SOFTWARE ENGINEERING TEACHER NETWORKS - WHAT MOTIVATES TEACHERS TO COLLABORATE?

T. Kilamo
Tampere University, Tampere, Finland
ORCID 0000-0002-9561-1116
A. Knutas
LUT University, Lappeenranta, Finland
ORCID 0000-0002-6953-0021
A. Korhonen
Aalto University, Espoo, Finland
ORCID 0000-0002-2784-7979
T. Poranen*
Tampere University, Tampere, Finland
ORCID 0000-0002-4638-0243
O. Sievi-Korte
Tampere University, Tampere, Finland
ORCID 0000-0002-4956-8989

Conference Key Areas: Building Communities and Coordination, Fostering Engineering Education Research

Keywords: Teaching collaboration, networking activities, software engineering teachers, motivation to collaborate.

ABSTRACT

Teaching collaboration between universities is becoming increasingly important. Student intakes are increasing, and students are including MOOC courses from other universities to their studies. In Finland, there has been a long tradition in national teaching collaboration in computing education field. Our main research question is what motivates software engineering teachers to collaborate among universities. In this paper, we first give a short overview of national teaching collaboration from 1990's to the present day. Then, we present findings from a questionnaire that had respondents from active network participants. Main factors and practices that motivated teachers to participate in networking activities were existing project funding, active leaders and enough participants, interesting topics such as new innovations in learning technology, regular meetings and remote participation.

*Corresponding author

doi:10.5821/conference-9788412322262.1199

1 INTRODUCTION

Finland has long traditions in national teaching and research collaboration in the fields of computing education (CE) and computing education research (CER). Typically, active Computer Science (CS) teachers have also done controlled experiments related to their courses to study different teaching methods, developing new tools and software, and doing learning analytics, just to mention a couple of research activities. These early adopters have also collaborated among teachers from other universities to share the new innovations and practices with other pioneering stakeholders.

Currently, Computer Science is taught in almost every Finnish university. Most of the university level teachers in Finland have a PhD degree. First year basic programming courses are large and typically each course has hundreds of students. This is one of the motivations to develop novel learning tools and methods, but also to share the best practices with colleagues. It is not surprising that the teachers giving these first bachelor's level courses are the most active in the Finnish teacher networks as well.

As our theoretical lens, we use Lave and Wenger's communities of practice [1], as reviewed by Li et al. [2]. The communities of practice are a type of learning community that are characterised by "the support for formal and informal interaction between novices and experts, the emphasis on learning and sharing knowledge, and the investment to foster the sense of belonging among members" [2, 7]. These communities of practice are often informal networks, where learning occurs in social interactions between professionals rather than in a formal classroom settings. Experts mentor novices and eventually a professional identity forms for these communities. The informal interactions between the professionals are eventually systematized as means for professionals to improve practice and generate new ways to address recurrent problems [2]. In an previous study, Ryoo et al. [3] found that learning communities helped computer science teachers at high school level to transition to an inquiry-based classroom culture and break professional isolation.

In this paper, we study the history of established networks to recognise the enablers and blockers of the collaboration in Finnish context. We describe three past *network projects*, which institute the current collaboration among Finnish university teachers giving software engineering education (see Table 1). By a network project, we refer to a granted project that has a principal investigator, and limited time frame after which it is reported to the funding agency. In addition, we analyse four latest thematic *teacher networks* (Data Structures and Algorithms, Databases, B.Sc level programming, and Web Software Development), which might be project spin-offs, but attract enough participants to make the collaboration continue also after the original project has ended. In these networks, there is some fluctuation in the collaboration intensity throughout the years. In addition, the latest undertakings to run teacher networks seem to have different levels of activity, thus we do not dare to call them Special Interest Groups (SIGs), which we consider to be more established organizations possibly under some legal entity such as an association.

Our main research question is what motivates software engineering teachers to collaborate among universities. We present findings from a literature study as well as from a questionnaire that had respondents from active network participants. The results show that the main cohesive factor is project funding. However, having an active leader and participants is what keeps the network running. Moreover, shared interesting topics, regular meetings and remote participation seem to be important factors as well.

The rest of the paper is structured as follows. In Chapter 2, we first give a short overview of national teaching collaboration from 1990's to the present day. Chapter 3 gives an overview of the current situation. In Chapter 4, we report the results from a survey we conducted between

Table 1: List of former and current Finnish universities, which field of study is engineering or information and communication technologies including their participation in three different Teacher Networks between 2001-2020. The year for founding universities are boldfaced and underlined. Some of the universities have merged during the years, e.g., TUT and UTA are nowadays known as TUNI.

Members of Finnish University Networks in Software Engineering			
University Name (Acronym)	OSCU	NBEP	ÄlyOppi
Aalto University (AALTO) / Helsinki University of Technology (HUT)	2007*	**2006**	**2018**
LUT University (LUT)	2009		**2018**
Tampere University (TUNI) / Tampere University of Technology (TUT) / University of Tampere (UTA)	**2001** **2001**	**2006**	**2018**
University of Eastern Finland (UEF) / University of Joensuu (UJ) / University of Kuopio (UKU)	2007*	2007*	**2018**
University of Helsinki (UH)	2007*	2007	**2018**
University of Jyväskylä (JYU)	2009	2006	**2018**
University of Oulu (UO)	**2001**		2020
University of Turku (UTU)	2007*	**2006**	**2018**
University of Vaasa (UWA)	2007*		2020
Åbo Akademi University (ABO)			2020

December 2021 and January 2022. Finally, in Chapter 5 we make some conclusions.

2 TEACHER NETWORKS IN FINLAND

Table 1 gives an overview of the network projects conducted between 2001 and 2020. The table indicates the year each university has attended the network.[1] The year is boldfaced and underlined in case of founding universities.

2.1 OSCu Network – Open Source Courseware Network

In 2001, three universities (UTA, TUT, UO) founded the Open Source Courseware (OSCu) network [4]. The philosophy behind OSCu was to be able to share courses and improve cooperation between universities. Courses given through the OSCu network were expected exceptionally high quality documentation following common guidelines and practices, and teachers were given training on how to give online lectures for remotely participating universities. The network grew quickly and in 2007 it already involved eight universities (TUT, UTA, UTU, UWA, UH, UO, HUT, UKU), with two more universities joining in 2009 (JYU, LUT). The network activities were at their peak in year 2007, when 8 course implementations were offered across universities, after which the offering started to diminish, as only 2-4 courses were arranged each year 2008-2010, and from 2011 onwards only one course was offered. OSCu's way of providing the courses was such that a course lectured on one of the participating universities was streamed to all other participating universities via a video bridge and Virtual Network Computing (VNC). In other shared sessions Adobe Connect Pro was used for communication. Each university had local arrangements for handling the stream, organizing exercise sessions, and so on. The most popular course was "Software Architectures", lectured at TUT, which

[1] Records from OSCu are incomplete between years 2001-2007. While universities have joined the network at different stages between 2001-2007, we only have access to data on who were part of the network at the end of 2007.

was given yearly while other offered courses varied every year. The course became such a staple that other universities in the network began to rely on receiving lectures from TUT, even after the OSCu network itself was discontinued. Software Architectures was shared to other universities until 2015, while other OSCu activities quietly faded away in the turn of 2010-2011, due to little enthusiasm to share other courses, critical changes in university representatives in the network (mainly, the surprising death of the founding member), and ending of ministry funding for virtual teaching networks.

2.2 NBEP – The Network for Basic Education in Programming

The Network for Teaching Basic Programming (NTBP) (Ohjelmoinnin perusopetuksen verkosto) was funded by Ministry of Education, and founded by three universities (HUT, UTU, TUT) in 2006 [5]. It was operating in the fields of engineering and computer science to coordinate national and international activities. These activities included seminars, which were organized in the participating universities and had also international keynote speakers from Europe, Australia, and USA. Many participating teachers were also active researchers in the field of Computing Education Research. The goal was to research, develop, disseminate, and maintain tools and systems for teaching basic computer science courses. The tools included software for automatic grading and feedback [6–9], style checkers [10], visualizations and animations [11], as well as learning management systems [7, 8]. Many researchers and research groups in the network still collaborate and attend many international conferences such as those organized by ACM/SIGCSE (Association for Computing Machinery / Special Interest Group on Computer Science Education). The SIGCSE provides an international discussion forum for teachers and researchers in the field to exchange ideas on software and training development, courses, training programs, and other elements related to the teaching and pedagogy of computer science.

The network worked in close co-operation with the Computer Science Teaching SIG, a thematic group still active within the Finnish Society for Computer Science. The aim of the theme group is to bring together teachers in the field of computing education and researchers interested in teaching technology. The purpose is to promote e.g. cooperation between teachers, the exchange of teaching and learning materials, the introduction of novel tools to support teaching, and taking a stand on current societal issues in the field of teaching programming.

3 CURRENT NETWORKS

Universities that were active in OSCU and NBEP projects started to prepare a funding application together with Arcada University of Applied Sciences, Metropolia University of Applied Sciences and Tampere University of Applied Sciences in early 2018. Aalto University coordinated the application preparation. The Ministry of Education and Culture gave funding for this project called Intelligent Systems and Content Creation project (Älykkäät oppimisympäristöt ja niiden sisällöntuotanto in Finnish, shortly ÄlyOppi) for years 2018-2021 [ÄlyOppi, 2022]. In addition of funding the development of smart learning environments, part of the funding was targeted to establish teacher networks where teachers can demonstrate new developed systems, give support to take systems in use in other universities, create course materials together and then to share teaching and learning materials.

First teacher network meeting was held in December 2019. There were about 40 participants coming from eight Finnish universities. Initially it was planned to divide participants into five groups: i) Bachelor level programming, ii) Data structures and algorithms, iii) Databases, iv) Web-programming, and v) other special areas, like machine learning and data security. First three of the groups got most of the pre-registrations, last two groups only a few. The three most popular groups began their networking activities, and last two are still waiting for active

leader and participants. These three groups also relate to topics lectured in all universities in the fields of engineering or information and communication technologies. Data structures and algorithms also has very active international teacher networks.

3.1 Data Structures and Algorithms

ÄlyOppi established The Teacher Network for Data Structures and Algorithms (TN-DSA) to enable call for participation for Finnish stakeholders that are interested in developing such courses in collaboration with instructors, teachers, researchers, and developers from other institutions. The different implementations of these courses range from pure lecture courses to realisations that heavily rely on automatically assessed assignments (AAA). The assignments can include simple multiple choice questions, but also smart content such as automatically assessed programming exercies [6, 9, 12] and algorithm simulation exercises [11, 13]. TN-DSA offers a forum to exchange ideas and materials among the teachers and other stakeholders. Due to COVID-19, TN-DSA has convened in Zoom. The meetings took place about every other month in 2020 (6 times), and almost every month in 2021 (11 times). Several interesting themes have emerged from the discussions, which are discussed in the following paragraphs.

Textbooks. The current trend in higher education is towards open courses and free learning materials including textbooks. This is also the case in DSA courses. There are several freely available textbooks that have gained a lot of interest in the DSA community. For example, Shaffer's OpenDSA [14] is a textbook developed further in collaboration with many researchers and instructors worldwide to include also interactive exercises and smart content [15]. Another example of freely downloadable material is the textbook written in Finnish by Laaksonen [16]. Both of these textbooks are examples of freely available materials to be shared among teachers and students, and which have introduced and discussed in the TN-DSA meetings.

Course arrangements. Many DSA courses were well prepared to COVID-19 as they already had a lot of automation and many students were already able to pass the courses almost without attending the campus teaching at all. For instance, the first automated assignments were introduced in Aalto University as early as 1991 [17]. In the corresponding DSA course, the development of novel automated exercises have continued ever since. For example, visual algorithm simulation exercises (VAS) [7, 13] were developed during the millennium changeover period. Nowadays, the course consists of multiple choice exercises, VAS exercises, and programming assignments that are all assessed automatically and the students can get immediate feedback on their submissions. This kind of assignments and learning management systems capable of hosting automatically assessed exercises are examples of demonstrations in the TN-DSA meetings. However, the idea is not only to exchange software, but also to learn from each other to maintain and develop courses further and discuss, for example, how to produce online learning materials or how to create innovative lecture videos.

New assignments. One of the issues with assignments is that they need to be varied once in awhile to prevent the students passing and sharing the solutions. Finding good new assignment topics can be laborious process. Thus, even though the assignments do not have any kind of automatic assessment, sharing the topics could be a valuable aid for teachers. Especially topics that have been tested with students and maybe elaborated based on the feedback can be very beneficial. This is also one of the frequent discussion points in the meetings.

Term Bank. New terms in computer science appear typically first in English. However, many universities in Finland teach DSA also in Finnish and Swedish. This has the consequence that different universities might have translated the terminology a little bit differently. It is a constant debate how to translate and try to harmonise the terminology. The developed

Term Bank includes definitions of the terms, links to external material (such as Wikipedia, Wiktionary or The Helsinki Term Bank for the Arts and Sciences [18]), and the local and most common translations among different universities in Finland. This is an ongoing project, which aims at expanding to other teacher networks than TN-DSA as well.

3.2 Databases

As a part of the ÄlyOppi project efforts to establish teacher networks, databases was identified as one of the networks due to SQL databases being an established concept with fundamentals changing only slowly. For this reason, it was thought that database fundamentals courses would have a potential to share.

The full establishment of the databases network (TN-DB) was slowed down due to COVID-19 and has convened few times per year over Zoom. During these introductory meetings, shared network interests were identified. Initially, these were: Sharing and listing open educational resources such as textbooks or query practise tools, showcasing course structures that use these resources, and comparing degree programs' course structures, to see how they implement the curriculum.

Although it has not been under the network umbrella, there has been some deeper one-to-one collaboration among TN-DB partners. For example, LUT has previously used JYU's openly available web-based coursebook for databases with embedded exercise tasks. Currently the databases network is in the process of mapping course units and has plans to reconvene to review the outcomes in 2022.

3.3 Bachelor level programming

Bachelor level programming lays the basis of computing curricula through higher education (HE). The teacher network for bachelor level programming (TN-BLP) was established in ÄlyOppi to support collaboration and information sharing. TN-BLP can be seen as a direct successor of NBEP.

Bachelor level programming is such a staple of computing curricula that it is a natural as well as a challenging topic for the network. Each HE provider offers teaching that covers the same learning outcomes while each have organized the teaching to meet the learning expectations of their specific degree programme. This means that the courses under the umbrella of TN-BLP vary in size, organization and programming language used.

TN-BLP included all universities participating in ÄlyOppi. The network initially met face to face but due to the COVID-19 pandemic, the meetings were held over Zoom. The network met based on availability and need. The topics discussed:

Courses included. At start the network collected the names and learning outcomes of courses that make the bachelor level programming in each participating university. This work covered also the programming languages used by each university. It is worth to note that while most commonly, all universities use Python, Java or Scala or C++ for learning, there are several different approaches to the order of the languages and the frequency of moving to a new language.

Course arrangements. The teaching of bachelor level programming has relied on automated grading and the use of flexible learning allowing for off-campus participation even before the COVID-19 pandemic hit. Still, how to organize teaching and especially assessment in remote teaching was a much discussed topic at the beginning of the pandemic. Experiences on the learning management systems and ways to automatically grade assignments and exam answers have also been shared.

Bridge Courses. As the student intake in computing has increased the backgrounds of students on the courses has become more heterogeneous. Furthermore, students' mobility between HE providers has increased. Hence one goal set by TN-BLP was to create a collection of small course modules coineed bridge courses to make collecting the necessary learning from several HE providers even though the programming languages are not compatible from the learning point of view directly. The pandemic brought this work to a halt.

Shared Courses. The idea of sharing course material and programming tasks has been discussed. There has also been sharing of full courses between two network participants. However, a wider sharing of course content has not been established. As a topic, this has increased interest to participate in the network among the Universities of Applied Sciences.

4 SURVEY ON WHAT MOTIVATES TEACHERS TO COLLABORATE?

Teacher communities are claimed to contribute to the improvement of teaching and schooling practices [19], where learning occurs in a social system through a dual process of meaning making. In addition to building and sharing profession-related artefacts (courses), these communities learn through activities and store the outcomes of learning as a body of knowledge in the form of words, tools, concepts, methods, stories, documents, and links to resources [20]. The participation and sharing has been reported to motivate newcomers [19].

This section describes the data collection and analysis in the context of Finnish teacher networks, a type of teacher community. In addition, findings on what themes the communities worked on and what factors supported participation in the communities are presented.

4.1 Data gathering

Data was collected during December 2021 and January 2022 using a web form. The form link with a brief description of the research was first sent to all teacher network participants, and later further distributed via the Finnish Society for Computer Science. The form had three background questions. First question asked consent (yes / no) to use respondent's answers in this research. Second question asked respondent's university, and third question asked respondent's teaching experience (0-4 years, 5-9 years, 10-14 years, 15-19 years, and 20 or more). Second and third questions were voluntary.

After the background questions, the fourth question was on which teacher network the respondent was involved in or whether they were answering to teacher network activities in general and not related to any specific network. The fifth question asked what themes the network has dealt with so far, and the sixth question asked what themes should be considered in future. The seventh question asked factors that support participation in networking activities. The last question was for free comments and future development ideas.

4.2 Findings

The questionnaire was answered by 13 respondents from 7 universities, all of which gave permission to use the data in the research. All the respondents were experienced teachers: nine had 20 years or more teaching experience, one 15-19 years, and three 10-14 years. The answers were spread over the teacher networks so that 5 were for TN-DSA, 2 for TN-DB, 1 for TN-BLP, 1 for Web Programming and 4 for network activities in general. Responses to question on themes covered in meetings match with the actual content of the network meetings according to the network leaders as described in Sub-chapters 3.1-3.3.

The following themes for future network meetings were mentioned in answers to the question six: common transition courses (between programming languages), more teachers should be involved in networking activities, electronic exams and evaluation in general, what kind of advanced level database courses are in universities, tools and methods related to distance

learning, university level collaboration, developing open access course materials and exercise banks, and developing and sharing knowledge on learning management systems. Some themes that require new networks were also mentioned: data security, platform economy, information systems, operating systems, and programming larger scale applications.

Answers of questions 7 and 8 were classified by keywords into different topics. The seventh question surveyed the factors that supported participation in the networking activities. Most of the answers were related to meeting arrangements. Possibility to participate remotely was mentioned by 9 respondents. A few respondents mentioned that occasional face to face meetings would also be good. Many of the respondents mentioned factors that were related to the meeting practices: regular and enough meetings to keep connected (mentioned by 2 respondents), one hour per meeting (1) and two hours is maximum (2), usually one main topic per meeting (1), champion who leads and organises the meetings (1) and updates shared memo (1), and meeting recordings (1). Another theme that clearly raised from the answers was content of the meetings. Possibility to discuss with colleagues, give or follow tool demos and share ideas that can be used in teaching were seen important (2). General level goal setting was also mentioned by a respondent. Teacher networks should agree common tasks or goals with responsible persons. Required effort of the tasks should be realistic especially when there is no funding. One of the respondents also mentioned concrete outcomes as a motivating factor.

5 DISCUSSION

From the questionnaire data, several collaboration enablers were detected. Our data is collected from Finnish University teachers and instructors. Most of the answers mentioned importance of the meeting practices and arrangements: regular meetings to keep teachers connected, remote participation possibility, length of the meeting from one hour to two hours, as well as, active champion who leads and organises the meetings. In addition, content of the meetings was seen important. A meeting should provide a place to discuss with the colleagues, see tool demos and share ideas, and ideate teaching collaboration and material creation. However, the same factors can be the blockers as well. The end of a project is a critical moment. The network might also lose the champion that brings the participants together if the continuation of the networking is not planned during the project.

Although information about the network and network activities was disseminated successfully and a significant portion of the relevant field of education was reached, it was found that uptake and reuse of the materials remains an open question despite project funding. Some successes have been found due to good relationships between individuals and build of trust, but no new widely applicable approaches have been found. To summarize the dichotomy, awareness about the materials has been increased, but increased uptake of shared resources requires ongoing work. Traditional textbooks are still the most popular way of disseminating content, but disseminating interactive smart content remains hard to achieve.

Fortunately, from the learner point of view, utilization of resources has taken other forms during the last 10–15 years. MOOCs (Massive Open Online Courses) seems to be the contemporary way of sharing educational content. In addition, open education movement is currently actively trying to broaden access to the learning traditionally offered through formal education systems. Automatic assessment is one of the enablers that makes it easy to allow participation in online learning even though the number of participants is very large. MOOC seems to be the new book.

Although our data is only from Finland, the ebb and flow in network activity is a common aspect of all networks. Depending on the current topics discussed in the network and on

the very human aspect of how busy the core network participants are in their daily work, the networks go through active phases followed by a quieter or even inactive phase. COVID-19 pandemic can be seen as one outside factor both driving network activity seen in TN-DSA, but driving network activity down due to the needed major shift in how education is provided seen in TN-BLP. There is also an overlap in the themes of the networks; when one network is active the participants may be less active in the other networks they participate in. The size of the network is important here. For the networks to thrive both shared interests or future directions and a leader or core group of people is needed to keep the networks going.

6 ACKNOWLEDGEMENTS

We thank the Intelligent Systems and Content Creation project (ÄlyOppi) funded by the Ministry of Education and Culture, and Finnish Society for Computer Science for supporting this work.

References

[1] Etienne Wenger et al. Communities of practice: Learning as a social system. *Systems thinker*, 9(5):2–3, 1998.

[2] Linda C Li, Jeremy M Grimshaw, Camilla Nielsen, Maria Judd, Peter C Coyte, and Ian D Graham. Evolution of wenger's concept of community of practice. *Implementation science*, 4(1):1–8, 2009.

[3] Jean Ryoo, Joanna Goode, and Jane Margolis. It takes a village: Supporting inquiry-and equity-oriented computer science pedagogy through a professional learning community. *Computer Science Education*, 25(4):351–370, 2015.

[4] Kirsti Ala-Mutka and T. Mikkonen. Experiences with distributed open source courses. *Informatica*, 27:243–254, 2003.

[5] Ari Korhonen and Eija Ristimäki. Finnish virtual university network for teaching basic programming. In Mats Daniels, editor, *Materials and Presentations at the 4th CeTUSS Workshop, 4–5 Dec 2006*, pages 1–2 p., 4th CeTUSS Workshop, December 4 - 5, Uppsala University, Uppsala, Sweden, 2006. Uppsala University, Department of Information Technology. URL `http://www.it.uu.se/research/group/cetuss/Events/2006-11/WS4-material/korhonen-abstract.pdf`.

[6] Kirsti Ala-Mutka. A survey of automated assessment approaches for programming assignments. *Computer Science Education*, 15(2):83–102, 2005.

[7] Ari Korhonen, Lauri Malmi, and Panu Silvasti. TRAKLA2: a framework for automatically assessed visual algorithm simulation exercises. In *Proceedings of Kolin Kolistelut / Koli Calling – Third Annual Baltic Conference on Computer Science Education*, pages 48–56, Joensuu, Finland, 2003.

[8] Ari Korhonen, Lauri Malmi, Panu Silvasti, Jussi Nikander, Petri Tenhunen, Pekka Mård, Harri Salonen, and Ville Karavirta. TRAKLA2. Computer program (http://trakla.cs.hut.fi/), 2003. URL `http://www.cs.hut.fi/Research/TRAKLA2/`.

[9] Riku Saikkonen, Lauri Malmi, and Ari Korhonen. Fully automatic assessment of programming exercises. In *Proceedings of The 6th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'01*, pages 133–136, Canterbury, UK, 2001. ACM Press, New York.

[10] Kirsti Ala-Mutka, Toni Uimonen, and Hannu-Matti Järvinen. Supporting students in C++

programming courses with automatic program style assessment. *Journal of Information Technology Education*, 3:245–262, 2004.

[11] Ville Karavirta, Ari Korhonen, and Lauri Malmi. MatrixPro. Computer program, November 2003. URL http://www.cs.hut.fi/Research/MatrixPro/.

[12] Sébastien Combéfis. Automated code assessment for education: Review, classification and perspectives on techniques and tools. *Software*, 1(1):3–30, 2022. ISSN 2674-113X. doi: 10.3390/software1010002. URL https://www.mdpi.com/2674-113X/1/1/2.

[13] Ari Korhonen and Lauri Malmi. Algorithm simulation with automatic assessment. In *Proceedings of The 5th Annual SIGCSE/SIGCUE Conference on Innovation and Technology in Computer Science Education, ITiCSE'00*, pages 160–163, Helsinki, Finland, 2000. ACM Press, New York.

[14] Clifford A. Shaffer, Ville Karavirta, Ari Korhonen, and Thomas L. Naps. Opendsa: Beginning a community active-ebook project. In *Proceedings of the 11th Koli Calling International Conference on Computing Education Research*, Koli Calling '11, page 112–117, New York, NY, USA, 2011. Association for Computing Machinery. ISBN 9781450310529. doi: 10.1145/2094131.2094154. URL https://doi.org/10.1145/2094131.2094154.

[15] Eric Fouh, Ville Karavirta, Daniel A. Breakiron, Sally Hamouda, Simin Hall, Thomas L. Naps, and Clifford A. Shaffer. Design and architecture of an interactive etextbook – the opendsa system. *Science of Computer Programming*, 88:22–40, 2014. ISSN 0167-6423. doi: https://doi.org/10.1016/j.scico.2013.11.040. URL https://www.sciencedirect.com/science/article/pii/S016764231300333X. Software Development Concerns in the e-Learning Domain.

[16] Antti Laaksonen. *Tietorakenteet ja algoritmit*. Published over the internet by Creative Commons BY-NC-SA 4.0, https://github.com/hy-tira/tirakirja, 2021. URL https://www.cs.helsinki.fi/u/ahslaaks/tirakirja/.

[17] Juha Hyvönen and Lauri Malmi. TRAKLA – a system for teaching algorithms using email and a graphical editor. In *Proceedings of HYPERMEDIA in Vaasa*, pages 141–147, 1993.

[18] The helsinki term bank for the arts and sciences. Visited Apr 29, 2022. ISSN 2489-7035. URL https://tieteentermipankki.fi/wiki/Termipankki:Etusivu/en.

[19] Panagiotis Tsiotakis and Athanassios Jimoyiannis. Critical factors towards analysing teachers' presence in on-line learning communities. *The Internet and Higher Education*, 28:45–58, 2016.

[20] Jean Lave and Etienne Wenger. Communities of practice. *Learning, meaning and identity*, 1999.