3.2. Related research

***Version 1 (Jätin tämän nyt tähän kun Haluan mielipiteen että onko tällä enää mitään arvoa* 🙂*):***

As automatic assessment in academic environment has been around for over sixty years, it has been the topic of numerous research during the years. This section briefly introduces a selection of research and their findings relevant to the topics of this thesis.

Manuel Rubio-Sánchez et al. assessed an AAT system *Mooshak* in their research *Student perception and usage of an automated programming assessment tool* in which they used *Mooshak* in an algorithm designing course. The research aimed to answer the following research questions:

1. “To what extent do students use *Mooshak* as a substitute for their own testing and debugging?”
2. “How do students experience the usage of *Mooshak* and how would they improve the tool?”
3. “Whether the usage of *Mooshak* influences the dropout rate?”

Out of the research questions, the 2nd and to some extent the 3rd questions are the most relevant to address for the sake of this research. In order to get an answer to the 2nd question, they collected data from students with questionnaires containing various questions about the performance of the AAT and analyzed the opinions and critiques, and for the 3rd question they analyzed the course statistics. [13]

The main finding of the study was that the students didn’t appreciate using the tool in general despite finding it interesting or good idea to implement in a course. The main problem the students had with the AAT was that the feedback was lacking and in need of enhancing. Regardless of the AAT being disfavored by the students, a correlation between using the AAT and the number of dropouts wasn’t found. [13]

Instead of assessing an already existing AAT, Li-Chen Cheng et al. created their own in their study *Effects of an automated programming assessment system on the learning performances of experienced and novice learners*. In this study, they created n AAT system called *APAMP*. The main focus of the research was to see if there was any difference in the learning achievements, attitude and motivation of the more experienced students compared to the novice programmer students during the course, using the feedback of the AAT as a learning tool. [11]

In the study, a preliminary test and questionnaire were conducted, which was used to evaluate the baseline of the student’s skills, motivation and attitude. After the end of the course, another set of questionnaires and tests were caried out, which was used to evaluate the changes that happened during the course. [11]

The study found out that while the AAT improved both group’s ability to program, experienced programmers benefited more from it. Additionally, both, the learning attitude and motivation of the novice group dropped significantly, while the experienced group’s attitude significantly increased, while the motivation stayed the same. [11]

Aleksi Järventausta did a master’s thesis about creating the assignments and utilizing CodeGrade in the previous web programming course of LUT University, named *Web applications*. In his study, Järventausta created the weekly assignments and automatic assessments for them, and researched, how CodeGrade compares to the previous assessment tool, and how should the assignments be constructed and assessed and how does the course compare to the previous course. [12]

The evaluation of the course was done using two feedback surveys. One smaller survey was conducted in the middle of the course, to give students a chance to express their concerns about the course. Additionally, a larger-scale survey was conducted in the end of the course to evaluate the course and the AAT. Additionally, student comments, behavior and feedback was taken into account. [12]

As in previously mentioned research, Järventausta’s research indicated the importance of the feedback provided by the AAT. Furthermore, the course was considered an improvement in almost every aspect compared to the previous course. An important find was the difficulty curve of the course, as the number of students who submitted an assignment dropped drastically from week 4. Järventausta’s research also pointed out that the feedback and assignment descriptions needed to be rewritten. [12]

Based on the research already done on the subject, the feedback provided by the AAT needs to be clear and give enough information about a failed test. Additionally, the assignment descriptions need to be detailed enough to not be ambiguous. The related research also suggest that more experienced programmers have a greater experience and gain more from automatic assessment.

It is also noteworthy that according to the previous research,

***Version 2***

As automatic assessment in education environment has been around for over sixty years, it has been the topic of numerous research. This section covers an integrative literature review of the topic of automatic assessment in a programming education environment and compiles relevant findings of different research and analyses it.