curricula.

Fourth, we have identified several programming languages and environments that are (or have been) used in the different countries or states.

Fifth, we have identified, described and compared several different examination forms that are applied in CSE. This could be used by the stakeholders to compare their approach with others or to introduce new examination forms in their country.

Finally, and also very importantly, we have compared the education policies for CS teachers over these countries (except USA) or states. By this, the stakeholders might get an impression of their educational level of the teachers in their respective countries or take some ideas to improve this level.

At the end of this work, we are confident that rigorous CSE will make its way as a regular subject into the curricula of all countries in the world. We hope that we were able to support this in some way or another.

10. ACKNOWLEDGMENTS

We have to thank several persons who stimulated or supported this long project. First, Barbara Owens for having the idea of the first ITiCSE working group in 2011, which developed the Darmstadt Model. Of course we have to thank all members of this group. Further, Valentina Dagiene deserves to be thanked for stimulating the special issues of the TOCE journals. Josh Tenenberg and Robert McCartney supported these issues with a lot of patience and many hints as editors-in-chief. All authors of the case studies put tremendous work in their papers and showed much patience with the long publication process. Finally, our special thanks go to Michal Armoni, who has contributed a lot of brilliant ideas, feedback and writing in the 2011 working group and by co-editing the two special issues with Peter and Michail. We regret very much that she was not able to join our group in Vilnius.

REFERENCES

[1] Anderson, L. W. and Krathwohl, D. R. 2001. *A taxonomy for learning, teaching, and assessing. A revision of Bloom's taxonomy of educational objectives*. Longman, New York.

[2] Astrachan, O. and Briggs, A. 2012. The CS principles project. *ACM Inroads* 3, 2, 38-42.

[3] Baron, G.-L., Drot-Delange, B., Grandbastien, M., and Tort, F. 2014. Computer Science Education in French Secondary Schools: Historical and Didactical Perspectives. *Trans. Comput. Educ.* 14, 2, 11:1-11:27.

[4] Bell, T., Andreae, P., and Lambert, L. 2010. Computer Science in New Zealand high schools. In *Proceedings of the Twelfth Australasian Conference on Computing Education - Volume 103*. Australian Computer Society, Inc, Brisbane, Australia, 15–22.

[5] Bell, T., Andreae, P., and Robins, A. 2014. A Case Study of the Introduction of Computer Science in NZ Schools. *Trans. Comput. Educ.* 14, 2, 10:1-10:31.

[6] Bellettini, C., Lonati, V., Malchiodi, D., Monga, M., Morpurgo, A., Torelli, M., and Zecca, L. 2014. Informatics Education in Italian Secondary Schools. *Trans. Comput. Educ.* 14, 2, 15:1-15:6.

[7] Brown, N. C. C., Kölling, M., Crick, T., Peyton Jones, S., Humphreys, S., and Sentance, S. 2013. Bringing Computer Science Back into Schools: Lessons from the UK. In *Proceeding of the 44th ACM Technical Symposium on Computer Science Education*. SIGCSE '13. ACM, New York, NY, USA, 269–274. DOI=10.1145/2445196.2445277.

[8] Brown, N. C. C., Sentance, S., Crick, T., and Humphreys, S. 2014. Restart: The Resurgence of Computer Science in UK Schools. *Trans. Comput. Educ.* 14, 2, 9:1-9:22.

[9] Choi, J., An, S., and Lee, Y. 2015. Computing Education in Korea&Mdash;Current Issues and Endeavors. *Trans. Comput. Educ.* 15, 2, 8:1-8:22.

[10] Dagiene, V. 2008. Teaching Information Technology and Elements of Informatics in Lower Secondary Schools: Curricula, Didactic Provision and Implementation. In *Informatics Education - Supporting Computational Thinking, Third International Conference on Informatics in Secondary Schools - Evolution and Perspectives, ISSEP 2008, Torun, Poland, July 1-4, 2008*. Lecture notes in computer science. Springer, 293–304. DOI=10.1007/978-3-540-69924-8_27.

[11] Diethelm, I. and Mittermeir, R. T., Eds. 2013. *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2013, Oldenburg, Germany, February 26 -- March 2, 2013, Proceedings*. Lecture notes in computer science 7780. Springer, Berlin, Heidelberg.

[12] Fecht, N. and Diethelm, I. 2014. Analysis of Computer Science Education in Venezuela Using the Darmstadt Model. In *Informatics in Schools. Teaching and Learning Perspectives*, Y. Gülbahar and E. Karataş, Eds. Lecture notes in computer science. Springer International Publishing, 41–53. DOI=10.1007/978-3-319-09958-3_5.

[13] Forbes, J. 2012. The CS 10K Project: mobilizing the computing community around high school education. *J. Comput. Sci. Coll.* 28, 1, 5.

[14] Gal-Ezer, J., Beeri, C., Harel, D., and Yehudai, A. 1995. A High School Program in Computer Science. *Computer* 28, 10, 73–80.

[15] Gal-Ezer, J. and Stephenson, C. 2014. A Tale of Two Countries: Successes and Challenges in K-12 Computer Science Education in Israel and the United States. *Trans. Comput. Educ.* 14, 2, 8:1-8:18.

[16] Goode, J. and Margolis, J. 2011. Exploring Computer Science: A Case Study of School Reform. *Trans. Comput. Educ.* 11, 2, 12:1-12:16.

[17] Gülbahar, Y. and Karatas, E., Eds. 2014. *Informatics in Schools. Teaching and Learning Perspectives*. Springer, Heidelberg, New York u.a.

[18] Guzdial, M., Ericson, B., Mcklin, T., and Engelman, S. 2014. Georgia Computes! An Intervention in a US State, with Formal and Informal Education in a Policy Context. *Trans. Comput. Educ.* 14, 2, 13:1-13:29.

[19] Heimann, P., Otto, G., and Schulz, W. 1965. *Unterricht. Analyse und Planung*. Schroedel, Hannover [etc.].

[20] Hubwieser, P. 2012. Computer Science Education in Secondary Schools - The Introduction of a New Compulsory Subject. *Trans. Comput. Educ.* 12, 4, 16:1-16:41.

[21] Hubwieser, P. 2013. The Darmstadt Model: A First Step towards a Research Framework for Computer Science Education in Schools. Keynote Talk. In *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages. 6th International Conference on Informatics in Schools: Situation, Evolution, and Perspectives, ISSEP 2013, Oldenburg, Germany, February 26 -- March 2, 2013, Proceedings*, I. Diethelm and R. T. Mittermeir, Eds. Lecture notes in computer science 7780. Springer, Berlin, Heidelberg, 1–14.

[22] Hubwieser, P., Armoni, M., Brinda, T., Dagiene, V., Diethelm, I., Giannakos, M. N., Knobelsdorf, M., Magenheim, J., Mittermeir, R. T., and Schubert, S. E. 2011. Computer science/informatics in secondary education. In *Proceedings of the 16th annual conference reports on Innovation and technology in computer science education - working group reports*. ITiCSE-WGR '11. ACM, New York, NY, USA, 19-38. DOI=10.1145/2078856.2078859.

[23] Hubwieser, P., Armoni, M., and Giannakos, M. N. 2015. How to Implement Rigorous Computer Science Education in K-12 Schools? Some Answers and Many Questions. *Trans. Comput. Educ.* 15, 2, 5:1-5:12.

[24] Informatics Europe and ACM. 2013. *Informatics education: Europe cannot afford to miss the boat. Report of the joint Informatics Europe & ACM Europe Working Group on Informatics Education*. http://europe.acm.org/iereport/.

[25] ISB. 2012. *Auswertung der Lehrplanumfrage im Fach Informatik*. http://www.isb.bayern.de/isb/ download.aspx?DownloadFileID=7f5ef746434bb6dc58bde1 72dc0eb98e. Accessed 16 August 2012.

[26] Khenner, E. and Semakin, I. 2014. School Subject Informatics (Computer Science) in Russia: Educational Relevant Areas. *Trans. Comput. Educ.* 14, 2, 14:1-14:10.

[27] Knobelsdorf, M., Magenheim, J., Brinda, T., Engbring, D., Humbert, L., Pasternak, A., Schroeder, U., Thomas, M., and Vahrenhold, J. 2015. Computer Science Education in North-Rhine Westphalia, Germany&Mdash;A Case Study. *Trans. Comput. Educ.* 15, 2, 9:1-9:22.

[28] Kurhila, J. and Vihavainen, A. 2015. A Purposeful MOOC to Alleviate Insufficient CS Education in Finnish Schools. *Trans. Comput. Educ.* 15, 2, 10:1-10:18.

[29] Mayring, P. 2000. Qualitative Content Analysis. *Forum: Qualitative Social Research* 1, 2.

[30] McCartney, R., Tenenberg, J., Hubwieser, P., Armoni, M., and Giannakos, M. 2015. Special Issue II on Computer Science Education in K-12 Schools. *Trans. Comput. Educ.* 15, 2.

[31] McCartney, R., Tenenberg, J., Hubwieser, P., Armoni, M., Giannakos, M. N., and Mittermeir, R. T. 2014. Special Issue on Computing Education in (K-12) Schools. *Trans. Comput. Educ* 14, 2.

[32] Raman, R., Venkatasubramanian, S., Achuthan, K., and Nedungadi, P. 2015. Computer Science (CS) Education in Indian Schools: Situation Analysis Using Darmstadt Model. *Trans. Comput. Educ.* 15, 2, 7:1-7:36.

[33] Repenning, A., Webb, D. C., Koh, K. H., Nickerson, H., Miller, S. B., Brand, C., Horses, Ian Her Many, Basawapatna, A., Gluck, F., Grover, R., Gutierrez, K., and Repenning, N. 2015. Scalable Game Design: A Strategy to Bring Systemic Computer Science Education to Schools Through Game Design and Simulation Creation. *Trans. Comput. Educ.* 15, 2, 11:1-11:31.

[34] Rolandsson, L. and Skogh, I.-B. 2014. Programming in School: Look Back to Move Forward. *Trans. Comput. Educ.* 14, 2, 12:1-12:25.

[35] Rudduck, J. 1985. Teacher research and research-based teacher education. *Journal of Education for Teaching,* 11(3) (1985), 281–289.

[36] Schulte, C., Hornung, M., Sentance, S., Dagiene, V., Jevsikova, T., Thota, N., Eckerdal, A., and Peters, A.-K. 2012. Computer science at school/CS teacher education: Koli working-group report on CS at school. In *Proceedings of the 12th Koli Calling International Conference on Computing Education Research*. Koli Calling '12. ACM, New York, NY, USA, 29–38. DOI=10.1145/2401796.2401800.

[37] Schwill, A. 1993. Fundamentale Ideen der Informatik. *Zentralblatt für Didaktik der Mathematik,* 25(1) (1993), 20–31.

[38] Seehorn, D., Carey, S., Fuschetto, B., Lee, I., Moix, D., O'Grady-Cuniff, D., Boucher Owens, B., Stephenson, C., and Verno, A. 2011. *CSTA K-12 Computer Science Standards. Revised 2011*. CSTA Standards Task Force. CSTA, New York.

[39] Stephenson, C., Gal-Ezer, J., Haberman, B., and Verno, A. 2005. *The New Educational Imperative: Improving High School Computer Science Education. Using worldwide research and professional experience to improve U. S. Schools. Final Report of the CSTA Curriculum Improvement Task Force*. ACM, CSTA, New York.

[40] Syslo, M. M. 2011. Outreach to Prospective Informatics Students. In *Informatics in Schools. Contributing to 21st Century Education*, I. Kalaš and R. T. Mittermeir, Eds. Lecture notes in computer science. Springer Berlin / Heidelberg, 56–70. DOI=10.1007/978-3-642-24722-4_6.

[41] Sysło, M. and Kwiatkowska, A. 2013. Informatics for All High School Students. In *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages*, I. Diethelm and R. Mittermeir, Eds. Lecture notes in computer science. Springer Berlin Heidelberg, 43–56. DOI=10.1007/978-3-642-36617-8_4.

[42] The Computer Science Teachers Association. 2013. *Bugs in the system: Computer science teacher certification in the U.S.*

[43] The Royal Society. 2012. *Shutdown or Restart. The way forward for computing in UK schools*. http://royalsociety.org/ uploadedFiles/Royal_Society_Content/education/policy/ computing-in-schools/2012-01-12-Computing-in-Schools.pdf. Accessed 12 November 2012.

[44] Tort, F. and Drot-Delange, B. 2013. Informatics in the French Secondary Curricula: Recent Moves and Perspectives. In *Informatics in Schools. Sustainable Informatics Education for Pupils of all Ages*, I. Diethelm and R. Mittermeir, Eds. Lecture notes in computer science. Springer Berlin Heidelberg, 31–42. DOI=10.1007/978-3-642-36617-8_3.

[45] Uljens, M. 1997. *School didactics and learning. A school didactic model framing an analysis of pedagogical implications of learning theory*. Psychology Press, Hove.

[46] Weinert, F. E. 2001. Concept of Competence: A conceptual clarification. In *Defining and Selecting Key Competencies*, D. S. Rychen and L. Salganik, Eds. Hogrefe & Huber, Seattle, 45–66.

[47] Wilson, C., Sudol, L. A., Stephenson, C., and Stehlik, M. 2010. *Running on Empty. Executive Summary*. http:// csta.acm.org/runningonempty/fullreport.pdf. Accessed 21 June 2011.

[48] Wing, J. M. 2006. Computational thinking. *Commun. ACM* 49, 3, 33-35.

APPENDIX

Table A1. Assessment-Examination in CSE in K-12 Schools

Case study	Level (e.g., upper secondary education)	Topics	Certification	Count for Admission	Application (e.g., universal, selected schools, counties)	Additional Information
NZ	Last three years of secondary	Wide Coverage	Yes: National Certificate of Educational Achievement	Yes	Universal	Work is assessed by a government agency (NZQA)
Bavaria	Upper and lower secondary	Wide Coverage	Yes (Abitur)	YES	Universal	1000 students per year
Israel	Junior High	Scratch, Robotics, Cyber			Still Piloting	Part of the matriculation exams
Sweden	Upper secondary education	Programming, Webserver programming, Web-technology			Universal	
England/ Wales (UK	GCSE (General Certificate of Secondary Education): optional, taken at age 14-16	Computer Science	Yes	May be used to judge entry to sixth form (age 16-18)	Theoretically all schools could offer it, but in practice it will only be a small proportion	School qualification bodies have developed General Certificate of SE
	A-Level (Advanced Level): optional, taken at age 16-18	Computing	Yes	Yes, counts for admission to university degree (age 18-21)	Theoretically all schools could offer it, but in practice it will only be a small proportion – a smaller proportion than GCSE above	Has existed for a long time, but currently being revamped to align better with the recently added GCSEs
France**	Independent (any level of Education)	ICT related	Yes (competency-based)		Universal	A new national curriculum is being implemented. It announces the provision of elements linked with CS starting from kindergarten. At junior high school level, new courses should also be launched, by teachers both from mathematics and technology.
Korea		Wide coverage	Yes	Yes	Universal	In addition to the school evaluation there is a test called CSAT
Russia	Compulsory for Lower secondary education Non comp for upper	Wide coverage	Yes (National Exams)	Yes (for CS related studies)	Universal for LSE School selection for HSE	In addition to the school evaluation there is a test called CSAT
India*	Upper secondary education	Wide coverage	Yes		Non Universal	20% Knowledge 30% Understanding 50% Application

* Unchecked by the authors, yet controlled by an Indian member of the working group
** Partly unchecked by the authors ("hard to say")

Table A2. Required Teacher Education

Case Study	Degrees			Training		Examination/ Certification	Informal teacher education
	B.Sc. in CS	Other Bachelor	M.Ed./M.Sc.	Pre-service training	In-service training		
New Zealand		X (in teaching)	X (post-graduation course)				X (local-national meetings, CS4HS workshops, other teacher support agencies)
Bavaria		X (related to the certain type of school)	X	X	X	X (one in subject and one on Ed/Psychol)	X (e.g., conferences
Israel	X			X (mandatory, CS teachers' preparation study program)	X (workshops by teachers' center)	X (mandatory)	X (training-workshops by a dedicated center)
Sweden				X			Swedish Informatics Teachers' Network (SITSNET)
UK	Not require, but would be advantageous	Required in normal state schools (usually in the subject you teach), along with PGCE (Postgraduate Certificate of Education), but in special "Academy" state schools, nothing is required	Not required, but would be advantageous.	The PGCE is generally trained for over a 1-2 year period.	Decided by individual schools	Qualified Teacher Status depends on the PGCE (but is not required in all schools)	CAS has many local volunteer-run hubs which offer informal training and meetup sessions. They also have a network of "master teachers" who offer training locally.
France	X (not all)				X	X (for all teachers)	
Korea		X (B.Ed.)		X	X	X (all teachers, plus a second certification for CS)	X (training programs on every vacations)
Russia		X (B.Ed with profile Informatics)		X (teaching practicum)	X (seminars and refresher course)	X (special final exams devoted to pedagogy and informatics)	X (e.g., conferences, societies)
Finland*		X (minor in school-pedagogy and a second school subject)	X (Major in subject)	X (included in minor pedagogy)			
India**		X (1 Y B.Ed.) + X (STEM degrees are prioritized)	X (To be able to teach grades XI and XII)	X			X (companies' guided teachers training)
Italy*			X (in STEM)				X (face to face workshops)

* Unchecked by the authors

** Unchecked by the authors, yet controlled by an Indian member of the working group